JOSIP JURAJ STROSSMAYER UNIVERSITY OF OSIJEK

FACULTY OF ELECTRICAL ENGINEERING, COMPUTER SCIENCE
AND INFORMATION TECHNOLOGY OSIJEK

Matej Arlović

Segmentation and Detection Of Indoor Fires Using Deep Learning Methods

Doctoral Dissertation

Osijek, 2026.

Doktorska disertacija izrađena je na Zavodu za programsko inženjerstvo Fakulteta elektrotehnike, računarstva i informacijskih tehnologija Osijek, Sveučilišta Josipa Jurja Strossmayera u Osijeku.

Doktorska disertacija izrađena je u sklopu projekta Hrvatske zaklade za znanost "Semantička segmentacija termovizijskih i rgb slika konvolucijskim neuronskim mrežama u svrhu autonomnog kretanja robota i rane detekcije požara" (broj projekta: DOK-2021-02-7509)

Mentor: izv. prof. dr. sc. Josip Balen, izvanredni profesor, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, Sveučilište Josipa Jurja Strossmayera u Osijeku

Disertacija ima STRANICA stranice.

Disertacija broj: UNIJETI BROJ PRIJE UVEZA.

Povjerenstvo za ocjenu doktorske disertacije:

1. Prof. dr. sc. Emmanuel Karlo Nyarko, redoviti profesor, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, Sveučilište Josipa Jurja Strossmayera u Osijeku
2. Doc. dr. sc. Hrvoje Leventić, docent, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, Sveučilište Josipa Jurja Strossmayera u Osijeku
3. Prof. dr. sc. Marina Ivašić Kos, redoviti profesor, Fakultet informatike i digitalnih tehnologija, Sveučilište u Rijeci

Povjerenstvo za obranu doktorske disertacije:

1. Prof. dr. sc. Emmanuel Karlo Nyarko, redoviti profesor, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, Sveučilište Josipa Jurja Strossmayera u Osijeku
2. Doc. dr. sc. Hrvoje Leventić, docent, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, Sveučilište Josipa Jurja Strossmayera u Osijeku
3. Prof. dr. sc. Marina Ivašić Kos, redoviti profesor, Fakultet informatike i digitalnih tehnologija, Sveučilište u Rijeci

Datum obrane doktorske disertacije: UNIJETI DATUM.

Acknowledgments

Sic parvis magna
— *Sir Francis Drake*

First, I would like to thank my supervisor and mentor, Josip Balen, for his patience, and for discussing all my various ideas. I would also like to thank my unofficial co-mentor, Franko Hržić, for everything he taught me, for guiding me through my PhD, for all the working holidays, and for inspiring discussions that sparked many crazy ideas.

I would like to express my sincere gratitude to the president and all members of my PhD Examination Board for their time and effort spent on the evaluation, as well as for their insightful feedback and involvement in the evaluation procedure. I would also like to thank my office mates, Petar and Davor. They were not only colleagues and co-authors, but also friends who stood by me in difficult moments, offering encouragement and contributing in many ways to my research. I wish to thank my colleagues from FERIT, in particular Miljenko, Robert, and Krešimir, for their support and assistance.

I would like to thank my friends Veronika R., Franjo Josip, and Kristijan for their support and for their patience with my many absences throughout my Ph.D. I am grateful to my family, who gave me great support and inspired me to pursue science. Especially my cousin Ana, who was a big inspiration for me to develop myself in scientific work, and my cousin Damir, who sparked my love for computers and technology. I would also like to thank my late parents, Ružica and Zlatko, who always made sure I had everything I needed so that my only concern was school. I am also thankful to my older brother Josip, who has been a great support throughout my PhD and has always been patient with me. Finally, I would like to thank my girlfriend Veronika T., who was my bedrock and who motivated me always to try again, who listened to all of my crazy ideas (and God knows I had too many of them), and who always gave me advice.

Matej Arlović
January, 2026.

Abstract

Segmentation and Detection Of Indoor Fires Using Deep Learning Methods

Fire is a chemical process of rapid oxidation of combustible material during which heat and light are released, often accompanied by various gaseous combustion products. From a physical perspective, fire represents a visible manifestation of energy transfer in the form of electromagnetic radiation and heat generated during this exothermic process. Fires are generally classified as either indoor or outdoor, depending on the spatial environment in which they occur. Indoor fires develop within enclosed spaces such as offices, warehouses, or factories, while outdoor fires emerge in open areas, including fields, forests, and the building exteriors. Their characteristics differ not only in terms of setting but also in their dynamics and methods of suppression. Indoor fires are characterized by the rapid accumulation of heat and smoke, which necessitate direct entry into the structure and careful fire suppression techniques. Outdoor fires, in contrast, are strongly influenced by weather conditions and terrain, making suppression reliant on wide containment lines and often aerial support. In 2023, fire departments in the United States responded to more than 1.39 million fires, which resulted in 3,670 civilian deaths and 13,350 injuries. The direct material losses were estimated at 23 billion US dollars.

There are two primary approaches to fire detection: traditional, sensor-based, and modern, image-based. The traditional method relies on sensors that monitor physical parameters such as temperature, humidity, pressure, and the concentration of toxic gases. While widely used, such systems are often have long response time and may fail to detect a fire in its early stages, or in some cases, miss it entirely. More recently, image-based methods have emerged as a superior alternative. By analyzing visual data, these systems provide critical information on the location, size, and development of a fire. They may also identify hazardous scenarios that could trigger ignition, thus preventing the fire. Early research employed classical computer vision techniques, such as contour detection and k-means clustering, but the results were often unreliable and slower than those of sensor-based approaches. The development of deep learning has marked a turning point. Deep neural networks can autonomously learn representations and extract features from visual data, thereby eliminating the need for handcrafted feature design, which was a central component of earlier methods. These networks identify hierarchical patterns, ranging from simple edges and textures to complex objects and scenes, making them highly effective for recognition, segmentation, and classification tasks. This dissertation places

particular emphasis on semantic segmentation, which enables the precise extraction of flame and smoke regions from an input image. Such precision is critical for reliable indoor fire detection. Consequently, deep neural networks are increasingly utilized for fire detection in images and videos, demonstrating superior detection speed and accuracy compared with earlier approaches. Nevertheless, detecting fires in indoor industrial environments remains a highly challenging task. Such incidents are rare, dangerous to document, and often occur in restricted or inaccessible spaces, which limits the availability of real-world training data. Furthermore, the visual characteristics of fire can resemble light reflections and other disturbances, making recognition more challenging. For these reasons, this dissertation focuses on the detection of indoor fires in industrial settings and examines how deep learning can help overcome these obstacles. Despite rapid advances in deep learning and the increase of computational power through graphics processing units (GPUs), researchers still face considerable challenges. The main challenge is the shortage of high-quality datasets. Majority of existing research has focused on outdoor fires, particularly forest fires, resulting in datasets tailored to such contexts.

In contrast, studies of indoor fires remain scarce, largely because specialized public datasets are lacking. Unlike outdoor scenarios, where extensive image and video collections exist, acquiring real-world indoor fire data is extremely difficult due to the rarity of such events, the risks involved in its acquisition, and the limitations imposed by private or restricted spaces. Another challenge is annotating fire images, as flames and smoke are naturally dynamic, with irregular and constantly shifting shapes that make precise annotation difficult. The problem is further compounded by visual similarities between fire and other phenomena such as glare, reflections, or environmental noise, which increase the likelihood of annotation errors and reduce the reliability of machine learning models. Synthetic data provides a promising solution to these challenges. Generated through algorithms and simulations, synthetic datasets are particularly valuable when real data are scarce, costly to annotate, or biased. If synthetic images incorporate realistic fire features and visual authenticity, they enable better generalization and help narrow the gap between real and generated scenarios. Researchers increasingly rely on diffusion models, generative adversarial networks, and 3D simulation tools to create visual representations of flames, smoke, and complex fire events in industrial environments. Such approaches enable the modeling of rare or hazardous situations that would otherwise be impossible to capture, allowing for safer testing and more robust training of fire detection models.

This dissertation addresses the detection of indoor fires in industrial settings by proposing a deep learning method based on semantic segmentation for the precise extraction of flame regions. Additionally, it examines the application of synthetic data as a substitute for scarce real-world data, with a focus on maintaining or even enhancing the performance of deep neural networks. The dissertation makes the following original scientific contributions:

1. **A semantic segmentation model of fire based on an ensemble of deep neural networks.** A novel segmentation model for fire detection is presented, which integrates the outputs of five state-of-the-art approaches to achieve improved segmentation accuracy. The evaluation of existing models on a proprietary dataset guided the development of the Feature Merging Model (F2M), designed to enhance both accuracy and reliability by incorporating uncertainty estimation. The application of Monte Carlo dropout during inference resulted in improved performance compared to the individual models.

2. **A publicly available dataset of synthetic images for semantic fire segmentation in industrial environments.** Despite advances in more robust machine learning models, their performance remains constrained by the scarcity of data. To address this issue, a new synthetic dataset called SYN-FIRE has been proposed for detecting indoor fires in industrial environments. This dataset is the first publicly available dataset of its kind, providing researchers with access to structured and annotated synthetic data for the semantic segmentation of fire. A comparison of the SYN-FIRE dataset with existing datasets containing real fire images demonstrated that synthetic images can, to some extent, serve as substitutes for real ones, thereby enabling broader use in model training and evaluation.

3. **Analysis of the impact of the ratio of synthetic to real data on the performance of fire segmentation and detection models.** The presented synthetic dataset requires further evaluation to precisely determine its impact on the performance of image semantic segmentation models. As part of this scientific contribution, the U-Net++ model was trained on a combination of real and synthetic data. Evaluation was conducted on a test subset of real data, with testing thresholds determined based on the best results obtained on the validation subset. The impact of synthetic data was examined through two ablation studies. The first study explored how model performance is affected when varying proportions of real images are substituted with synthetic ones. The second study evaluated the effect of combining synthetic and real data on overall model detection capabilities. The findings indicate that synthetic data can partially replace real data without causing a notable decline in performance, and in cases with smaller datasets, it even led to performance gains.

As a result of the research presented in this doctoral dissertation, four papers were published in international scientific journals (three as first author and one as co-author) and seven papers were presented at international scientific conferences (including one as first author).

Keywords: Deep Learning, Fire Dataset, Fire Detection, Industrial Fire, Synthetic Data.

Sažetak

Segmentacija i detekcija unutarnjih požara korištenjem metoda dubokog učenja

Vatra je kemijski proces brze oksidacije gorive tvari pri kojem se oslobađa toplina i svjetlost, a često i različiti plinoviti produkti sagorijevanja. S fizikalnog stajališta, vatra predstavlja vidljivu manifestaciju prijenosa energije u obliku elektromagnetskog zračenja i topline nastale tijekom tog egzotermnog procesa. Požari se dijele na vanjske i unutarnje, a razlikuju se prema prostornom okruženju u kojem nastaju. Unutarnji požari razvijaju se u zatvorenim prostorima (uredski prostori, skladišta, tvornice), dok se vanjski požari razvijaju na otvorenim površinama (livade, šume, građevinski objekti izvana). Požari se razlikuju i prema dinamici širenja požara (kod unutarnjih požara dominira brzo nakupljanje topline i dima, dok kod vanjskih požara veliku ulogu imaju vremenski uvjeti i topografija terena) i pristupu gašenja (unutarnji zahtijevaju ulazak u objekt i kontrolu ventilacije, a vanjski zahtijevaju korištenje širokih fronti i zračnih snaga). U 2023. godini vatrogasne postrojbe u Sjedinjenim Američkim Državama (SAD) intervenirale su na više od 1,39 milijuna požara, pri čemu je smrtno stradalo 3.670 civila, a njih 13.350 zadobilo je ozljede. Osim toga, požari su prouzročili izravnu materijalnu štetu procijenjenu na 23 milijarde američkih dolara. Rana detekcija i prevencija požara od presudne su važnosti jer omogućuju pravovremenu kontrolu širenja, smanjenje ugroze ljudskih života te minimiziranje materijalne štete. Postoje dvije metode detekcije požara. Prva se temelji na senzorima i pripada tradicionalnom pristupu, dok druga koristi kamere i obradu slike kao suvremeni oblik praćenja. Tradicionalni senzorski sustavi prate fizičke parametre, poput temperature, vlage, tlaka i koncentracije toksičnih plinova, no oni su spori i često ne uspijevaju otkriti požar u njegovoj ranoj fazi, a ponekad ga uopće ne registriraju. Metode zasnovane na obradi slike pokazale su se znatno učinkovitijima jer pružaju informacije o prostoru u kojem požar nastaje, o njegovoj veličini, brzini širenja i preciznoj lokaciji. Osim same detekcije, takvi sustavi mogu prepoznati i rizične situacije koje prethode izbijanju požara, čime doprinose njegovoj prevenciji. U prvim istraživanjima koristile su se klasične tehnike računalnog vida, poput detekcije kontura ili grupiranja metodom K-srednjih vrijednosti, no rezultati su često bili nepouzdani i sporiji od senzorskih sustava. Razvoj dubokog učenja u području računalnog vida omogućio je značajan iskorak. Duboke neuronske mreže samostalno uče reprezentacije i izdvajaju značajke iz vizualnih podataka, čime se uklanja potreba za ručnim oblikovanjem značajki koje je bilo ključno u klasičnim

metodama. Sustavi tako prepoznaju hijerarhijske obrasce, od jednostavnih rubova i tekstura do složenih objekata i scena, što ih čini iznimno djelotvornima u prepoznavanju, segmentaciji i klasifikaciji. U ovoj doktorskoj disertaciji naglasak je na primjeni semantičke segmentacije koja omogućuje precizno izdvajanje područja plamena i dima unutar slike, što predstavlja ključan korak u pouzdanoj detekciji unutarnjih požara. Zbog toga se duboke neuronske mreže sve češće primjenjuju u detekciji vatre na slikama i videozapisima te postižu veću brzinu i pouzdanost u odnosu na ranije pristupe. Unatoč ostvarenim napredcima, detekcija požara u zatvorenim prostorima i dalje je posebno zahtjevna. Unutarnji požari u industrijskim okruženima rijetko se događaju, otežano se dokumentiraju zbog opasnosti i nedostupnosti lokacija, a njihovo prepoznavanje dodatno kompliciraju vizualne sličnosti sa svjetlosnim odsjajima i drugim smetnjama. Zbog toga se ova doktorska disertacija usmjerava upravo na problem detekcije unutarnjih požara u industrijskim okruženjima te istražuje kako primjena dubokog učenja može doprinijeti prevladavanju navedenih izazova. Unatoč brzom napretku u području dubokog učenja i razvoju grafičkih procesorskih jedinica (GPU-a), istraživači se i dalje suočavaju s brojnim izazovima u razvoju modela za detekciju vatre. Najveći izazov pritom ostaje nedostatak visokokvalitetnih skupova podataka, neovisno o rastućoj složenosti i učinkovitosti modela. Dosadašnja su istraživanja uglavnom bila usmjerena na vanjske požare, osobito šumske, što je rezultiralo skupovima podataka primarno prilagođenima takvim scenarijima. Nasuprot tome, istraživanja usmjerena na požare u zatvorenim prostorima znatno su rjeđa, ponajviše zbog izrazite oskudice specijaliziranih skupova podataka. Za razliku od vanjskih požara, za koje postoje opsežne baze slika i videozapisa, stvarne podatke o požarima u zatvorenim prostorima izrazito je teško prikupiti, budući da su takvi incidenti rijetki, opasni za dokumentiranje te se često odvijaju u privatnim ili ograničenim prostorima, što otežava njihovu dostupnost i anotaciju. Dodatni izazov predstavlja samo anotiranje slika za detekciju vatre. Dinamična priroda plamena i dima, s nepravilnim i stalno promjenjivim oblicima, otežava precizno označavanje konvencionalnim tehnikama. Složenost dodatno povećava činjenica da vizualne karakteristike vatre često nalikuju svjetlosnim odsjajima, refleksijama ili drugim vizualnim šumovima u okolini, što povećava vjerojatnost pogrešne anotacije i posljedično narušava kvalitetu modela strojnog učenja. Sintetički podaci pružaju učinkovit način za prevladavanje ograničenih skupova podataka u razvoju modela dubokog učenja za detekciju požara u industrijskim prostorima, a stvaraju se računalnim algoritmima i simulacijama te su posebno vrijedni kada su stvarni podaci oskudni, skupi za označavanje ili pristrani. Ako sintetičke slike sadrže realistične značajke vatre i vizualnu autentičnost, omogućuju modelima bolju generalizaciju i smanjenje razlike između stvarnih i generiranih prikaza. Za takve zadatke istraživači se često oslanjaju na difuzijske modele, generativne suparničke mreže i 3D simulacijski softver, što omogućuje stvaranje prikaza plamena, dima i složenih požarnih scenarija u zatvorenim industrijskim okruženjima. Time se mogu obuhvatiti rijetke ili opasne situacije koje je teško dokumentirati u praksi.

Primjena sintetičkih podataka u ovom kontekstu postaje sve važnija jer omogućuje sigurnija testiranja i robusnije treniranje modela za detekciju unutarnjih požara.

Ova doktorska disertacija usmjerena je na problem detekcije unutarnjih požara u industrijskim okruženjima te predlaže metodu dubokog učenja temeljenu na semantičkoj segmentaciji za precizno izdvajanje područja plamena. Dodatno, istražuje se uporaba sintetičkih podataka, kao alternative stvarnim, kako bi se prevladala njihova oskudica uz naglasak na očuvanju ili poboljšanju performansi dubokih neuronskih mreža. Disertacija donosi sljedeće izvorne znanstvene doprinose:

1. **Model semantičke segmentacije vatre temeljen na ansamblu dubokih neuronskih mreža.** Predstavljena je nova metoda segmentacije slika za prepoznavanja požara koja objedinjuje rezultate pet najsuvremenije modela segmentacije, čime se postiže viša razina preciznosti segmentacije. Provedena je evaluacija postojećih modela na vlastitom skupu podataka, a na temelju dobivenih rezultata razvijen je F2M model s ciljem povećanja točnosti i pouzdanosti kroz procjenu nesigurnosti. Primijenjena je Monte Carlo dropout tehnika, čijom je uporabom ostvareno poboljšanje u odnosu na pojedinačne modele segmentacije.

2. **Javno objavljen podatkovni skup sintetičkih slika za semantičku segmentaciju vatre u industrijskim prostorima.** Unatoč razvoju robusnijih modela strojnog učenja, njihova je izvedba i dalje ograničena nedostatkom podataka. Motivirani ovim problemom, predložen je novi sintetički skup podataka namijenjen detekciji unutarnjih požara u industrijskim okruženjima. Riječ je o prvom javno dostupnom skupu takve vrste, koji istraživačima omogućuje pristup strukturiranim i anotiranim sintetičkim podacima za semantičku segmentaciju slika. Usporedbom SYN-FIRE podatkovnog skupa s postojećim skupovima stvarnih slika požara pokazano je da sintetičke slike u određenoj mjeri mogu zamijeniti stvarne, čime se otvara mogućnost šire primjene u treniranju i evaluaciji modela.

3. **Analiza utjecaja omjera sintetičkih i stvarnih podataka na performanse modela za segmentaciju i detekciju vatre.** Predstavljeni sintetički podatkovni skup potrebno je dodatno evaluirati kako bi se precizno utvrdio njegov utjecaj na performanse modela za semantičku segmentaciju slike. U okviru ovog znanstvenog doprinosa U-Net++ model treniran je na kombinaciji stvarnih i sintetičkih podataka. Evaluacija je provedena na ispitnom podskupu stvarnih podataka, a pragovi korišteni pri testiranju određeni su na temelju najboljih rezultata dobivenih na validacijskom podskupu. Uloga sintetičkih podataka analizirana je kroz dvije ablacijske studije. U prvoj studiji ispitana je povezanost između stvarnih i sintetičkih podataka tako da je određeni udio stvarnih slika zamijenjen generiranim sintetičkim slikama, čime je procijenjen učinak različitih omjera na performanse modela. U drugoj studiji istražen

je doprinos integracije sintetičkih i stvarnih podataka na ukupnu učinkovitost modela. Rezultati ovih analiza pokazali su da sintetički podaci mogu u određenoj mjeri zamijeniti stvarne podatke bez značajnog gubitka performansi modela, dok su se na manjim skupovima podataka performanse modela dodatno poboljšale.

Kao rezultat istraživanja predstavljenog u ovom doktorskom radu, objavljeno je četiri rada u međunarodnim znanstvenim časopisima (tri kao prvi autor i jedan kao drugi) i sedam radova na međunarodnim znanstvenim skupovima (od kojih jedan kao prvi autor).

Ključne riječi: Detekcija vatre, Duboko učenje, Industrijski požari, Podatkovni skupovi za vatru, Sintetički podaci

Contents

Abstract **4**

Sažetak **8**

List of Figures **4**

List of Tables **7**

1 Introduction **1**
- 1.1 Contributions 2
- 1.2 Publications 4
  - 1.2.1 Publications in Scientific Journals 4
  - 1.2.2 Publications in Scientific Conferences 4
- 1.3 Organization of the Thesis 5

2 Background and Existing Approaches to Fire Detection **7**
- 2.1 Fundamentals of Fire Detection 8
  - 2.1.1 Physical Characteristics of Fire 8
  - 2.1.2 Fire Detection Using Sensor-Based Methods 10
  - 2.1.3 Fire Detection Using Image-Based Methods 16
- 2.2 Traditional Computer Vision Techniques for Fire Detection 18
  - 2.2.1 Color-Based Thresholding 18
  - 2.2.2 K-Means Clustering 20
  - 2.2.3 Watershed Segmentation 22
  - 2.2.4 Contour Detection 23
- 2.3 Deep Learning in Visual Fire Detection 25
  - 2.3.1 Image Classification 25
  - 2.3.2 Object Detection 26
  - 2.3.3 Image Segmentation 27
  - 2.3.4 Applications of Deep Learning in Fire Detection 29
- 2.4 CNN Architectures for Image Segmentation 29
  - 2.4.1 Feature Pyramid Network (FPN) 29

2.4.2 U-Net 31
2.4.3 U-Net++ 32
2.4.4 MANet 33
2.4.5 DeepLabV3+ 34
2.4.6 SegFormer 36
2.5 Datasets and Dataset Creation Challenges 37
2.5.1 Image Annotation 38
2.5.2 Datasets for Fire Detection 41

3 Ensemble-Based Model for Indoor Fire Detection with Uncertainty Estimation 44
3.1 Research Objectives 44
3.2 Fire Segmentation Model Benchmark 45
3.2.1 Utilized Dataset 46
3.2.2 Dataset preparation 47
3.2.3 Model Training 50
3.2.4 Benchmarking Models 51
3.3 F2M Model 54
3.3.1 Architecture Overview 54
3.3.2 Ensemble Evaluation 56
3.3.3 Discussion 58
3.4 Conclusion 62

4 Impact of Synthetic Data on Fire Segmentation Models 63
4.1 Research Objectives 64
4.2 Existing methods for synthetic data generation 64
4.2.1 Image Generation Using 3D Software 65
4.2.2 Image Generation Using Deep Learning Networks 69
4.3 SYN-FIRE Dataset 72
4.3.1 Environments 72
4.3.2 Dataset Statistics 77
4.4 Implementation Details 78
4.4.1 Experimental Setup 78
4.4.2 Real Data Datasets 79
4.5 Experiments 80
4.5.1 Ablation Study 1 80
4.5.2 Ablation Study 2 80
4.6 Results 82
4.6.1 Corsican Fire Database 83

4.6.2 FLAME Dataset . . . 87
4.6.3 FireBot Dataset . . . 90
4.6.4 BowFire Dataset . . . 92
4.7 Statistical Analysis of Results . . . 93
4.7.1 Statistical Analysis of Ablation Study 1 Results . . . 93
4.7.2 Results For Ablation Study 2 . . . 94
4.8 Model Generalization on Unseen Real-Life Data . . . 96
4.9 Conclusion . . . 98

5 Conclusion **101**

Bibliography **104**

Curriculum vitae **118**

List of Figures

2.1 The figure depicts flame stratification from the fuel-rich core through the reaction zone (2000–2500 K), the continuous flame region (1500–2000 K) with incandescent soot particles, to the pulsating intermittent tip. 9
2.2 A graph illustrating fire phases, damage amount over time, and potential detection methods in each phase. Image source: *Antunovic et al.* [26]. 11
2.3 Classification of traditional and emerging sensor-based fire detection technologies with corresponding detection mechanisms. 12
2.4 The visualization of using color-based thresholding to segment fire. Image source: *Celik et al.* [18]. 19
2.5 The visualization of using K-Means clustering to segment fire. Image source: *Rudz et al.* [59]. 21
2.6 The visualization of using the Watershed algorithm to segment fire. 22
2.7 The visualization of Contour Detection use to detect fire on images. Image source: *Zhang et al.* [67]. 24
2.8 Application of image classification for fire detection, demonstrating the process of analyzing visual input to identify the presence of flames or smoke. 26
2.9 Visualization of object detection showing the detection and localization of fire within images. 27
2.10 Visualization of image segmentation used to segment fire regions within images. 28
2.11 The structure of FPN architecture. Image source: *Lin et al.* [92]. 30
2.12 The structure of U-Net architecture. Image source: *Ronneberger et al.* [93]. . 31
2.13 The structure of U-Net++ architecture. Image source: *Zhou et al.* [94]. . . . 33
2.14 The structure of MANet architecture. Image source: *Rui et al.* [95]. 34
2.15 The structure of DeepLabV3+ architecture. Image source: *Chen et al.* [96]. . 35
2.16 The structure of SegFormer architecture. Image source: *Xie et al.* [97]. . . . 36
2.17 Evaluated datasets and examples of input RGB images. 43

3.1 Flowchart showing the study workflow: data preparation, model development, performance evaluation, and final assembly with uncertainty analysis. Image source: *Arlovic et al.* [11]. 46

3.2 The visualization of RGB images in the utilized dataset. Image source: *Arlovic et al.* [11]. 49
3.3 The architecture of the F2M network for an image size of 256x256 pixels, with layers adjusted for different resolutions. Image source: *Arlovic et al.* [11]. . . 55
3.4 Dice score progression during the validation phase for F2M and the Feature Extraction Branch at a resolution of 640×640. The dark blue line represents the smoothed outcome (60% smoothing), whereas the light blue line illustrates the recorded metrics. Image source: *Arlovic et al.* [11]. 59
3.5 Graph showing the Monte Carlo dropout uncertainty across 100 iterations of the same input image. Image source: *Arlovic et al.* [11]. 60
3.6 The visualization of fire detection results illustrates the RGB input image, the ground truth mask, the predicted mask at 33% threshold, the standard deviation (STD) mask derived via the Monte Carlo dropout uncertainty estimate, and the final amalgamated mask. The STD mask delineates regions of ambiguity in the forecasts. Image source: *Arlovic et al.* [11]. . . . 61

4.1 Comparison of rendered scenes in NVIDIA Omniverse (top row) and Unreal Engine 5.3 (bottom row). Image source: *Arlovic et al.* [12]. 65
4.2 Qualitative comparison of images generated by three diffusion-based models: DALL·E 3, SDXL 1.0, and ChatGPT 4o. Image source: *Arlovic et al.* [12]. . 71
4.3 Synthetic image samples for the Modular Warehouse environment. 74
4.4 Synthetic image samples for the Sorting Warehouse With Conveyor Belts & Robots environment. 74
4.5 Synthetic image samples and their annotations for the Garage environment. . 75
4.6 Synthetic image samples and their annotations for the Warehouse With Various Sections environment. 76
4.7 Synthetic image samples and their annotations for the Metro Maintenance Station environment. 77
4.8 The flowchart depicts the process of data allocation for the training subset in the first ablation study when the $\alpha = 0.2$. Image source: *Arlovic et al.* [5]. . 81
4.9 The flowchart depicts the process of data allocation for the training subset in the second ablation study when the $\beta = 0.1$. Image source: *Arlovic et al.* [5]. 81
4.10 The U-Net++ model performance is demonstrated on four randomly selected samples from the BowFire dataset. Image source: *Arlovic et al.* [5]. 82
4.11 Model evaluation on the test subset using the SYN-FIRE and Corsican FireDB datasets. (a) Dice Score when real data is substituted with synthetic data, and (b) Dice Score when synthetic data is incorporated with real data. Image source: *Arlovic et al.* [5]. 91

4.12 Evaluation of the model on the test subset utilizing the SYN-FIRE and Flame datasets. (a) Dice Score when actual data is substituted with synthetic data, and (b) Dice Score when synthetic data is integrated with real data. Image source: *Arlovic et al.* [5]. 91
4.13 Evaluation of the model on the test subset utilizing the SYN-FIRE and FireBot datasets. (a) Dice Score when actual data is substituted with synthetic data, and (b) Dice Score with the integration of synthetic data with real data. Image source: *Arlovic et al.* [5]. 92
4.14 Evaluation of models on the test subset, trained using the SYN-FIRE and BowFire datasets. (a) Displays the Dice Score when real data is replaced with synthetic data, and (b) Displays the Dice Score when integrating synthetic data alongside real data. Image source: *Arlovic et al.* [5]. 92
4.15 The paired T-Test statistical analysis of Dice Score changes in Ablation Study 1. 95
4.16 The paired T-Test statistical analysis of Dice Score changes in Ablation Study 2. 96
4.17 Evaluation of the model's performance using 40 real-world scenario images left out from the training dataset. 98
4.18 Comparison of original images and Grad-CAM overlays showing model attention on fire regions. 100

List of Tables

3.1 Dataset statistics grouped by metadata collected from image annotation. . . 47
3.2 Benchmark results of the performance of each tested model on the validation and test datasets. 53
3.3 The F2M model performance comparison against the best benchmarked model for each resolution. 56
3.4 Comparison of model performance: Average Inference Time and Number of Parameters per sample. 57
3.5 Importance of Bias branch: Results of F2M with and without Bias branch . 58

4.1 Statistics for the SYN-FIRE dataset, highlighting the number of images per scenario. 78
4.2 Statistics of real data datasets that utilized in the experiment. 79
4.3 Results of training the model on the validation and test subsets of the Corsican Fire Database dataset. The underlined results represent the baseline results from the model trained with $\alpha = 1$. 83
4.4 Results of training the model on the validation and test subsets of the FLAME dataset. The underlined results represent the baseline results from the model trained with $\alpha = 1$. 84
4.5 Results of training the model on the validation and test subsets of the FireBot dataset. The underlined results represent the baseline results from the model trained with $\alpha = 1$. 85
4.6 Results of training the model on the validation and test subsets of the BowFire dataset. The underlined results represent the baseline results from the model trained with $\alpha = 1$. 86
4.7 Results of training the model on the validation and test subsets of the Corsican Fire Database dataset. The underlined results represent the baseline results from the model trained with $\alpha = 1$. 87
4.8 Results of training the model on the validation and test subsets of the FLAME dataset. The underlined results represent the baseline results from the model trained with $\alpha = 1$. 88

4.9 Results of training the model on the validation and test subsets of the FireBot dataset. The underlined results represent the baseline results from the model trained with $\alpha = 1$. 89
4.10 Results of training the model on the validation and test subsets of the BowFire dataset. The underlined results represent the baseline results from the model trained with $\alpha = 1$. 90

1

Introduction

Fire is a highly complex natural event that poses a serious hazard, especially in densely populated areas, where it can cause severe property damage and loss of life. Fire has been studied extensively across scientific disciplines, including physics, chemistry, ecology, and computer science [1]. As urban environments grow denser and the vision of smart cities becomes reality, the demand for dependable, intelligent fire detection systems continues to increase [2]. This demand is particularly vital in indoor environments such as residential buildings, hospitals, schools, and commercial spaces, where rapid and accurate fire detection can be the difference between a minor incident and a devastating tragedy. At its core, fire is a chemical reaction in which fuel is rapidly oxidized in the presence of heat and oxygen, releasing energy in the form of light and heat[1]. The combustion process produces characteristic flames whose color, intensity, and behavior depend on the chemical composition of the burning material and the availability of oxygen [3]. Understanding fire from a scientific perspective is inherently multidisciplinary, namely physics, chemistry, and engineering. However, the practical challenge of detecting and responding to unwanted fires before they cause catastrophic harm falls increasingly within the domain of computer science and artificial intelligence. The difficulty of achieving reliable early detection has driven extensive research into fire detection systems, which are commonly categorized into sensor-based and image-based approaches [4].

Traditional, sensor-based fire detection relies on sensors that respond to smoke particles, elevated temperatures, or combustion gases. These systems have served well in many contexts, particularly in enclosed spaces where smoke can accumulate and trigger alarms. However, they have inherent limitations. Environmental factors can further degrade performance. Dust, humidity, high ceilings, and strong air currents all affect how quickly smoke or heat reaches a sensor. Additionally, certain harmless aerosols, such as those from cooking or steam, can trigger false alarms, eroding trust in these systems over time. In large or open environments, such as industrial facilities and outdoor settings, conventional sensors may fail to detect fires until they have grown substantially. Indoor environments present unique challenges, including variable lighting conditions, cluttered backgrounds, reflective surfaces, and objects that may visually resemble flames, such as candles, lamps, or television screens displaying fire imagery.

In contrast to sensor-based systems, image-based fire detection offers a compelling alternative that addresses many of the above shortcomings. Cameras, whether operating in the visible spectrum or capturing infrared radiation, can identify flames and smoke from considerable distances and at very early stages. Flames produce characteristic color patterns, typically in the yellow, orange, and red ranges, and emit thermal radiation that infrared cameras can detect even when visible flames are obscured. The flickering dynamics of fire provide another discriminating feature that helps distinguish genuine combustion from static light sources or reflections. By analyzing these visual and temporal features, image-based systems can detect fires sooner than traditional sensors, work effectively in both indoor and outdoor environments, and provide richer contextual information about the nature and extent of an incident.

With the development of artificial intelligence, researchers have investigated a variety of algorithms to detect fire more effectively [5]. In general, computer vision approaches for fire detection can be categorized into three main areas: image classification, object detection, and image segmentation. Image classification determines whether fire is present in an image, object detection localizes fire using bounding boxes, and image segmentation provides a pixel-level representation of the fire region. While classification and detection can deliver useful results, they are limited in their ability to capture the exact extent of fire. For this reason, image segmentation is adopted in this study, as it enables more precise localization of fire boundaries and progression, which is essential for reliable monitoring and timely response. Image segmentation is a common approach utilized across diverse domains, including satellite and medical image analysis [6]. It plays a vital role in tasks such as facial feature recognition [7, 8] and text recognition [9], highlighting its versatility in computer vision and pattern recognition. Moreover, it can be used to efficiently detect and track fires [10].

1.1 Contributions

A semantic segmentation model of fire based on an ensemble of deep neural networks.

We propose a F2M model that combines the outputs of five state-of-the-art models to produce more precise results for fire detection tasks. Following the evaluation of existing segmentation models on the newly created dataset, we designed the F2M model to improve performance and uncertainty estimation. This was achieved by using the Monte Carlo dropout which led to improved performance compared to the individual segmentation models. Our method outperforms the best CNN and U-Net++, offering superior segmentation results and reliability, and setting a new standard for fire detection systems. This contribution is explained in more detail in Chapter 3 and has been published in a journal paper [11].

A publicly available dataset of synthetic images for semantic fire segmentation in industrial environments.

Despite advancements in model complexity and performance, a significant challenge in developing deep learning models for fire detection remains the scarcity of high-quality datasets. Most existing research has focused on outdoor fires, particularly wildfires, which has led to the development of datasets primarily tailored to those scenarios. Research on indoor fires has received limited attention, as datasets specifically designed for such environments remain extremely scarce. In contrast to outdoor fires, where large-scale image and video datasets are available, real-world indoor fire data are difficult to obtain because incidents are infrequent, dangerous to document, and often take place in private or restricted areas that limit accessibility and documentation. Consequently, synthetic and simulated data have emerged as a practical alternative, made possible by recent advances in graphics and simulation technologies. Building on this approach, we present the *SYN-FIRE dataset*, the first synthetic dataset specifically developed to help detect indoor fire in industrial environments. By making this resource publicly available, our work directly addresses the longstanding scarcity of indoor fire data and establishes a foundation for future research in this underrepresented domain. This contribution is further detailed in Chapter 4 and has been published in a journal paper [5] and a conference paper [12].

Analysis of the impact of the ratio of synthetic to real data on the performance of fire segmentation and detection models.

To further evaluate and expand on the previous contribution, the U-Net++ model was trained using both real and synthetic datasets to evaluate the impact of synthetic data on the performance of the trained model. Segmentation models were trained using publicly available datasets of real fire images and the newly introduced SYN-FIRE dataset of synthetic fire images. The results were obtained from a test subset of real datasets, using thresholds that yielded the best performance on the validation subset. We conducted two distinct ablation studies to analyze the impact of synthetic data on model performance. In the first ablation study, we examined the influence of the correlation between synthetic and real data on model performance by substituting a portion of real data with generated synthetic images. The second ablation study evaluated the impact of integrating synthetic with real data on overall model performance. The contribution is detailed in Chapter 4 and has been published in a journal paper [5].

1.2 Publications

1.2.1 Publications in Scientific Journals

1. Antonio Antunović, Matej Arlović, Josip Balen, and Ljiljana Šerić. "Advances in Fire Detection and Suppression: A Review of Contemporary Methods and Technologies". *IEEE Access*, doi:10.1109/ACCESS.2025.3638631

2. Matej Arlović, Franko Hržić, Mitesh Patel, Tomasz Bednarz, and Josip Balen. "Evaluation of synthetic data impact on fire segmentation models performance". *Scientific reports*, 15/1 (2025), 16759, 14. doi:10.1038/s41598-025-01571-5

3. Matej Arlović, Mitesh Patel, Josip Balen, and Franko Hržić. "F2M: Ensemble-Based Uncertainty Estimation Model for Fire Detection in Indoor Environments". *Engineering applications of artificial intelligence*, 133 (2024), 108428, 12. doi:10.1016/j.engappai.2024.108428

4. Matej Arlović, Tomislav Rudec, Josip Miletić, and Josip Balen. "Proposal of OptDG Algorithm for Solving the Knapsack Problem". *International journal of advanced computer science & applications*, 15 (2024), 9, 98, 7. doi:10.14569/ijacsa.2024.0150998

1.2.2 Publications in Scientific Conferences

1. Antonio Antunovic, Davor Damjanovic, Matej Arlovic, Emmanuel Karlo Nyarko, Franko Hrzic, and Josip Balen. "Mutual-Training Pseudo-labeling Framework for Fire Segmentation". *Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA 2025)*. Coimbra, Portugal: Springer, 2025. str. 133-147. doi:10.1007/978-3-031-99565-1_11

2. Josip Balen, Antonio Antunovic, Matej Arlovic, Davor Damjanovic, Edvin Borovac, Dunja Caleta, and Gabrijel Krilcic. "Intelligent fire safety in indoor environments: Technologies for fire prevention, detection, and suppression". *12 th International New York Conference on Evolving Trends in Interdisciplinary Research & Practices*. Manhattan, New York City: New York (NY): Liberty Academic Publishers, 2025. str. 349-358.

3. Matej Arlović, Davor Damjanović, Franko Hržić, and Josip Balen. "Synthetic Dataset Generation Methods for Computer Vision Application". *International Conference on Smart Systems and Technologies (SST 2024)*. Osijek, Croatia: IEEE, 2024. str. 69-74. doi:10.1109/sst61991.2024.10755475

4. Josip Balen, Davor Damjanović, Petar Marić, Krešimir Vdovjak, Matej Arlović, Goran Martinović. "FireBot - An Autonomous Surveillance Robot for Fire Prevention, Early Detection and Extinguishing". *15th International Conference on Computer and Automation Engineering (ICCAE)*. Sydney, Australia: IEEE, 2023. str. 400-405. doi:10.1109/iccae56788.2023.10111251

5. Petar Marić, Franko Hržić, Matej Arlović, Mitesh Patel, Muhammad Azeem Moazam, and Josip Balen. "YOLOv5: Case Study of Image Resolution Influence on Fire Detection". *2023 International Symposium ELMAR*. Zadar, Croatia: IEEE, 2023. str. 13-18. doi:10.1109/ELMAR59410.2023.10253913

6. Petar Marić, Matej Arlović, Josip Balen, Krešimir Vdovjak, and Davor Damjanović. "A Large Scale Dataset For Fire Detection and Segmentation in Indoor Spaces". Malé, Maldives: IEEE, 2022. str. 1-8. doi:10.1109/ICECCME55909.2022.9987926

7. Krešimir Vdovjak, Petar Marić, Josip Balen, Ratko Grbić, Davor Damjanović, and Matej Arlović. "Modern CNNs Comparison for Fire Detection in RGB Images". *18th International Conference on Machine Learning and Data Mining (MLDM 2022)*. New York, USA: Springer, 2022. str. 239-254

1.3 Organization of the Thesis

This thesis focuses on fire detection and segmentation in indoor environments. It is structured into five primary chapters. After the introduction, a background and related work chapter reviews existing research. The subsequent chapters detail the principal contributions, including the proposed methods and results. The concluding chapter summarizes the findings and outlines potential directions for future research.

CHAPTER 2: BACKGROUND AND EXISTING APPROACHES TO FIRE DETECTION

This chapter presents an overview of the background and current approaches used in fire detection. The chapter starts with a description of fire characteristics and their application in various detection systems, encompassing both sensor-based and image-based methods. The chapter continues with a summary of traditional computer vision techniques for detecting fire in images, followed by a discussion of deep learning methods that learn directly from visual data. It also outlines challenges in dataset creation, especially the difficulty of collecting diverse and realistic fire examples. The chapter concludes with a review of deep learning models for image segmentation, focusing on architectures commonly used to locate fire regions in visual scenes.

CHAPTER 3: ENSEMBLE-BASED MODEL FOR INDOOR FIRE DETECTION WITH UNCERTAINTY ESTIMATION

This chapter presents an ensemble-based model for indoor fire detection with uncertainty estimation. We introduce the F2M model, which incorporates the fire segmentation outputs from the five best-performing models identified through model benchmarking that we previously conducted in that research. The F2M combines these outputs to produce a single, more reliable segmentation mask and uses Monte Carlo dropout to estimate prediction uncertainty. Although the improvements are relatively small, the model exhibits slightly better performance than the strongest individual baseline, U-Net++, across all three tested image resolutions. Evaluation is based on total error, Dice Score, and IoU score. This approach aims to enhance model robustness and provide uncertainty estimates that can support more informed decision-making in fire detection systems. This chapter also discusses the design of the ensemble mechanism and its role in improving segmentation consistency across varying image conditions.

CHAPTER 4: IMPACT OF SYNTHETIC DATA ON FIRE SEGMENTATION MODELS

This chapter presents SYN-FIRE, the first synthetic dataset of indoor fires in industrial environments, which includes pixel-level annotations and fire images generated with NVIDIA Omniverse. Afterward, we evaluated the impact of integrating synthetic and real data on the performance of fire segmentation models through benchmark evaluations of publicly available fire datasets. Our findings emphasize the significant potential of synthetic data in enhancing the performance of deep learning models, particularly in scenarios with scarce real-world data like fire detection.

CHAPTER 5: CONCLUSION

The final chapter summarizes the main findings of the research and highlights the contributions made in relation to the research objectives. It considers the significance of the results within the broader field of fire detection research and outlines potential directions for future research.

2

Background and Existing Approaches to Fire Detection

This chapter provides an overview of existing fire detection approaches, with a focus on methods relevant to fire detection in indoor environments. Fire represents a serious hazard in densely populated urban areas, where incidents can result in significant property damage, economic losses, and loss of human life. The rapid expansion of modern cities and the increasing complexity of indoor spaces further amplify these risks, underscoring the need for reliable and timely fire detection systems. Fire detection is a complex task that requires an understanding of both the physical behavior of fire and the technologies used to detect it. Conventional fire detection systems typically rely on smoke or heat sensors, which often exhibit slow response times and provide limited spatial information. In contrast, image-based methods enable earlier detection by identifying small or emerging flames and offer richer information, including precise fire localization, severity estimation, and monitoring fire propagation.

This chapter is structured as follows. Section 2.1 describes the fundamental characteristics of fire, including flame, smoke, heat, and combustion gases, and discusses their influence on fire detection strategies. Section 2.2 presents classical computer vision methods for fire detection based on handcrafted features such as color, intensity, motion, and region boundaries. Section 2.3 introduces deep learning approaches that learn features directly from data and provide improved robustness and performance. Section 2.4 reviews state-of-the-art convolutional neural network (CNN) architectures for fire segmentation. Finally, Section 2.5 discusses challenges related to dataset creation and annotation, with particular emphasis on the scarcity of indoor fire data and the dynamic nature of fire and smoke.

2.1 Fundamentals of Fire Detection

Understanding the fundamentals of fire detection is crucial for ensuring an early warning and an effective response to fire emergencies. This section examines the fundamental principles, technologies, and system components that contribute to detecting fires and threats that can lead to fire.

2.1.1 Physical Characteristics of Fire

Fire is a complex phenomenon composed of several key components: flame, heat, smoke, and combustion gases. The visible flame and smoke are the most recognizable elements, resulting from an exothermic chemical reaction between a fuel source and an oxidizer (typically oxygen) [13]. A flame is a chemical reaction that generates a temperature of at least 1500 K in general, and a maximum of 2500 K in air. From a physical perspective, fire can be described as a complex flow field composed of interacting flames or flamelets [14]. In addition to flames, smoke forms as a byproduct of incomplete combustion and contains a mixture of fine particulate matter (soot), water vapor, and toxic gases such as carbon monoxide (CO), hydrogen cyanide (HCN), and volatile organic compounds (VOCs), all of which depend on the burning materials and fire stage [15]. In indoor environments, such as industrial setting, fire behavior differs significantly from open-air combustion. One of the characteristics of fire is the development of a fire plume, a vertical column composed of hot gases, combustion byproducts, and ambient air that rises from the flame [16]. The heat release rate of the fire and the ambient ventilation determine the temperature, velocity, and turbulence of the flame. A ceiling jet flame is specifically formed when a fire plume expands laterally upon reaching the ceiling. This ceiling jet produces an intense heat discharge from the ceiling, which can have a substantial impact on combustible materials in the proximity. These effects have the potential to initiate secondary fires and inflict damage on noncombustible materials [17]. In parallel, this process also leads to thermal stratification, where hot gases accumulate near the ceiling, and cooler air remains below. The stratified layers descend as the fire intensifies, causing a decrease in visibility and an increase in toxicity for the occupants [13].

Spatial and thermal dynamics are essential for understanding the process by which fire spreads indoors and for developing reliable detection systems. Fire exhibits not only physical and thermal indicators but also distinct visual features that can be detected through image-based methods. Although these systems can be challenging to implement, the visual nature of fire provides useful context that supports effective detection. These include dynamic motion patterns, irregular contours, flickering behavior, and color distributions typically within the red, orange, and yellow spectrum [18, 19]. Fire and smoke can have different shapes, colors, and levels of transparency, all of which may alter the area that is detected [19]. Flame color varies significantly depending on the combustion efficiency, fuel type, and oxygen availability. The color of a flame depends on several factors, including its temperature, the chemical composition of the fuel, the presence of soot particles, and the amount of available oxygen during combustion. Incomplete combustion often produces yellow, orange, or red flames due to the incandescence of soot particles. In contrast, complete combustion in well-ventilated conditions can result in blue flames, indicating higher temperatures and cleaner burning [13]. The structural composition of a flame and the corresponding temperature zones in a diffusion flame are illustrated in Figure 2.1.

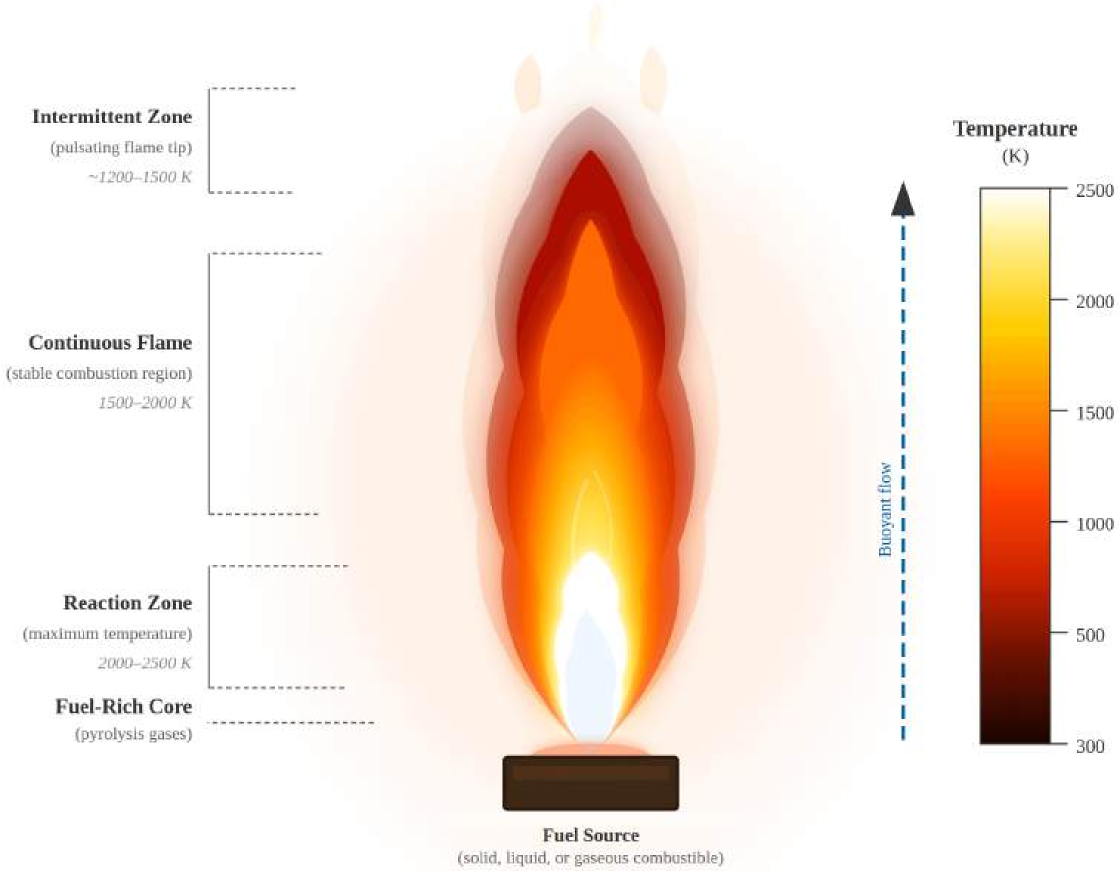


Figure 2.1: The figure depicts flame stratification from the fuel-rich core through the reaction zone (2000–2500 K), the continuous flame region (1500–2000 K) with incandescent soot particles, to the pulsating intermittent tip.

The temporal flickering of flames, usually oscillating between 8 and 12 Hz, stems from turbulent combustion and buoyant vortex shedding and can be used as a temporal signature for fire [19]. Fire regions also tend to be highly luminous, with chaotic, non-rigid shapes that vary from frame to frame, which helps differentiate them from static light sources or fixed geometric objects. However, the visual features typically used to detect fire, such as color and intensity variations, can also be produced by non-fire sources. Common examples include vehicle headlights, sunlight reflections on shiny surfaces, or colored lighting in indoor environments. These conditions can closely resemble flame-like patterns in both color distribution and dynamic behavior, which often mislead computer vision algorithms. This often results in false positive detections, which reduces the reliability of computer vision models in real-world conditions where such visual distractions are common.

Although these visual and physical characteristics are shared across environments, significant differences exist between indoor and outdoor fires that affect both their behavior and detectability. Detecting fire in indoor environments presents a unique set of challenges due to the nature of enclosed spaces. Restricted airflow in such environments leads to the rapid accumulation of smoke, and thermal stratification, all of which influence how heat and combustion gases disperse throughout the space [13, 20]. These conditions reduce visibility, increase toxicity for occupants, and may delay the activation of detection and suppression systems [21]. Alongside these thermal effects, indoor visual fire detection becomes more difficult due to environmental artifacts. Reflections from glossy materials such as polished floors, windows, or metal surfaces can create flame-like visual patterns that resemble the color and flickering motion of real fire [22]. Insufficient ventilation increases smoke accumulation, which obscures flame contours and distorts critical features required for accurate segmentation. Artificial light sources, such as incandescent bulbs, LED indicators, or flickering monitors, often emit warm color tones and fluctuating brightness levels that overlap with the visual appearance of flames, increasing the likelihood of false positives [19]. In contrast, outdoor fires occur in open-air environments where abundant oxygen and wind facilitate faster flame spread and more pronounced convective behavior [23]. These fires generally lack the thermal layering observed indoors, but their detection introduces a different set of visual complexities. Outdoor scenes are affected by background clutter, natural occlusions such as vegetation, and highly variable lighting conditions caused by sunlight, shadows, or atmospheric effects [22]. Hence, in order to ensure the reliability of fire detection, image-based deep learning models must be both adaptable to the contextual distinctions between indoor and outdoor environments and resistant to mentioned domain-specific challenges.

2.1.2 Fire Detection Using Sensor-Based Methods

Traditional sensor-based fire detection methods, such as those relying on smoke, heat, or gas, provide only limited capabilities [24]. To trigger an alarm, a fire must burn long enough for its byproducts to reach the sensor and exceed a predefined threshold. This unavoidable delay is critical, as it allows the fire to escalate and spread rapidly, increasing the potential damage and narrowing the window for effective intervention [25]. In contrast, image-based techniques can detect early visual cues of flames, enabling significantly faster reaction times and greater reliability in hazardous environments. Fire development in indoor environments typically progresses through four phases: incipient, growth, fully developed, and decay. These stages, the detection methods suited to each stage, and the corresponding extent of damage over time are illustrated in Figure 2.2.

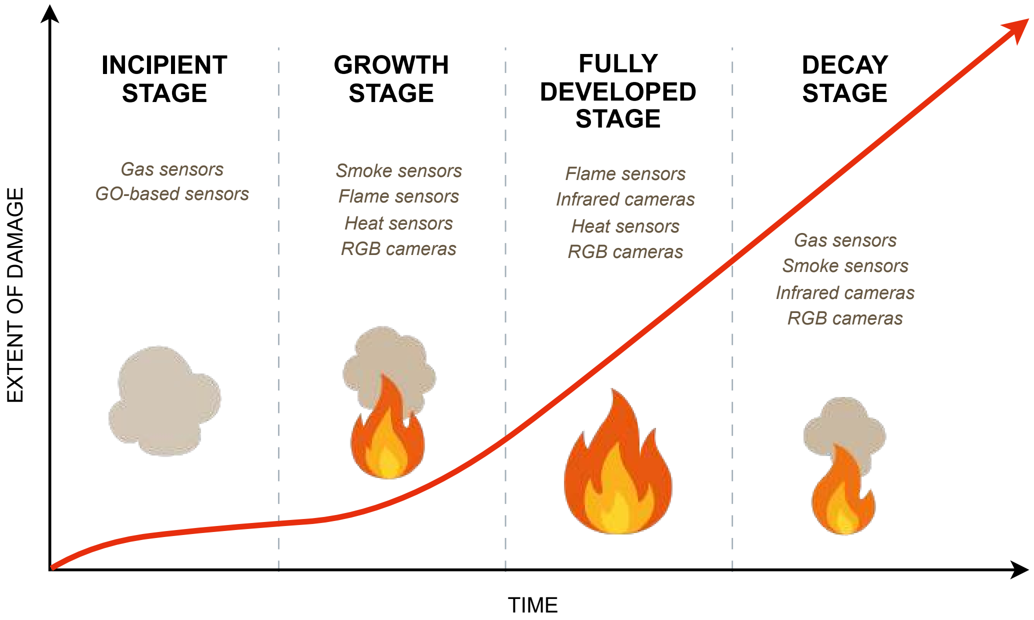


Figure 2.2: A graph illustrating fire phases, damage amount over time, and potential detection methods in each phase. Image source: *Antunovic et al.* [26].

In the incipient and growth stages of a fire, the primary focus must be on the rapid and secure evacuation of all occupants, as protecting human life remains the most critical objective throughout these early periods. Early detection of a structure fire is crucial for safe evacuation and effective fire extinguishing [27]. Due to their limited coverage, conventional sensors are often ineffective in large indoor areas or open environments [27]. These systems primarily rely on physical parameters like air temperature and smoke concentration to identify fire presence. Despite widespread use, sensor-based fire detection remains vulnerable to various environmental conditions. Factors such as installation height, dust accumulation, and airflow can significantly affect sensor performance. These challenges may lead to false alarms, missed alerts, or delayed detection [11]. Due to these delays, emergency response teams may be unable to respond in time to contain fires during their initial phases. Furthermore, the reliability of fire alarm systems is often reduced by the frequent occurrence of false alarms and the challenge of detecting fires in their early stages, particularly when smoke or heat is not yet prominent. Figure 2.3 shows a summary of traditional sensor-based detectors for fire detection.

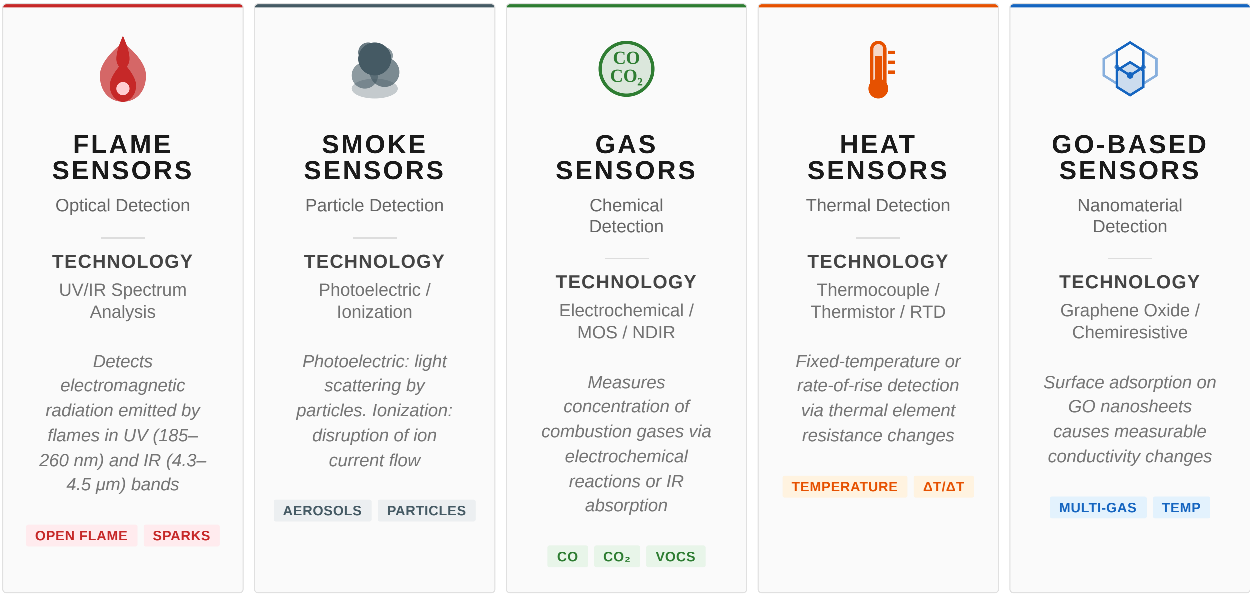

FLAME SENSORS
Optical Detection
TECHNOLOGY
UV/IR Spectrum Analysis
Detects electromagnetic radiation emitted by flames in UV (185–260 nm) and IR (4.3–4.5 μm) bands
OPEN FLAME
SPARKS
SMOKE SENSORS
Particle Detection
TECHNOLOGY
Photoelectric / Ionization
Photoelectric: light scattering by particles. Ionization: disruption of ion current flow
AEROSOLS
PARTICLES
CO
CO2
GAS SENSORS
Chemical Detection
TECHNOLOGY
Electrochemical / MOS / NDIR
Measures concentration of combustion gases via electrochemical reactions or IR absorption
CO
CO₂
VOCS
HEAT SENSORS
Thermal Detection
TECHNOLOGY
Thermocouple / Thermistor / RTD
Fixed-temperature or rate-of-rise detection via thermal element resistance changes
TEMPERATURE
ΔT/ΔT
GO-BASED SENSORS
Nanomaterial Detection
TECHNOLOGY
Graphene Oxide / Chemiresistive
Surface adsorption on GO nanosheets causes measurable conductivity changes
MULTI-GAS
TEMP

Figure 2.3: Classification of traditional and emerging sensor-based fire detection technologies with corresponding detection mechanisms.

Flame Sensors

Heat transfer in combustion process primarily occurs via conduction, convection, and radiation [28]. Due to the high temperatures, thermal radiation plays a significant role in the spread and behavior of fire [13]. The fire itself is a powerful radiation source that enhances heat transfer to adjacent materials, thereby accelerating flame development [29]. A flame is the visible aspect of fire, emerging from an exothermic reaction between a fuel and an oxidant [30]. Its temperature depends on the specific combustible material involved. Flame features two principal characteristics: color, indicating chromatic properties, and the emission of radiation [27]. Flame detectors are advanced sensors that detect fire by analyzing the radiation emitted during combustion. They distinguish the spectral signature of flame radiation from other sources, such as hot surfaces, artificial light, or sunlight [29]. This spectral discrimination is essential for reducing false detections and ensuring accurate flame recognition in diverse environments [29]. The main types of flame detectors are Ultraviolet (UV), Infrared (IR), UV/IR, Multi-Spectrum Infrared (MSIR), and Multispectral Flame Detectors.

Smoke Sensors

Smoke sensors are commonly used in fire detection, as they detect airborne particles produced during the combustion process. The principal factors for smoke detection are smoke particle concentration, volume fraction, and particle size distribution. Smoke detectors must be able to detect combustion and smoke produced by burning flames, as there are significant changes in the structure and content of smoke formed during these types of events [31].

The type, volume, and density of smoke emitted during the fire development process vary considerably depending on the fuel type and the oxygen supply. The peak concentration of visible smoke usually occurs during the incipient and smoldering phases. The accuracy of smoke measurement is closely related to the type of combustion, including pyrolysis, flaming, and smoldering. Smoke sensors are typically divided into two main categories: photoelectric and ionization. The choice of detection method is influenced by the fire's characteristics and the surrounding environment. Photoelectric sensors work based on the principle of light scattering. A signal is initiated when smoke enters the chamber, disrupting the light beam and causing the infrared light to fall below the receiver's threshold. Photoelectric sensors are generally more responsive to smoldering fires and offer quicker response times [27]. Ionization sensors contain a radioactive source that ionizes the air inside the sensing chamber. They react to both visible and invisible products of combustion. When smoke particles are present, they interfere with the ionized particles, reducing the flow and triggering the alarm [32]. Ionization-type sensors are more effective at detecting flaming fires. Multimodal smoke detectors combine smoke detection with supplementary sensing technologies to improve fire detection accuracy. Using various sensors, these detectors can identify several fire-related attributes, such as smoke, temperature, and toxic gases [33]. Multimodal detectors integrate data from multiple sources to improve fire detection accuracy and reduce false alarms. This approach aims for high sensitivity to smoke without compromising reliability, making it a practical option for diverse environments [27].

Gas Sensors

Gases are released during each combustion stage, and their specific characteristics can be leveraged for fire detection. This aspect becomes particularly critical in modern buildings, where synthetic materials such as plastics, polymers, and foams have largely replaced natural substances like wood and cotton [34]. The combustion of these synthetic materials leads to a more rapid spread of fire and the emission of significantly higher quantities of hazardous fumes and toxic gases, such as HCN and CO. Unlike natural materials, synthetic furnishings, and insulation present greater risks due to the volume and toxicity of their combustion byproducts [34]. Gas sensors are commonly employed to detect these gases effectively, typically falling into two main categories: electrochemical sensors and metal oxide semiconductor (MOS) sensors. Electrochemical gas sensors are highly effective for detecting CO, HCN, nitrogen oxides, and VOCs due to their high specificity and sensitivity. The use of advanced materials, including carbon nanostructures, noble metal catalysts, and metal-organic frameworks, has enhanced performance by improving sensitivity and decreasing response time [34]. MOS gas sensors are valued for their high sensitivity and low cost. Their sensing mechanism is based on resistance changes that occur when chemical reactions between target gases and oxygen ions adsorbed on the MOS surface take place [27].

Heat Detectors

Heat is a form of thermal energy that travels from hotter to cooler environments [31]. Fire detection uses heat sensors and changes in ambient temperature to detect fire inside environments. They are alternatives to the smoke detectors in environments where it is normal to have smoke in the space due to working conditions (e.g., machine rooms, concert rooms, steel mills, etc.). A heat sensor consists primarily of a signal conditioning circuit, an amplification circuit, and a thermal element. The thermal element senses the temperature corresponding to the resistance variation, refractive index, displacement, and other factors. The heat sensor evaluates the indoor building temperature [35]. Based on their functionality, heat sensors can be classified as fixed-temperature or rate-of-rise detectors.

A fixed-temperature heat detector is designed to trigger an alarm when the temperature exceeds a predetermined threshold value. Various forms of fixed-temperature sensors exist, including distributed fiber optics, fuse elements, and bimetals [27]. The fuse-element heat detector is mainly used in the fire sprinkler system and operates at a predetermined temperature level dependent on the melting of the heating element. They are non-restoring detectors, as the fusible element requires replacement after activation [31]. Bi-metal heat detectors function by the thermal expansion process of the metals. In response to an increase in temperature, the bimetallic strip will flex toward the metal with a low coefficient of thermal expansion [27]. When activated, the heat sensor consistently detects a surrounding temperature exceeding its operational threshold. The disparity between these two temperatures is referred to as thermal lag, which correlates with the increase in temperature [31].

Distributed optical fiber heat sensing represents a promising and increasingly adopted fire detection and thermal monitoring technology. Utilizing distributed temperature sensing (DTS) principles, these systems employ passive fiber-optic cables to deliver precise and continuous temperature measurements along their entire length [27]. This heat-sensing method quantifies the heat along the fiber using the Raman effect. When an optical pulse is sent through the fiber, part of the light is scattered and reflected back to the source. The system determines temperature changes along the cable by analyzing the backscattered signal. The intensity of Raman scattering is directly correlated with temperature, allowing for the precise measurement of heat distribution over large distances. This approach has proven effective in identifying abnormal heat patterns and enabling early fire detection. As a reliable and scalable solution, distributed optical fiber heat sensing is gaining recognition for its effectiveness in fire prevention and safety-critical environments [36]. Distributed optical fiber static heat sensing is widely applied in fire detection across complex and high-risk environments such as tunnels, industrial facilities, electrical substations, and conveyor systems, where continuous and accurate temperature monitoring is essential for early fire detection.

Bi-metal heat sensors operate through thermal expansion, utilizing two metals with different expansion coefficients that combine together. As the temperature increases, the strip curves toward the metal with the lower thermal expansion coefficient. The heat from the surrounding flames induces a bending motion that completes an electrical circuit, activating an alarm. The distance between contacts dictates the activation temperature. Common types include the bimetal strip and the bimetal snap disk.

Rate-of-rise heat detectors monitor the ambient temperature and the speed at which it increases over time. These sensors typically activate when the temperature rises rapidly, often between 12 and 15 degrees Fahrenheit per minute [31]. They are designed to ignore slow or minor fluctuations unless those persist for an extended period. However, their extreme sensitivity to quick environmental changes makes them prone to false alarms. As a result, they are rarely used in fire suppression systems that require reliability and precision [31].

Graphene Oxide Based Detectors

Graphene oxide (GO)-based fire detectors have been investigated as experimental fire-sensing technologies that take advantage of GO's thermal, electrical, and chemical reactivity [37, 38]. GO is a graphite derivative composed of single-layer carbon sheets decorated with oxygen-containing functional groups, which make it disolvable in water and easy to process [39]. These oxygen groups impart hydrophilicity and chemical activity, but they also render GO electrically insulating and thermally unstable compared to the pristine graphene [40]. Although these traits have inspired research into advanced sensing concepts [38], they also introduce reliability challenges under real-world conditions. GO-based fire detectors typically operate through thermally induced reduction: heating breaks down oxygen functional groups, partially restoring the conjugated carbon structure and increasing conductivity [39, 41]. The resulting drop in resistance can be used as a detection signal, and in some cases, films exhibit color changes that serve as potential visual indicators [42]. However, these responses are strongly dependent on the stability of GO's surface chemistry, which tends to degrade over time. Because GO contains abundant oxygen groups, it has also been explored as a gas-sensitive material. In principle, physisorption or chemisorption of combustion-related gases (NH_3, NO_2, CO) can alter their electrical resistance [34, 43–45]. Yet in practice, this approach suffers from limited selectivity, strong cross-sensitivity, and signal drift, particularly in humid or oxygen-rich environments [46]. Even under controlled conditions, prolonged exposure to ambient air leads to a gradual loss of functional groups and a decline in sensing performance [47]. Overall, while GO offers interesting chemical reactivity for proof-of-concept studies, its susceptibility to humidity, instability in ambient air, and degradation over time hinder its suitability for robust fire safety systems. These limitations highlight the need for alternative strategies, such as image-based fire detection, which avoid the material instability issues inherent to chemical sensing approaches.

Limits of Sensor-Based Fire Detection

Sensor-based fire detection systems, including smoke, heat, and gas detectors, have long been the cornerstone of traditional fire safety infrastructure. Although widely used, these systems face key limitations that can reduce their reliability under real-world conditions. Traditional sensor-based technology typically relies on physical indicators, such as smoke concentration and air temperature, to detect the presence of fire. However, temperature and smoke detectors may be influenced by external conditions, including the height at which they are installed, the presence of dust, and the airflow speed [11]. When alarms are delayed, fire services may be unable to suppress the initial phase of fire promptly and effectively [48]. Since fires can escalate rapidly within just a few minutes, even short delays in detection significantly reduce the chances of controlling the fire at its earliest stage. Sensor response times are often delayed in the early stages of fire development when heat, gases, or smoke remain minimal or have yet to reach the detection point [49]. This delay becomes more severe in large or ventilated areas, where heat and smoke may disperse before reaching detectable levels [49]. Another significant limitation is the tendency of these systems to trigger false alarms, which commonly occur due to non-fire elements such as cooking emissions, steam vapors, or airborne dust [49]. In addition to limited sensitivity and specificity, sensor-based systems usually generate only binary outputs and lack spatial data on the fire's location or size [50]. This absence of localization makes emergency response more difficult and limits the ability of automated systems to operate in a targeted way. Moreover, in large or structurally complex indoor areas, sensor placement might not ensure complete coverage. Fires that start distant from the sensors may remain undetected during critical periods, reducing the system's overall reliability. Due to these limitations, there is growing interest in image-based methods that offer faster detection, enhanced spatial awareness, and greater resilience in dynamic environments.

2.1.3 Fire Detection Using Image-Based Methods

Image-based methods provide significant advantages compared to traditional sensor-based fire detection systems. In contrast to sensors, which typically require substantial fire development to trigger, image-based approaches can detect even small flames or early smoke [11]. Fire detection at early stages improves reaction times and lowers the risk of fire spread [11]. Additionally, visual information from cameras provides rich contextual information, allowing systems not only to identify the presence of fire but also to estimate its location, size, and progression over time. Such spatial and temporal awareness is crucial for enhancing situational understanding and guiding emergency response efforts more effectively [4]. Traditional image-based detection systems employ image-processing techniques that rely on attributes that describe fire on the image, but the state-of-the-art approaches use deep-learning methods. In addition, image segmentation has proven effective in detecting and tracking fire events [10]. The goal of fire semantic segmentation is to classify each pixel as fire or non-fire. This task, along with salient object detection and general semantic segmentation, has greatly benefited from advancements in image processing [51]. Nevertheless, challenges remain due to varying backgrounds, different fire scales at multiple stages, and interference from visually similar objects [52]. The following section discusses traditional fire detection approaches based on image processing algorithms for image segmentation, which are most relevant to the objectives of this thesis.

Advantages and Challenges of Image-Based Fire Detection

Image-based fire detection has significant advantages over traditional sensor-based systems. These image-based systems identify fire by directly analyzing image data from camera feeds, unlike traditional sensor-based detectors that depend on the accumulation of combustion byproducts. This enables faster reaction times, particularly during the smoldering phases of fire development when heat and smoke are still scarce. Another substantial advantage is the capacity to specify the fire's location within the scene. The spatial information is critical for emergency response and supports targeted activation of alarms or suppression systems inside complex environments [11]. Image-based systems can also estimate the size and spread of the fire and provide contextual information about the surrounding environment, including the presence of people, obstacles, or nearby hazards [11]. Such capabilities enhance situational awareness and can be integrated into autonomous platforms or decision support systems [53].

Despite their potential, image-based fire detection systems face several significant challenges in real-world environments. One major issue is the presence of reflections from glossy surfaces such as glass, polished floors, or metal, which can create flame-like visual patterns and result in false alarms [19]. As mentioned before, artificial light sources, including incandescent bulbs, LEDs, and monitors, often emit warm hues and fluctuating brightness that closely resemble the appearance and motion of actual fire [19], posing a significant challenge in image-based fire detection. In addition, objects with similar color or texture to flames, such as orange clothing or flickering decorative lights, can confuse the detection model, particularly in rule-based or poorly generalized systems [19]. Visibility can also be significantly reduced by environmental factors such as smoke, fog, or steam. These elements obscure flame contours and reduce contrast, making it difficult for the system to extract reliable visual features [54]. These limitations motivated early efforts to use traditional computer vision techniques such as color thresholding, motion detection, and rule-based filtering, which laid the foundation for more robust learning-based methods.

2.2 Traditional Computer Vision Techniques for Fire Detection

Traditional fire detection methods in computer vision rely on handcrafted features and rule-based algorithms that focus on visual properties such as color, shape, motion, and texture. While simple and computationally inexpensive, these approaches struggle to generalize under real-world conditions, where variations in lighting, smoke, or environment can cause false alarms. Because the features are manually designed, they lack adaptability and must be re-engineered for each new environment or dataset. As a result, their performance often plateaus on complex detection tasks, making them less robust and scalable compared to modern deep learning–based approaches that can automatically learn richer and more transferable representations.

2.2.1 Color-Based Thresholding

One of the earliest and most commonly used methods in traditional fire detection is color-based thresholding. This approach is based on the idea that fire has a distinctive appearance in images, often appearing in shades of red, orange, and yellow colors. By setting fixed boundaries within a chosen color space, the system can recognize these characteristic colors. For example, pixel values can be analyzed in the RGB, HSV, or YCbCr space, and those that fall within the known range for fire are marked as potential fire pixels. HSV is frequently used because it separates brightness from color, which allows better handling of changes in lighting conditions. Large areas of the image that resemble fire can be rapidly identified by employing logical principles on individual pixels. The low computational cost of this technique is one of its advantages. It is suitable for real-time applications and devices with limited processing power, as it does not require complex models or training data. It can also be utilized to detect the presence of fire in video feeds by applying color thresholds to each frame. Adaptive thresholding has been implemented in certain studies to enhance effectiveness in dynamic lighting environments. The boundaries can be adjusted by this adaptation based on the average color values of the scene or by utilizing statistical models to minimize false detections. However, the majority of implementations still depend on static principles, which limits their ability to generalize across a diverse range of environments. Color-based thresholding is subject to substantial constraints despite its apparent simplicity. It is extremely sensitive to the presence of other objects that show similar colors to fire, ambient illumination, and camera quality. Reflections, bright clothing, and artificial lights can often produce false positives. Moreover, fire in its early stages or smoke-heavy environments may not emit strong light, and the color contrast may be weak. In such cases, the method may fail to detect the event completely. Because thresholding considers each pixel independently, it does not utilize the spatial or temporal structure of the flame. As a result, it may identify isolated pixels or disconnected regions, reducing the overall accuracy of fire localization. To address these challenges, color thresholding is often combined with other features such as motion, texture, or shape. Motion detection can help filter out static objects that share the same color range, while texture features can highlight the flickering nature of flames. In modern systems, thresholding may be used as a pre-processing step before more complex analysis is performed. This multi-step approach can improve reliability while maintaining low computational requirements. Overall, color-based thresholding remains a useful method for basic fire detection tasks, especially in controlled indoor environments or as an initial filter in larger systems. Figure 2.4 visualizes fire segmentation using color-based thresholding.

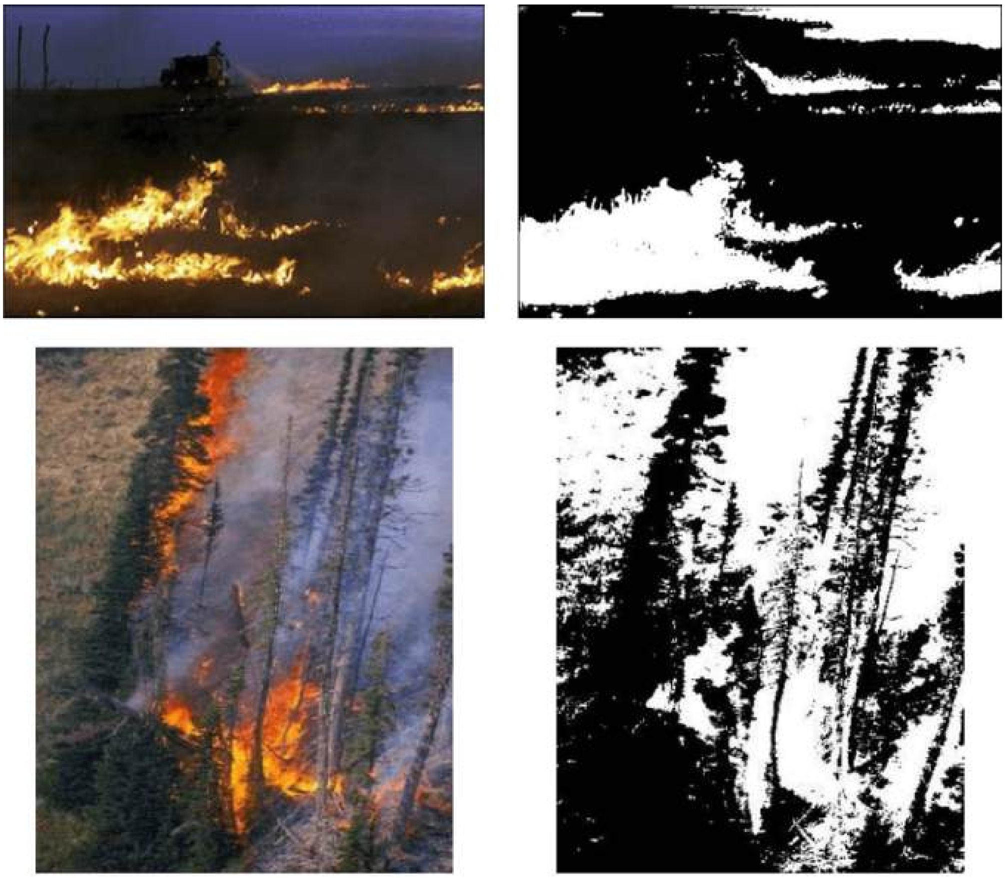


Figure 2.4: The visualization of using color-based thresholding to segment fire. Image source: *Celik et al.* [18].

Color thresholding has been applied in a wide range of fire detection research. Chen et al. [55] proposed an early fire-alarm method using video processing that extracts fire and smoke pixels based on RGB chromatic analysis and disorder measurement, with fire detection verified through flame growth dynamics and smoke presence. The method achieved fully automatic surveillance with low false alarm rates. Toreyin et al. [56] proposed combining color and flicker analysis to reduce false alarms in fire detection from video. The authors utilized spatial wavelet transforms to monitor the decrease in high-frequency energy and chrominance values caused by semi-transparent smoke while employing a Hidden Markov

Model to capture the temporal flicker behavior of smoke regions. The proposed method effectively combines edge analysis, background comparison, and flicker modeling, showing promising results for both real-time and offline smoke detection in video surveillance systems. Celik et al. [18] developed a real-time method using YCbCr color space and pixel intensity models. The authors proposed a fire detection approach that combines adaptive background subtraction with a statistical fire color model, where each color channel is modeled using Gaussian distributions. Foreground objects are first extracted and then verified using fire color statistics derived from sample images, enabling accurate identification of fire candidates in video frames. The system showed high efficiency and accuracy in real-time fire detection, operating smoothly on low-resolution video.

2.2.2 K-Means Clustering

K-Means clustering is an unsupervised image segmentation method that groups pixels based on similarity in color or intensity [57]. The algorithm begins by selecting a predefined number of clusters. It then assigns each pixel to the cluster whose center is closest in terms of feature values, typically based on color channels. After all pixels are assigned, the centers of the clusters are updated based on the average values of their assigned members. This process repeats until the assignments no longer change significantly [58]. A key advantage of this method is that it does not require labeled training data. This makes K-Means suitable for exploratory analysis or use in settings where annotated fire datasets are limited. It can adapt to various visual scenes and identify potential fire regions, even when the flame shape is irregular or partially obstructed. Additionally, K-Means can serve as a pre-processing step within larger fire detection systems. For example, once flame-colored clusters are identified, other components of the system can further analyze those regions to confirm whether fire is truly present. Despite its simplicity, K-Means clustering has significant limitations. The algorithm requires specifying the number of clusters beforehand, which can be difficult when dealing with scenes that vary in complexity [57]. If too few clusters are chosen, different areas, such as fire and background, may be grouped. If too many clusters are selected, the image may be broken into segments that are too detailed, making interpretation more difficult [58]. Additionally, K-Means begins with randomly selected centers, and this initial randomness may yield inconsistent results across different runs [57]. Because of this, the algorithm may need to be executed several times to produce consistent results. A further limitation is that K-Means relies only on basic visual features such as brightness and color. It does not utilize spatial or semantic information, which may lead to the misclassification of non-fire objects that resemble flames in appearance. This issue becomes more noticeable in scenes containing reflective surfaces, bright lighting, or objects with warm-toned colors. To improve reliability, K-Means is often combined with additional techniques, such as color thresholding, motion analysis, or contour-based filtering. These complementary steps help distinguish actual flames from other visually similar regions. This method is a useful tool for segmenting images in traditional fire detection systems. While it has limitations in terms of accuracy and stability, it offers a fast and adaptable method for identifying candidate fire regions. When integrated with other algorithms, it contributes to more robust and flexible detection pipelines.

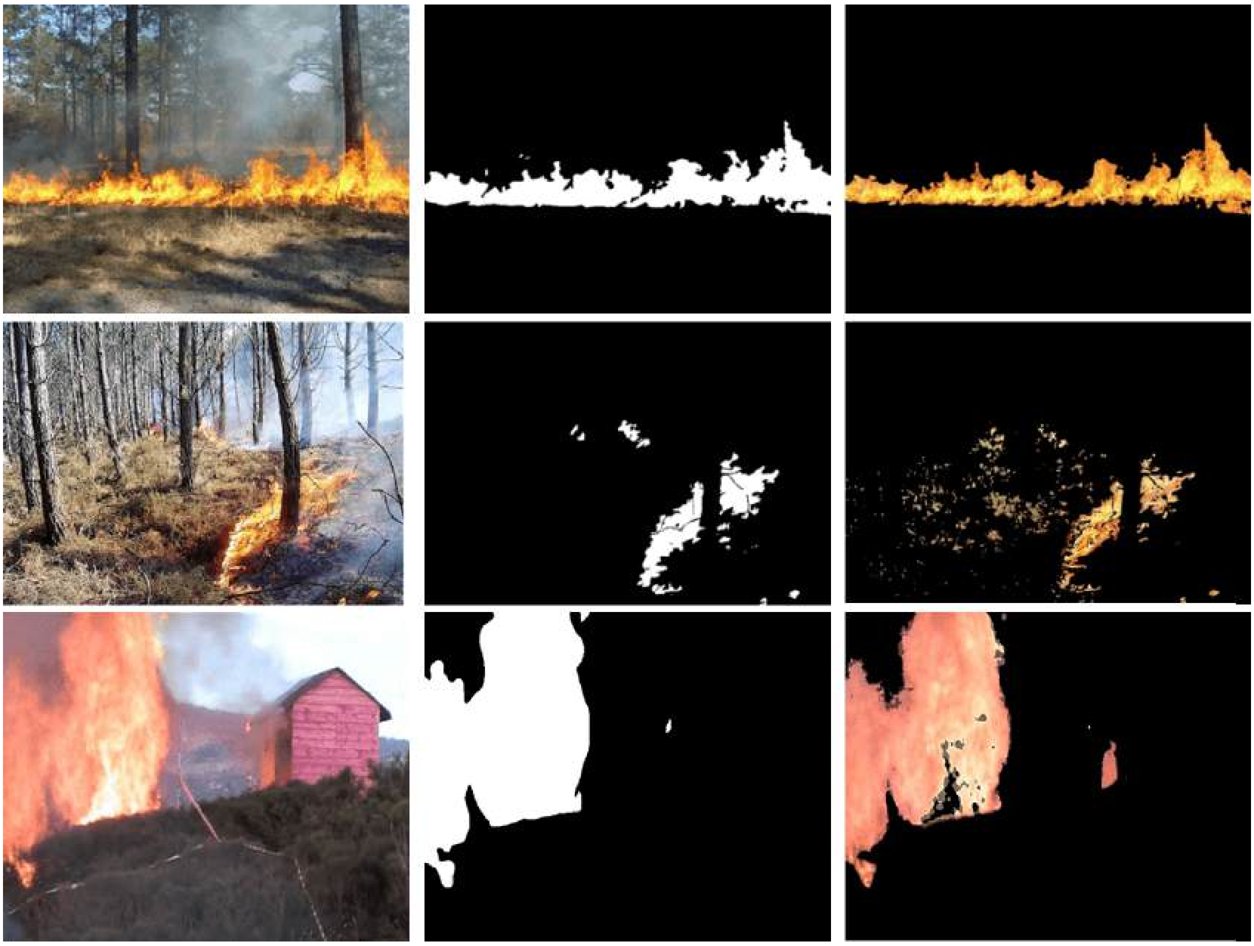


Figure 2.5: The visualization of using K-Means clustering to segment fire. Image source: *Rudz et al.* [59].

In the context of fire detection, this technique has been utilized to segment flame-like regions from background elements in visual data for fire detection tasks [60]. By grouping pixels based on color similarity, this unsupervised technique effectively isolates areas that share visual characteristics with those of a fire. Figure 2.5 illustrates utilizing K-Means clustering to segment fire on RGB images. Typically, one or more resulting clusters exhibit flame-like hues, enabling the identification of potential fire regions within an image or video frame. Anitha et al. [61] utilized K-Means clustering to identify forest fires by using land surface temperature. The authors utilized satellite imagery to determine the mean wavelengths of abnormal temperature distributions in a small region compared to their surroundings. Rudz et al. [59] proposed a two-step image segmentation algorithm for forest fire detection, combining optimized K-Means clustering on the blue chrominance (Cb) channel of the YCbCr color space with a filtering process based on local histogram comparison using reference fire data. The method demonstrated superior performance over existing approaches available at the time when evaluated through three supervised criteria, highlighting its accuracy in isolating true fire pixels while minimizing false detections.

2.2.3 Watershed Segmentation

The watershed algorithm is an image processing technique that interprets a grayscale image as a topographic surface. In this representation, the intensity values of pixels correspond to the elevation levels. Darker regions mark basins (valleys), while brighter regions represent ridges or peaks. The algorithm simulates a flooding process starting from local minima, and regions expand until they meet at ridges, which form the boundaries of segmented regions [62]. This natural analogy allows the method to produce closed contours around objects in a way that respects intensity gradients and local structure [63].

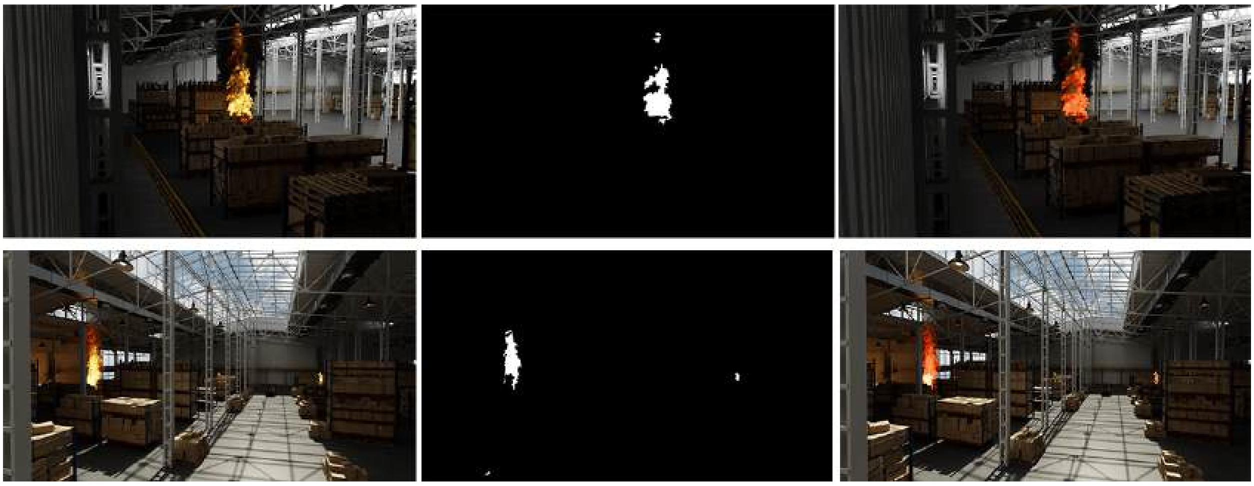


Figure 2.6: The visualization of using the Watershed algorithm to segment fire.

This method is especially effective for separating connected regions that have gradual transitions between them. However, it is also known to be sensitive to image noise and variations in fine texture. These small fluctuations can cause the algorithm to produce a large number of small and irrelevant segments, a problem known as over-segmentation. To mitigate this problem, gradient pre-processing is often implemented to accentuate significant edges while reducing noise. Figure 2.6 illustrates utilizing Watershed algorithm to segment fire on RGB images. Furthermore, variants of the algorithm based on markers have been developed. These versions include predefined foreground and background regions that direct the flooding process, helping the algorithm focus on significant regions. Watershed segmentation can extract fine boundary details and handle touching or overlapping objects, which makes it useful for detecting objects with unclear or diffuse edges [63]. This is particularly important in fire imagery, where flames often exhibit irregular, blurred, or rapidly changing boundaries. The flames may blend into the surrounding scene due to variations in brightness or smoke interference, and the spatial contrast between the flame and background may not be sharp. In such cases, the watershed algorithm can accurately trace flame shapes when guided by appropriate markers and gradient information. In fire detection systems, watershed segmentation is typically integrated as a post-processing step to refine regions identified by earlier methods such as color thresholding or motion analysis. After coarse flame areas are located, the watershed algorithm provides precise region boundaries that better match the true extent of the fire. This boundary information can then be used to calculate geometric features, estimate the spread of the flame, or filter out false detections caused by bright but non-fire regions. As such, watershed segmentation contributes to improved spatial accuracy and robustness in fire detection pipelines, especially in indoor or industrial environments where background clutter and lighting variability are common.

2.2.4 Contour Detection

Contour detection is a traditional method in computer vision that identifies the outlines or boundaries of objects in an image. A contour is formed by a set of continuous points that have the same intensity or follow a strong edge [64]. The most common approach to finding contours is first to apply an edge detection algorithm, such as the Canny or Sobel detector. These detectors compute changes in pixel intensity across the image and highlight areas where those changes are strong [65]. Once the edges are extracted, the contours can be traced by connecting neighboring edge pixels. Contour detection allows the system to understand the shape and structure of objects based on their boundaries. One of the benefits of contour detection is that it focuses on geometric features instead of color or texture. This makes it a valuable addition to other methods, especially in situations where color information may be misleading. Contour-based methods can help identify moving or irregular shapes in the image. They also enable the use of shape descriptors, such as area, aspect ratio, or circularity, to describe objects. These measurements can then be used to filter out unwanted objects or to classify regions based on known shape patterns. For example, small closed contours can be removed as noise, and large irregular shapes can be flagged as more relevant. Contour methods are preferable for detecting changes in the form of flames in the context of fire detection. Frequently, fire shows non-rigid and shimmering patterns that change over time. The dynamic contours generated by evolving patterns aid in flame recognition in images. When used in conjunction with motion or color cues, contour detection enhances system reliability by verifying that detected areas are not only bright or warm-colored but also exhibit shapes that reflect typical fire behavior [66]. For instance, extended and wavy contours with fast boundary changes could indicate active flames, but circular and steady contours may imply a false detection from a bright source. Nonetheless, contour detection has certain drawbacks when applied on its own. In environments with textured backgrounds, lighting shifts, or strong shadows, edge detectors may produce broken or noisy contours. This may result in a decrease in the accuracy of flame boundary detection [65]. Furthermore, contours solely define the object outline, not its interior. Consequently, to evaluate the entire region, these methods must be implemented in conjunction with other approaches. To improve the accuracy of region detection and reduce the number of false alarms, contour features are often utilized during the segmentation or refinement phases of fire detection applications. By integrating contour data with motion, texture, or color-based features, fire detection systems can achieve greater accuracy and robustness, especially in complex or indoor scenes. Figure 2.7 visualizes utilization of Contour Detection to detect fire in images.

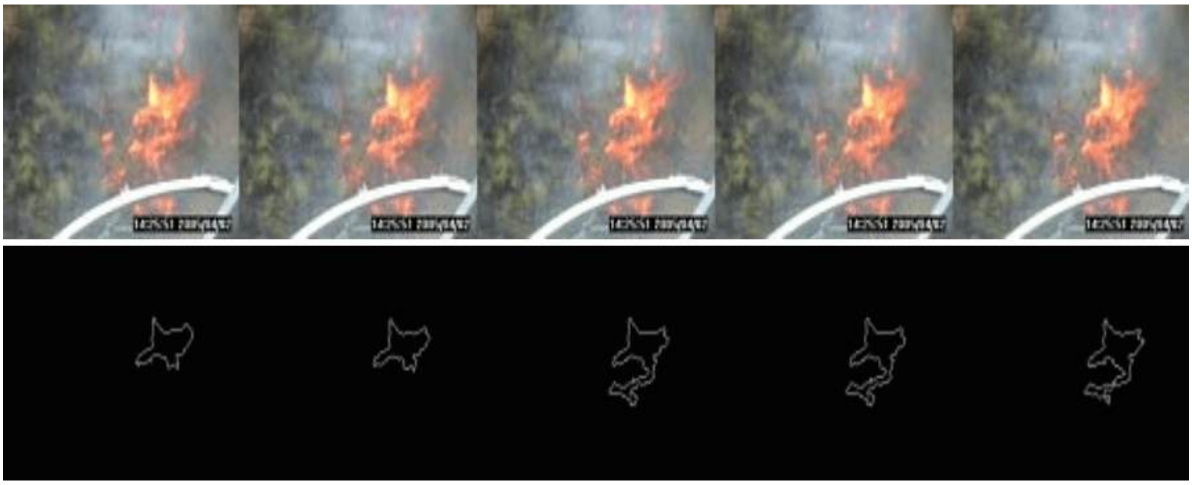


Figure 2.7: The visualization of Contour Detection use to detect fire on images. Image source: *Zhang et al.* [67].

Several studies have explored the role of contour detection in fire detection. Celik et al. [18] improved the detection performance of surveillance videos by utilizing both shape and color to filter out non-fire areas. Building upon this concept, the paper suggests a real-time fire detection system that combines a statistical fire color model with adaptive background subtraction to identify fire regions in video sequences. The approach is effective for early fire detection in dynamic environments due to its high accuracy and efficiency, which are achieved by modeling background pixels with Gaussian distributions and verifying the closest objects against fire color characteristics. In order to detect and observe flame shapes in real time, Töreyin et al. [66] implemented contour tracking in conjunction with flicker information. The paper introduces a real-time fire detection method that utilizes both spatial and temporal wavelet transforms to capture flame flickering, color variation, and boundary irregularities, which expands upon this approach. The method effectively reduces false alarms and improves detection performance in both surveillance and video analysis scenarios by incorporating these features with traditional motion and color signals. Zhang et al. [67] combined contour and frequency analysis for outdoor forest fire detection and achieved improved boundary accuracy. In a similar direction, the paper presents a method that first extracts fire contours and represents them using the Fast Fourier Transform, then applies temporal wavelet analysis to capture their dynamic changes over time, resulting in more accurate detection of frames containing fire, especially during the growth and fully developed stages.

2.3 Deep Learning in Visual Fire Detection

Deep learning, a subset of machine learning, employs computational models with multiple processing layers to learn hierarchical representations of data. These models are designed to mimic the ability of the human brain to acquire and interpret information from various sources, thereby implicitly recognize complex patterns within large datasets [68]. Deep learning has significantly advanced the field of computer vision, particularly through the application of CNNs, which are capable of learning hierarchical features directly from image data [69]. Unlike traditional computer vision techniques that rely on manually created features, such as specific color ranges or edge outlines, CNNs learn to identify important patterns at various levels of complexity. This ability to learn enables these models to perform well even with varying visual inputs, which has significantly improved tasks such as image classification, object detection, and semantic segmentation [70]. Deep learning methods in computer vision are typically categorized into three main tasks: classification, object detection, and image segmentation.

2.3.1 Image Classification

Image classification is a fundamental task in computer vision that deals with automatically understanding the content of an image. In this process, machine learning models are trained to categorize images into predefined classes. By learning to recognize patterns within input data, these models can assign a single or multiple labels to an image, indicating the presence of a particular object or event in a way that resembles human interpretation. Various types of image classification methods and techniques are used depending on the complexity of the task and the nature of the images. These methods are usually grouped by labeling scheme and by learning approach. In terms of labeling, a task may involve: single-label or multi-label classification. In single-label classification, each image is assigned to exactly one class. In multi-label classification, an image can belong to multiple classes at the same time. From the learning perspective, classification can be supervised, unsupervised, or semi-supervised. Supervised methods rely on annotated data. Unsupervised methods group images by similarity without predefined categories. Semi-supervised methods combine a small labeled dataset with a larger pool of unlabeled examples. Figure 2.8 illustrates how image classification techniques can be applied to detect fire in real time, enabling rapid response and minimizing potential damage.

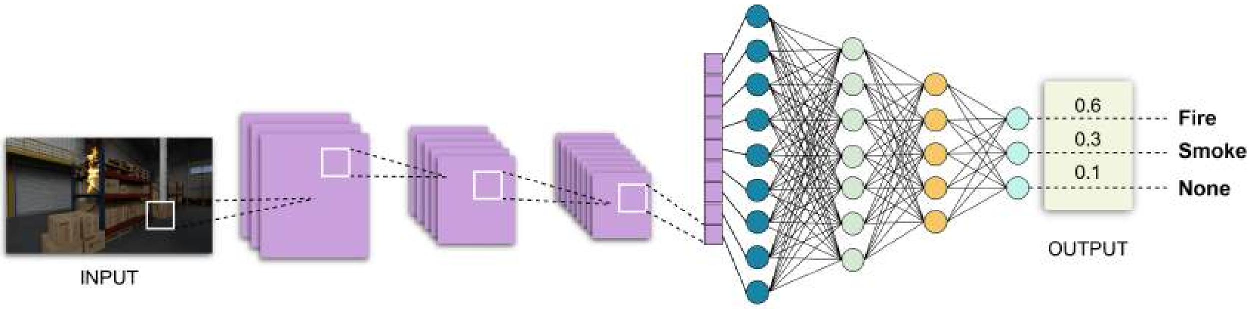


Figure 2.8: Application of image classification for fire detection, demonstrating the process of analyzing visual input to identify the presence of flames or smoke.

In practice, image classification has often been applied to the problem of fire detection, where the goal is to decide whether flames appear in a scene [71]. Several earlier studies have adopted this strategy for identifying fire in images [72–74], and they demonstrate that classification models can successfully determine if a fire is present. However, this type of approach provides only information on whether fire is visible in the image and does not reveal its location within the image. Because of that limitation, classification alone is not a practical choice when the objective is to suppress the fire, since spatial information is necessary to guide any intervention.

2.3.2 Object Detection

In image analysis, classification assigns a category to an object but does not specify its position within the frame. Localization advances this task by identifying not only the object category, but also its approximate location, most often represented with a bounding box [75]. The accuracy of such bounding boxes can differ depending on the employed method. Object detection builds upon these foundations by enabling the simultaneous recognition and localization of multiple objects in the same image, with each instance enclosed by a bounding box [76]. Figure 2.9 shows the application of Object Detection for fire detection in images.

This technique has become essential across a wide range of applications, including medical image analysis, pedestrian tracking, facial recognition, and fire detection. Recent research has explored the use of object detection models, such as YOLO and Faster R-CNN, for fire recognition in images and video streams [77–80]. These methods demonstrate that detection models can not only confirm the presence of fire but also localize it, enabling more effective monitoring and early warning systems. Despite its broad utility, object detection remains challenged by image variations such as changes in scale, viewpoint, and lighting.

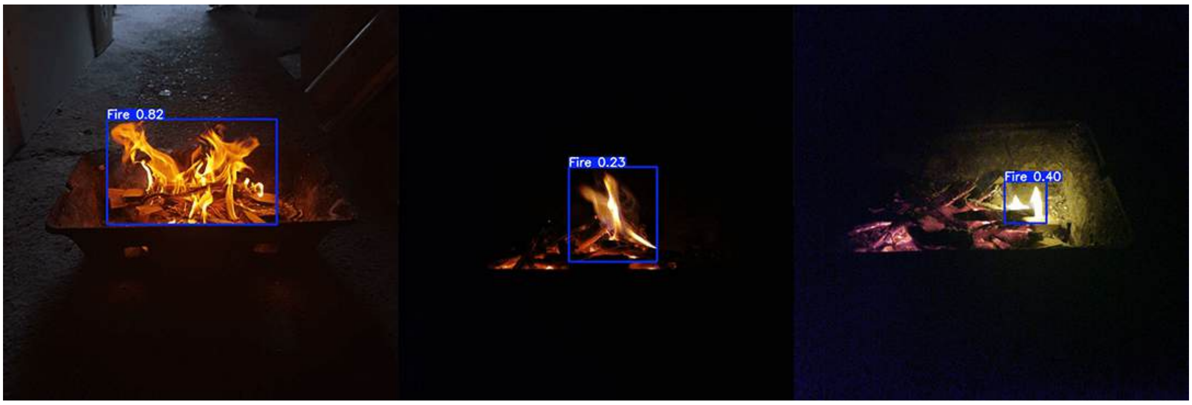


Figure 2.9: Visualization of object detection showing the detection and localization of fire within images.

2.3.3 Image Segmentation

Image segmentation in deep learning refers to the process of dividing an image or video frame into multiple meaningful regions, such as objects or boundaries. This division allows for more effective analysis and interpretation of visual data [76, 81]. Unlike image classification, which assigns a single label to an entire image, or object detection, which identifies locations by using bounding boxes, image segmentation provides pixel-level predictions where each pixel is assigned a specific class. This fine-grained approach enables a deeper understanding of visual content. Image segmentation can be categorized into different types, each capturing a distinct level of detail in scene interpretation.

1. **Semantic segmentation** assigns a class label to every pixel in an image. If multiple objects from the image are assigned the same pixel-level class no distinction is made between separate object instances. This method is particularly relevant in tasks such as fire detection or medical image interpretation, where the overall distribution of classes is the primary focus.

2. **Instance segmentation** builds upon semantic segmentation by distinguishing between individual objects of the same category. Each occurrence of an object is separately identified and segmented. This capability is crucial in applications such as counting vehicles in urban environments or identifying multiple pathological regions in medical scans.

3. **Panoptic segmentation** integrates the principles of semantic and instance segmentation. It assigns class labels to all pixels while also providing unique identifiers for each object instance. This combined approach enables a more complete understanding of complex scenes by capturing both object-level detail and contextual information.

Image segmentation provides a more detailed representation of visual data than image classification or object detection. It not only identifies the presence of objects but also delineates their precise shapes and spatial boundaries. This pixel-level understanding allows for accurate analysis in complex scenes where objects may overlap or display irregular structures. Such precision is crucial in domains that demand reliable interpretation, including medical imaging, autonomous driving, and fire detection. In the context of fire detection, segmentation has become particularly important for improving monitoring and safety. While object detection methods such as YOLO or Faster R-CNN can confirm the presence of fire and indicate its approximate location through bounding boxes, segmentation produces fine-grained masks that capture the exact extent of fire regions. This boundary information is essential for estimating the spread, intensity, and area affected, which directly supports more accurate risk assessment and timely response. Recent studies have employed segmentation models for analyzing fire in both images and video streams [11, 82, 83]. These studies demonstrate that segmentation provides spatial detail that object detection methods cannot fully capture, leading to a more complete understanding of fire behavior. Building on these advances, deep learning has become the dominant approach in image analysis, as it consistently outperforms conventional rule-based techniques that struggle with the variability and complexity of real-world conditions [84]. Figure 2.10 shows the application of Image Segmentation for segmenting fire in images.

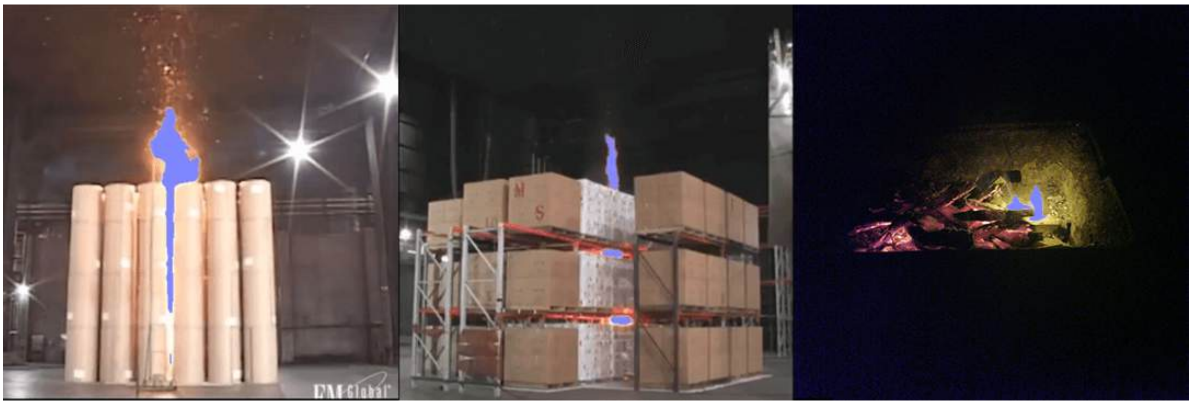


Figure 2.10: Visualization of image segmentation used to segment fire regions within images.

In visual fire detection, deep learning offers a robust approach for capturing the spatial structures and intensity changes typical of flames and smoke. While classification can suffice for triggering alerts, adding detection and segmentation provides precise localization, which improves risk assessment, assists emergency response, and supports automated suppression systems. These models also generalize well across various scenarios and camera angles, which is especially valuable in dynamic settings such as industrial sites, public spaces, and surveillance systems.

2.3.4 Applications of Deep Learning in Fire Detection

Sharma et al. [85] explored fire detection by adding a fully connected layer to well-known VGG16 and ResNet50 models. Dunnings i Breckon [86] used super-pixels along with Inceptionv1, AlexNet, and VGG16 to detect fire without using temporal information from the scene. The complexity of CNNs was reduced by keeping only a few convolutional, pooling, and dense layers. To detect fire, Xie et al. [87] used static features and dynamic motion flicker information. Hou et al. [88] proposed an improved DeepLabv3+ model that accurately segments flames and smoke in indoor settings. The integration of atrous convolutions into the network enhanced segmentation quality across various image resolutions. In Mseddi et al. [89] a method integrating YOLOv5 and U-Net for fire detection was proposed. This method employs YOLOv5 to detect and extract the bounding boxes that contain fire, while U-Net is utilized to segment the fire areas within those bounding boxes. Likewise, Kim i Lee [90] utilized Faster R-CNN to identify potential fire locations based on spatial features and employed a Long Short-Term Memory (LSTM) network to analyze fire dynamics. A known limitation of CNN-based methods is their reduced ability to detect small fire regions, which is attributed to the fixed size of their receptive fields. To address this issue, recent work in fire detection includes attention mechanisms that help preserve localized features. Niknejad i Bernardino [91] proposed a spatial self-attention approach to capture long-range pixel dependencies, along with a novel channel attention module that uses classification probabilities as attention weights. Shahid et al. [4] developed a spatiotemporal self-attention model for fire detection and segmentation. Their method uses self-attention to enhance spatial and temporal features, reducing reliance on shape or size, and improving performance through stronger spatial-temporal connections.

2.4 CNN Architectures for Image Segmentation

Numerous researchers have investigated the use of various algorithms for fire detection, leveraging recent advancements in artificial intelligence. Many of them have implemented CNNs to address challenges in fire detection and segmentation. Although various architectures are available for image segmentation, this section focuses on the most influential CNN architectures used in our research.

2.4.1 Feature Pyramid Network (FPN)

Feature pyramids are a fundamental component in recognition systems for detecting objects at different scales, but they are often avoided because they are computationally- and memory-intensive [92]. The Feature Pyramid Network [92] is an efficient feature extractor that upgrades pyramid networks with effective multi-scale feature representation.

The FPNs generate correspondingly scaled feature maps at different levels in a fully convolutional manner by utilizing an image of arbitrary size as input. Although initially developed for object detection, FPNs have been effectively applied to a variety of tasks, including instance segmentation, semantic segmentation, keypoint estimation, and even depth prediction and panoptic segmentation. Due to their capacity to provide robust semantics across all spatial resolutions, they are a highly adaptable backbone module.

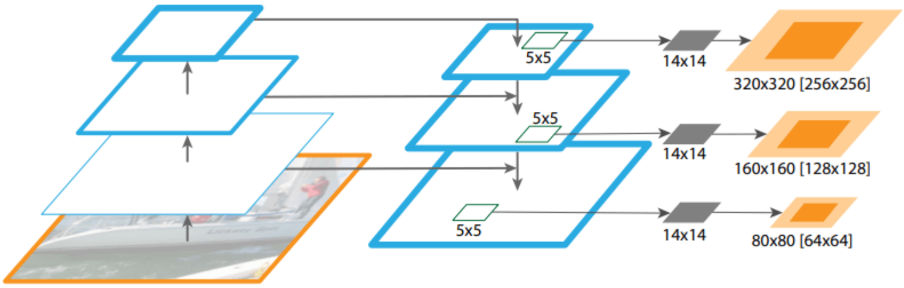


Figure 2.11: The structure of FPN architecture. Image source: *Lin et al.* [92].

FPN employs a top-down pathway with lateral connections to integrate semantically rich features extracted from deeper network layers with high-resolution spatial details. The top-down pathway generates higher-resolution features by upsampling spatially coarser but semantically richer feature maps from the upper layers of the pyramid. The upsampled features are enhanced via lateral connections that combine information from the bottom-up pathway. Each lateral connection integrates feature maps with corresponding spatial dimensions from both paths. Although the bottom-up feature maps underwent fewer subsampling steps, their activations are more precisely localized despite the fact that they contain lower-level semantic information. The network architecture is illustrated in Figure 2.11. This hierarchical structure enables the network to effectively capture both fine-grained details and high-level contextual information, thereby enhancing segmentation performance across objects of varying sizes. FPN enhances the robustness of segmentation models by extracting multi-scale features, particularly in complex environments characterized by occlusions and intricate backgrounds.

FPNs offer several significant benefits in multi-scale vision tasks. They obviate the need for traditional image pyramids by deriving all scales from a single backbone, and their lightweight additional layers impose only modest parameters and computational overhead, preserving real-time inference capability. The modularity of the top-down and lateral design allows for seamless integration with diverse pretrained model backbones, and attaching task-specific model heads at each scale ensures that small objects utilize high-resolution maps while large objects draw on semantically richer, lower-resolution features. Nevertheless, FPN also involves certain drawbacks. Retaining intermediate feature maps from every backbone stage can substantially increase memory consumption when processing very deep architectures or high-resolution inputs. The use of nearest-neighbor interpolation may introduce minor misalignment between fused features, which can degrade localization accuracy. Additionally, fixed channel dimensions across all pyramid levels may not be optimal for every application, prompting later work to explore adaptive fusion weights or dynamic scaling. Finally, the two-stage fusion pattern of the original FPN may be less expressive than more elaborate bidirectional or densely connected schemes in capturing complex cross-scale interactions.

2.4.2 U-Net

U-Net [93] is an image segmentation network characterized by its U-shaped architecture depicting its encoder and decoder branches. The encoder extracts spatial and semantic features, while a decoder reconstructs a segmentation map from the encoded information. The network is symmetrical, where each encoder layer is connected to a corresponding decoder layer with a corresponding skip connection. These connections allow the decoder to access fine-grained spatial details from earlier layers, improving the accuracy of boundary localization and the representation of object structures. Also, because U-Net does not include any dense layers, it can process input images of arbitrary size.

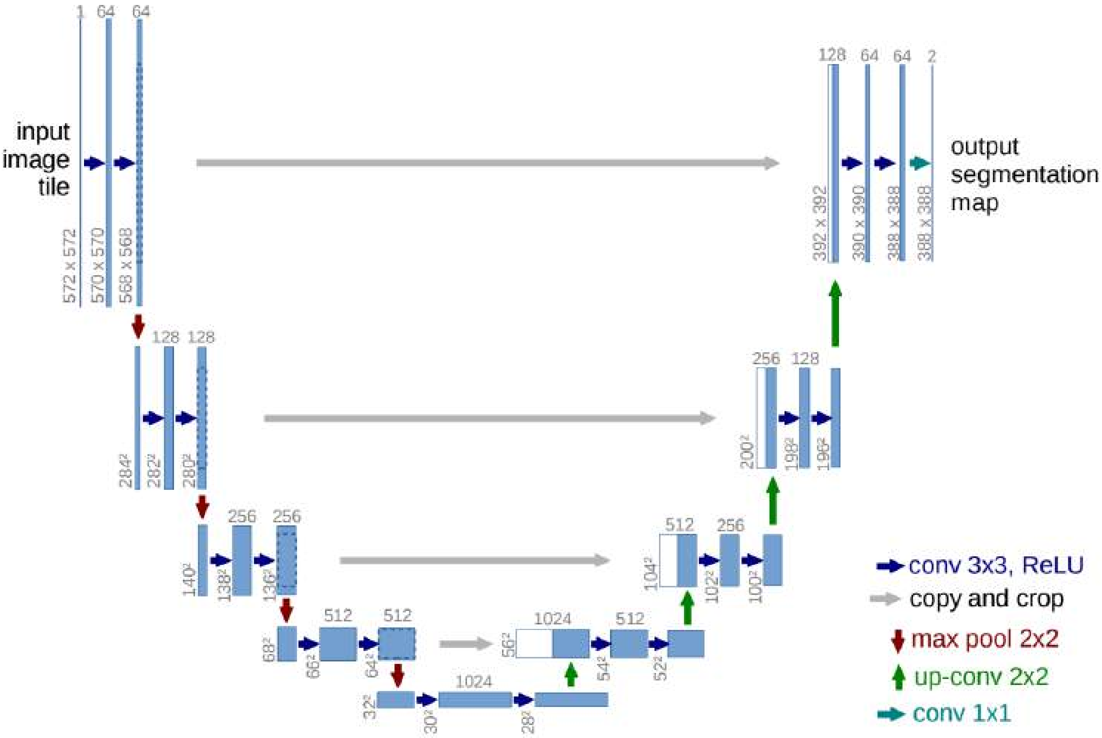


Figure 2.12: The structure of U-Net architecture. Image source: *Ronneberger et al.* [93].

The encoder path utilizes convolutional layers followed by max-pooling operations to reduce spatial dimensions and extract hierarchical feature representations. The decoder path progressively upscales the feature maps using transposed convolutions, restoring them to the original resolution of the input image. Each stage of upsampling is followed by convolutional layers that refine the output of the segmentation. Convolutions and max-pooling are essential in the encoder for learning complex patterns and compressing spatial information, while transposed convolutions in the decoder ensure accurate reconstruction of spatial structures. The use of skip connections between encoder and decoder layers is crucial, as it enables the transfer of localization information that may otherwise be lost during downsampling. The U-Net architecture is visualized in Figure 2.12. U-Net offers multiple benefits for image segmentation tasks. Integrating high-level semantic features with low-level spatial information enables accurate segmentation, especially in applications that require precise boundary delineation. The architecture is efficient in terms of computation and performs well, even with limited datasets, due to its ability to learn from augmented data. Despite these strengths, U-Net also has certain limitations. The memory requirements can be high, especially for large input images, due to the need to store multiple intermediate feature maps. Additionally, the fixed receptive field of standard convolutions may limit the network's ability to capture long-range dependencies. Furthermore, skip connections can transmit irrelevant textures or noise, which may negatively affect the final segmentation output. To address these challenges, later variants have introduced attention mechanisms and atrous convolutions to improve feature selection and expand the receptive field.

2.4.3 U-Net++

U-Net++ [94] is an extension of the U-Net neural network aimed at enhancing feature propagation and segmentation accuracy by incorporating nested and dense skip connections. While U-Net uses simple skip connections between corresponding encoder and decoder layers, U-Net++ introduces additional convolution blocks between these connections. These form nested pathways that allow features to pass through multiple intermediate steps before reaching the decoder. As a result, each decoder layer receives features from several encoder stages, which improves feature reuse and helps combine information at different levels of detail. The architecture of U-Net++ still follows the same general structure as U-Net. It consists of an encoder path that uses convolutional layers and downsampling to extract spatial and semantic information from the input image. The decoder then upsamples these features back to the original resolution to create a pixel-wise segmentation map. What sets U-Net++ apart is the presence of dense skip connections between intermediate encoder and decoder layers. The overall network structure is shown in Figure 2.13.

These connections allow the model to retain both high-level context and fine details, which improves its performance in scenarios involving complex objects or ambiguous boundaries. U-Net++ also benefits from more reliable training in addition to improved segmentation results. The dense connections facilitate the flow of gradients through the network, leading to improved convergence during the learning process. The design also

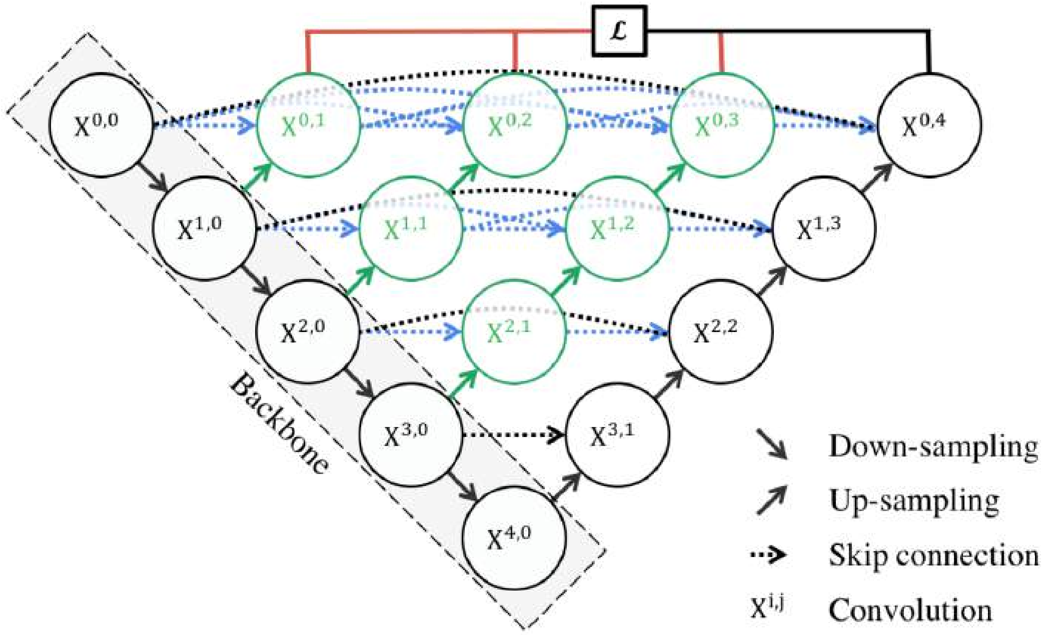


Figure 2.13: The structure of U-Net++ architecture. Image source: *Zhou et al.* [94].

accommodates deep supervision, which enables the extraction of outputs from a range of decoder depths. This feature enables the pruning of the model during inference, allowing for a balance between speed and accuracy that is task-dependent. Nevertheless, these improvements are accompanied by specific trade-offs. U-Net++ demands a greater amount of memory and computation than U-Net, and it is more complicated. It may also take longer to train and can overfit if the dataset is small and not well-augmented. Despite these challenges, U-Net++ remains a strong choice for tasks that require precise and reliable segmentation.

2.4.4 MANet

The Multi-scale Attention Network (MANet) [95] is an image segmentation model that integrates attention mechanisms with multi-scale feature fusion to improve segmentation accuracy and contextual interpretation. It follows an encoder-decoder design, where the encoder captures features using convolution and downsampling, and the decoder restores the spatial resolution through upsampling. Attention modules are embedded throughout the network to help focus on relevant regions while filtering out less informative areas, which enhances the detection of object edges and fine details. MANet incorporates a multi-scale feature block that applies convolutional filters of different sizes to extract varied spatial patterns. This allows the network to recognize both fine and broad visual structures in the image. By merging features across scales and applying attention, the network becomes more capable of identifying objects with varying shapes and appearances. These combined mechanisms improve performance in images containing noise or overlapping components, which can pose challenges for other models. The architecture of MANet model is illustrated in 2.14.

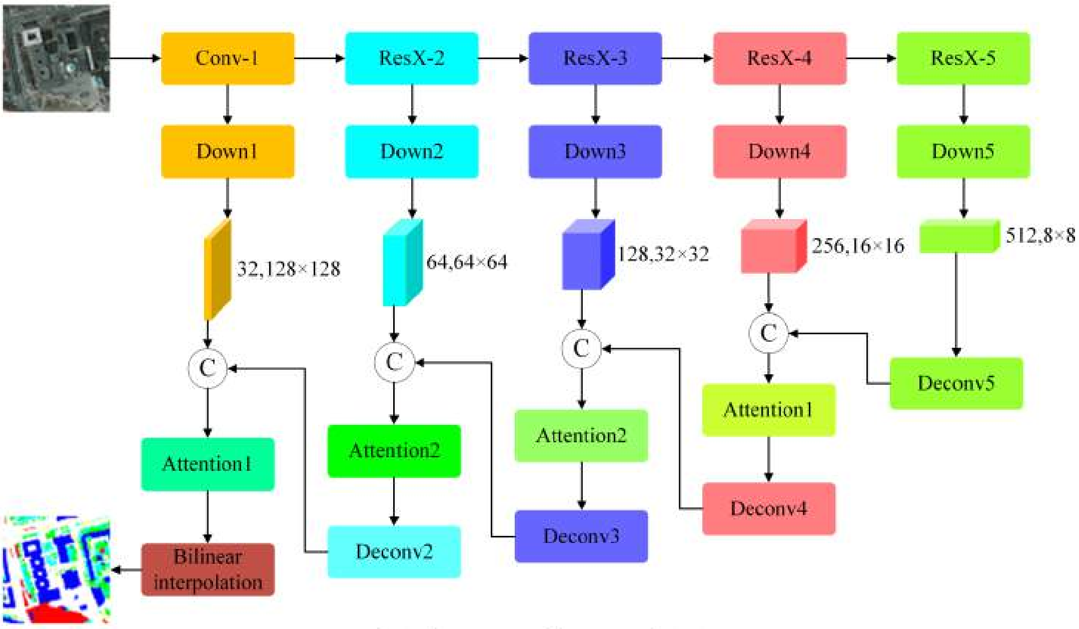


Figure 2.14: The structure of MANet architecture. Image source: *Rui et al.* [95].

MANet provides several key advantages, including more accurate object detection across various object sizes, a better focus on relevant areas, and improved performance in visually complex environments. However, these benefits come with increased memory usage and higher computational costs. In addition, the deeper architecture and larger number of parameters make the training process more demanding. Despite these trade-offs, MANet remains an effective model for context-rich and detailed image segmentation tasks and can outperform U-Net++ in scenarios where attention and multi-scale learning are essential.

2.4.5 DeepLabV3+

DeepLabV3+ [96] is an image segmentation architecture that combines several methods to collect both local and global information while keeping spatial details. The network uses atrous convolutions in the encoder to expand the area each filter can cover without reducing the resolution of the feature maps. This allows the model to detect patterns at different scales. A key part of the encoder is the Atrous Spatial Pyramid Pooling (ASPP) block. ASPP applies parallel atrous convolutions with different rates and includes global average pooling. This approach enables the model to acquire features from both small and large objects simultaneously, providing it with a more comprehensive understanding of the scene. The encoder is often built on an efficient backbone network, such as ResNet or EfficientNet. These backbones are trained to extract detailed features and form powerful representations. In DeepLabV3+, the Xception backbone is often employed, along with depthwise separable convolutions, to enhance the model's speed and efficiency. The decoder module then refines the output from the encoder. It upsamples the features and combines them with low-level features from earlier layers. These low-level features contain fine spatial details, such as edges and textures, which help improve the accuracy of object borders in the final segmentation map. The decoder employs several regular convolutional layers to process the combined features and then applies bilinear upsampling to restore the output to its original image size. This encoder-decoder structure enables the model to learn both global context and local details, thereby improving performance in complex scenes. The full architecture is shown in Figure 2.15.

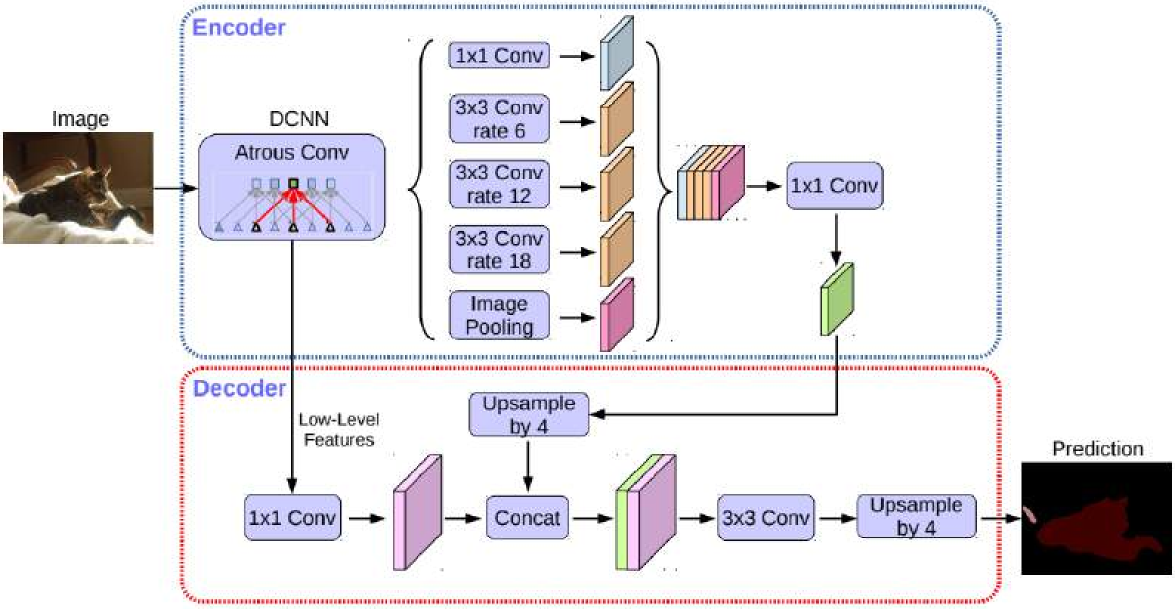


Figure 2.15: The structure of DeepLabV3+ architecture. Image source: *Chen et al.* [96].

DeepLabV3+ has several beneficial features. It can recognize objects of various sizes by combining information from different spatial scales. The use of atrous convolutions and the ASPP module enables the model to understand both close-by and distant details in the image. Its decoder refines object boundaries by combining fine details from previous layers. These features make DeepLabV3+ particularly useful in scenes with small objects, blurry edges, or shifting textures and shapes. By employing atrous convolutions, the model reduces the total number of operations, thereby improving efficiency compared to standard convolutions. At the same time, DeepLabV3+ has a few major drawbacks. It requires more memory and processing power than simpler models. The large backbone and multiple paths in the ASPP block increase the total number of calculations. As a result, training and inference may be slower, particularly when working with high-resolution images. It might not be the best option for real-time usage unless it is streamlined or modified. Even so, it may miss very small details or confuse objects with similar colors or textures.

2.4.6 SegFormer

SegFormer [97] is a transformer-based model for image segmentation that utilizes a hierarchical vision transformer (ViT) encoder and a lightweight Multilayer Perceptron (MLP) based decoder for feature extraction and the generation of segmentation maps. The encoder has several transformer stages, wherein spatial resolution is systematically diminished, but feature dimensionality is enhanced, facilitating multi-scale representation. SegFormer employs a self-attention mechanism to capture long-range dependencies rather than conventional convolutional methods. It employs overlapping patch embedding, which preserves local features more efficiently than non-overlapping tokenization. SegFormer, in contrast to conventional image transformers, omits positional encoding, enabling the model to adapt to various input resolutions. The hierarchical encoder, known as the Mix Transformer (MiT), is designed to produce feature maps at four different resolutions. This structure enables the model to extract information across different spatial scales while maintaining a lower number of operations compared to standard attention mechanisms. Each transformer block within the encoder utilizes spatial reduction attention to reduce memory usage while still capturing useful global context. Because of this, SegFormer can learn rich features that incorporate both local detail and wider context, which is crucial for segmenting objects that appear in various shapes and sizes. SegFormer's decoder was designed for multi-scale feature fusion, incorporating upsampling and combining features gathered from several encoder stages through MLP layers. This enables the model to preserve fine spatial features and high-level semantic information without relying on complex upsampling techniques, such as transposed convolutions. The decoder employs a linear projection to obtain per-pixel classification scores, thereby facilitating efficient segmentation with minimal computational cost. The network architecture is visualized in Figure 2.16.

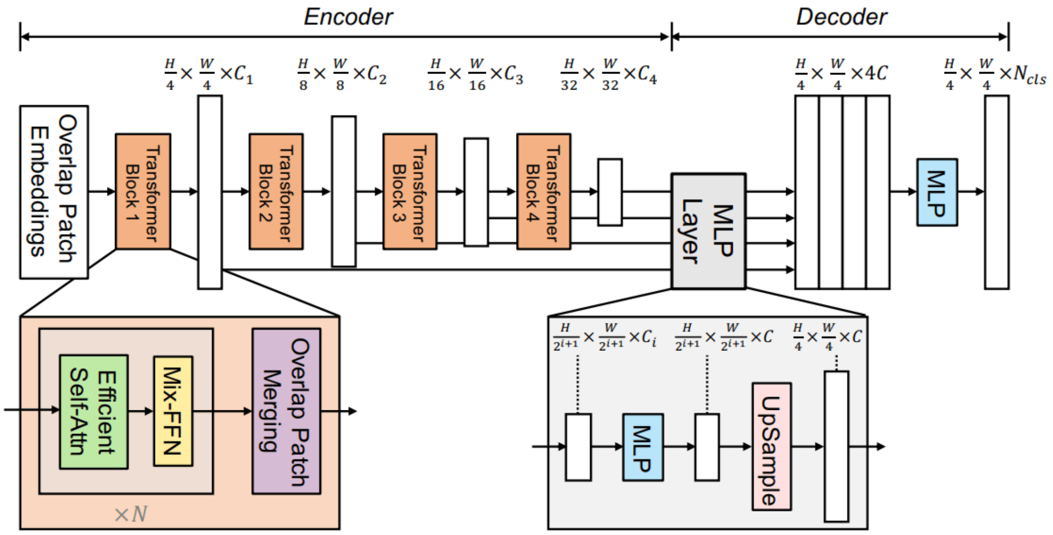

Encoder
Decoder
$\frac{H}{4}\times\frac{W}{4}\times C_1$
$\frac{H}{8}\times\frac{W}{8}\times C_2$
$\frac{H}{16}\times\frac{W}{16}\times C_3$
$\frac{H}{32}\times\frac{W}{32}\times C_4$
$\frac{H}{4}\times\frac{W}{4}\times 4C$
$\frac{H}{4}\times\frac{W}{4}\times N_{cls}$
Overlap Patch Embeddings
Transformer Block 1
Transformer Block 2
Transformer Block 3
Transformer Block 4
MLP Layer
MLP
Efficient Self-Attn
Mix-FFN
Overlap Patch Merging
$\times N$
$\frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C_i$
$\frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C$
$\frac{H}{4}\times\frac{W}{4}\times C$
MLP
UpSample

Figure 2.16: The structure of SegFormer architecture. Image source: *Xie et al.* [97].

SegFormer brings several advantages. It runs efficiently even with fewer parameters and does not require pretraining on large datasets to perform well. It generalizes across different tasks and domains, and its simple decoder design helps it run faster with lower memory requirements. At the same time, there are some limitations. Since it depends on transformer blocks, training can be slower compared to fully convolutional models, and the simplified decoder may not capture all fine details in very complex images. Still, SegFormer's architectural design enables it to capture both local and global contexts efficiently, making it suitable for a wide range of image segmentation applications.

2.5 Datasets and Dataset Creation Challenges

The scarcity of high-quality datasets remains a persistent challenge in the development of deep learning models despite the progress made in the complexity and efficacy of the models [98]. As a result, the development of more resilient and efficient deep learning models was facilitated by the creation of large datasets such as ImageNet and Microsoft COCO. The performance and generalization of deep learning models are highly dependent on the diversity, resolution, and precision of annotations in the training data [5]. However, gathering and annotating real data is a laborious and expensive process, especially in the field of fire detection.

In contrast to the medical domain, where this challenge has been well studied, fire detection has been investigated only sparsely. The development of high-quality fire datasets presents considerable challenges, primarily due to the inherent unpredictability of fire events Fires vary significantly in terms of size, intensity, and environmental factors, which complicates the creation of datasets that accurately represent real-world fire scenarios. Furthermore, access to fire scenes is often restricted due to safety concerns. Emergency responders and firefighters operate in high-risk environments, limiting opportunities for data collection in such situations. Legal and logistical barriers also hinder the process, as obtaining the necessary permissions to record or share footage of fire incidents can be a complex and sometimes insurmountable task. These various constraints make it particularly challenging to compile comprehensive datasets with reliable annotations, which are essential for training models or conducting detailed research on fire dynamics.

These challenges can be mitigated by synthetic data, which reduces the time and cost of data generation and circumvents the legal constraints that are associated with the use of real-world datasets. For synthetic data to be effective, it must sustain visual realism and include task-specific information. This contributes to the enhancement of generalization and the reduction of the disparity between real and computer-generated images. Synthetic data has been extensively investigated by researchers in a variety of machine learning applications and disciplines, either as a replacement for real data or its supplement.

2.5.1 Image Annotation

The performance of computer vision models depends heavily on the quality and accuracy of the training data, which is composed of annotated images or videos. Since a model learns by generalizing over the provided annotations, any error in the labels is also learned and reproduced during inference, making precise annotations a critical step in training an accurate neural network. Image annotation is the process of adding meaningful labels or tags to images to provide context for machine learning models, particularly in computer vision tasks [99]. In image segmentation, annotation requires pixel-level precision, where each pixel in an image is assigned a specific class. This fine-grained labeling allows models to capture detailed spatial patterns and boundaries between different regions. One application where precise annotation is especially critical is fire detection. In this context, image segmentation–based annotation enables models to differentiate flames and smoke from complex backgrounds at the pixel level. By labeling fire regions with fine-grained accuracy, segmentation not only improves detection performance but also enhances the model's ability to generalize across varying environments, lighting conditions, and fire intensities. Thus, image annotation for fire detection using image segmentation plays a vital role in building reliable systems that can support early warning and disaster management.

Beyond fire detection, the broader development of reliable computer vision systems also depends on large, high-quality datasets. However, producing such datasets through manual annotation alone is demanding. Annotations may be created by human experts or assisted by automated methods, both aiming to generate accurate ground truth data that enables models to interpret visual information. Because manual labeling is time-consuming and error-prone, growing attention has turned to semi-automatic and AI-assisted approaches that improve efficiency while maintaining accuracy. To meet these needs, a variety of annotation tools have been developed to make dataset construction more scalable, consistent, and less labor- and time-intensive. Such tools are essential not only for segmentation tasks like fire detection but also for object detection and other core computer vision applications. The following subsection examines existing annotation tools, highlighting their strengths and limitations in supporting robust dataset creation.

Existing Image Annotation Tools

A wide range of tools has been developed to support image annotation, each designed to meet the needs of different computer vision tasks. At a minimum, these tools provide core functionalities such as drawing polygons, rectangles, points, and lines to mark objects or regions of interest. Equally important is the ability to export annotations in formats compatible with various deep neural network architectures. Since different tools adopt different export standards, the choice of an annotation tool often depends not only on its labeling features but also on its compatibility with the intended model and workflow [19]. When selecting an image annotation tool, several factors should be carefully considered to ensure compatibility with the intended application. The choice often depends on the type of annotation required, since some tools are designed for bounding boxes, others for keypoints, and others for pixel-level segmentation. Scalability and collaboration features are also important, particularly for projects that involve large datasets or multiple annotators who need to coordinate tasks and monitor progress. Many modern platforms now include semi-automatic or model-assisted labeling functions, which reduce the amount of manual work while maintaining high accuracy. Additional considerations include the availability of export formats that match the training pipeline, the usability of the user interface, the infrastructure required for deployment, and the overall cost of use. Ultimately, the most suitable tool is one that combines annotation precision with workflow efficiency while aligning with both the dataset requirements and the broader objectives of the computer vision project.

Computer Vision Annotation Tool (CVAT) [100] is a free and open-source application that runs in a web browser and is designed to support the annotation of digital images and videos. It can be applied to tasks such as image classification, object detection, and image segmentation. In addition, it enables users to manage projects either locally or online, which makes collaborative annotation possible. CVAT provides a range of advanced features, including automatic annotation through the TensorFlow API, the ability to interpolate bounding boxes across video frames, and flexible project settings that allow for options such as image flipping, segmentation layers, dataset partitioning, and adjustments to display quality. The tool is valued for being feature-rich and adaptable, and it is particularly well-suited for collaborative annotation efforts. These qualities have made it a popular choice for developing datasets in both research and industry. CVAT has several limitations. Firstly, it is only available on Google Chrome and other Chromium-based browsers, which limits its accessibility for users across different client environments. Secondly, the configuration process and overall performance can be challenging when handling large-scale datasets. Finally, the user interface requires some time to learn in order to be used efficiently. Despite these limitations, CVAT has been widely adopted in domains such as autonomous driving, medical imaging, video surveillance, and robotics, where the availability of robust and scalable annotated datasets is critical.

Make Sense [101] is an open-source, browser-based image annotation tool. It is primarily used by researchers and practitioners in computer vision to construct datasets required for tasks such as object detection and image segmentation. The tool supports annotation through bounding boxes and polygons, thereby covering common requirements in image labeling workflows. It also enables the export of annotations into established formats, including YOLO, Pascal VOC, COCO, CSV, and VGG JSON, which facilitates compatibility with widely adopted machine learning pipelines. Since the application runs directly in a web browser, it does not require installation on local machines and can be used once images are uploaded. This design increases ease of use but simultaneously introduces limitations, particularly when projects involve multiple annotators, as concurrent work on the same dataset is not natively supported. In contrast, server-based frameworks such as CVAT or Label Studio are better suited for collaborative or large-scale annotation tasks. Within these constraints, Make Sense is best characterized as a lightweight and adaptable tool that serves individual projects or smaller teams, rather than as a platform optimized for extensive multi-user annotation.

LabelMe [102] is an image annotation tool and dataset created at MIT. The original system was a web application where users could draw polygons on images and assign labels. These annotations were stored in XML files and shared with the community. The web version is no longer open to new users, but the dataset remains available for research. A Python implementation of LabelMe exists today as a desktop application with a graphical interface. This version supports polygon, rectangle, circle, and line annotations. It can also be used to annotate a video by treating frames as images. The program relies on local storage for importing and exporting data. It can export annotations in formats such as JSON, VOC, COCO, and CSV. A key strength of LabelMe is the precision of polygon annotations and the large number of contributed images. Another limitation is the lack of built-in collaboration features. Datasets must be shared manually across computers if multiple users are involved. Despite these challenges, LabelMe remains a widely used and accessible tool for creating annotated datasets in computer vision.

Labelbox [103] is a paid annotation platform used to prepare datasets for machine learning research. It provides a visual interface for both annotation and data exploration, and supports images, video, geospatial imagery, natural language documents, audio, and HTML. Labeled data can be exported in the Labelbox JSON format and converted to other formats to match model requirements. The platform includes collaboration features that streamline work on large datasets, including a review queue for approval or revision, time tracking for labeling effort, summaries of label distribution, and in-context commenting. A label assistant can apply private models to propose annotations, and integrated exploration tools help researchers assess class balance and feature coverage, with optional training utilities to close the loop between labeling and model evaluation. Projects can be managed through dataset versioning and access controls, and the system integrates with common storage and SDKs for scripted import and export.

2.5.2 Datasets for Fire Detection

To support fire and smoke detection tasks, a variety of datasets have been created, each with its own unique characteristics, including modality, annotation type, and environmental diversity. These variances facilitate model training and evaluation, ensuring their robustness in diverse environments. The Corsican Fire Database [104] contains RGB and near-infrared (NIR) images for detecting and identifying wildland fires. In this dataset, the NIR images are captured with a longer exposure time, which increases the brightness of fire areas and simplifies segmentation using basic image processing methods. It includes 500 RGB fire images, 100 simultaneously captured RGB and NIR image pairs, and five sequences of RGB and NIR image pairs. All images are annotated at the pixel level, supporting the development of semantic segmentation models. The *BowFire* [105] dataset comprises of 226 images with resolutions ranging from 640 × 480 to 1653 × 1024 pixels. The dataset includes images depicting fires in various contexts such as building fires, car accidents, street riots, and woodland fires. The *D-Fire* dataset [106] contains images intended for detecting both fire and smoke. It is primarily used to train the YOLO object detection models by applying annotations formatted in YOLO to identify fire and smoke in individual images. The dataset includes 1,164 fire images, 5,867 smoke images, 4,658 images showing both fire and smoke and 9,388 images containing neither. It also provides 14,692 bounding boxes for fire and 11,865 bounding boxes for smoke. Most of the images are captured from video streams recorded by cameras placed on buildings or poles with a wide view of open areas. The image resolution varies significantly, ranging from 210 × 150 to 3791 × 2538 pixels. The *FLAME dataset* [107] includes drone images of fire collected during a controlled burn of forest debris in a pine area located in Arizona. It contains data from both RGB cameras and thermal heatmaps captured using infrared sensors. The RGB images support classification tasks using 39,375 labeled training examples, marked as either "Fire" or "Non-Fire," along with 8,617 labeled examples for testing. In addition, 2,003 images in the dataset include pixel-level annotations, which are suitable for developing segmentation models. The successor dataset, *FLAME 2* [108], is a multi-modal collection gathered from a drone using side-by-side dual-feed video that includes both RGB and thermal images of fire in a pine forest with an open canopy. Each image frame was carefully labeled as either "fire" or "no-fire" by two human experts, who used both RGB and thermal views to make accurate decisions. The publication also introduces a deep learning method developed for detecting fire and smoke at the pixel level. The *Dataset for Fire and Smoke Detection (DFS)* [109] is a high-quality dataset designed to support the advancement of research in fire and smoke detection using object detection. It is suitable for a variety of detection tasks, as it comprised of 9,462 real-scene images, each labeled according to the size of the fire. The dataset not only includes annotations for fire and smoke to enhance accuracy, but also introduces a new class, *other*, for items that may resemble fire due to their similar color or luminosity, thereby reducing false detections. All annotations follow strict and reasonable rules, and extensive experiments using various detection models provide a robust benchmark for evaluating performance on this dataset. The *Flame and Smoke Semantic Dataset (FSSD)* [82] provides detailed semantic annotations of fire-related objects captured in realistic indoor environments with diverse fire sources. This dataset has been evaluated using prominent deep learning architectures such as FCN, PSPNet, and DeepLabV3+, all of which demonstrated notable improvements in segmentation performance. Addressing the scarcity of training data specific to embers, the *FireFly dataset* [110] introduces a synthetic dataset of outdoor fires, generated with Unreal Engine 4. It comprises 19,273 frames designed to assess the performance of four advanced object detection models. Results show that incorporating this synthetic data led to an increase of up to 8.57% in mean average precision when applied to real-world wildfire scenarios, compared to models trained solely on limited real data. In a related direction, the *SWIFT dataset* [111] provides a large-scale synthetic dataset focused on outdoor wildfire scenarios, particularly in forested biomes. Developed using Unreal Engine 5, it combines high graphical realism with varied environmental conditions to depict fire and smoke under diverse outdoor settings. The dataset contains approximately 69,000 annotated images, each labeled for fire, smoke, both, or neither, and accompanied by segmentation masks and contextual metadata (e.g., humidity, wind, and camera viewpoint), thereby offering a valuable benchmark for wildfire-focused computer vision research. While several synthetic datasets have been developed for outdoor wildfire research, none have addressed indoor fire scenarios. To fill this gap, the *SYN-FIRE dataset* [5] introduces the first publicly available synthetic dataset dedicated to indoor fire detection in industrial settings. It comprises 1,402 images that simulate fire in industrial environments across five distinct scenarios, each designed to reflect typical indoor conditions by varying fire appearance, camera angles, and illumination at different times of day. By providing the first synthetic dataset for indoor fire imagery, SYN-FIRE represents a significant contribution to advancing research in image-based fire detection systems. Figure 2.17 displays the RGB image along with the corresponding annotations from all the described datasets in this subsection.

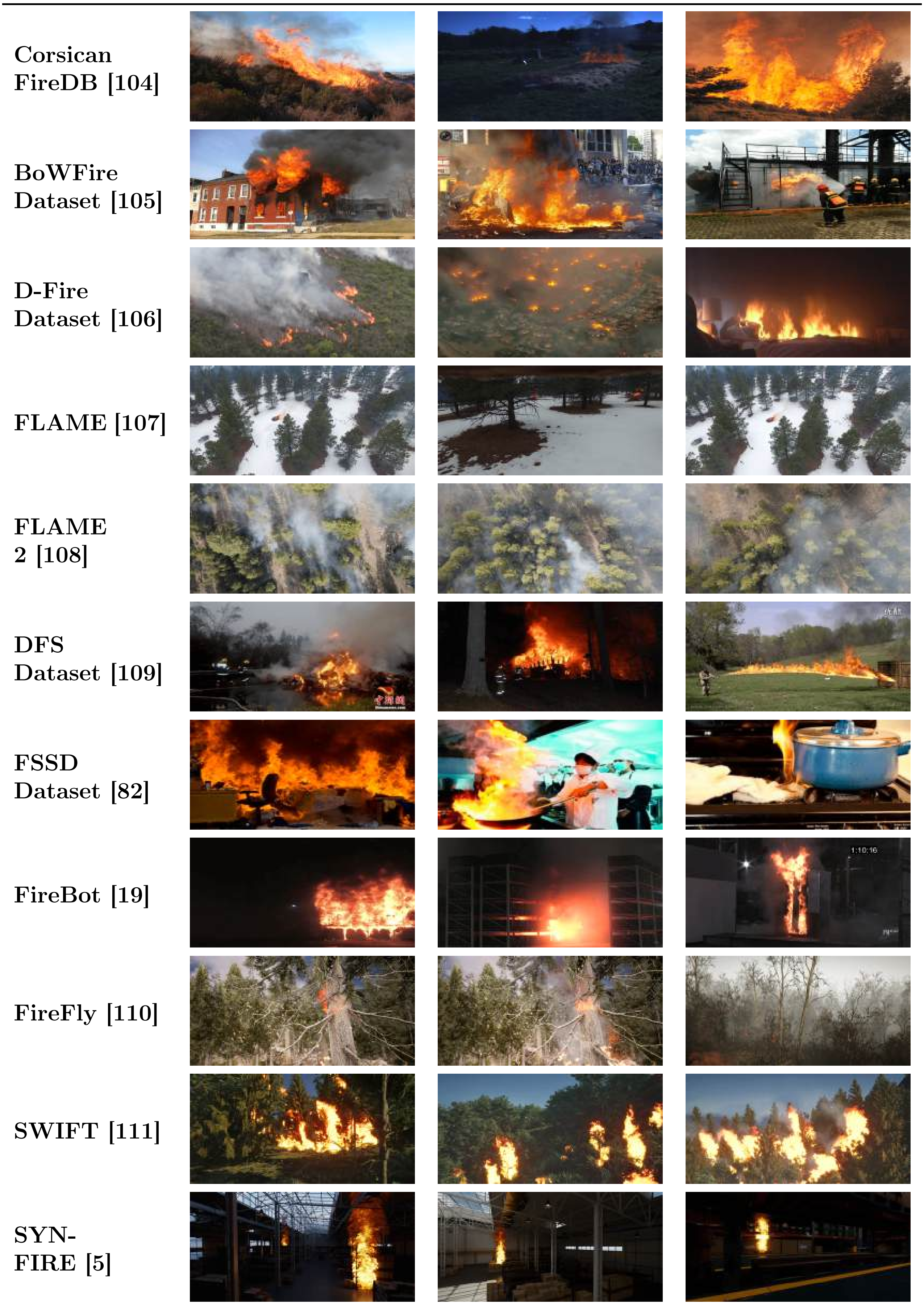


Figure 2.17: Evaluated datasets and examples of input RGB images.

3

Ensemble-Based Model for Indoor Fire Detection with Uncertainty Estimation

This chapter introduces a novel ensemble-based uncertainty estimation model for fire detection in indoor environments. The primary challenge of an image-based fire detection system is accurately identifying the shape and contours of flames. This task is complicated by several factors, including diverse backgrounds, varying fire sizes, and interference from objects that may resemble flames. This chapter begins with an overview of CNNs used in related research and compares their performance on our indoor fire dataset. Next, we propose the F2M model, which combines the fire segmentation masks produced by the five state-of-the-art models benchmarked in the first part of the chapter.

The chapter is organized as follows. Section 3.1 presents the main research objectives. Section 3.2 describes the benchmark dataset, explains the training process, and evaluates six modern semantic segmentation models. The F2M architecture and its evaluation results are discussed in Section 3.3, together with the strengths and limitations of the proposed method. Final remarks and concluding observations can be found in Section 3.4.

3.1 Research Objectives

This research aims to introduce the F2M model for semantic segmentation of indoor fires, adapting and evaluating state-of-the-art segmentation architectures for this task rather than proposing a new general backbone, and to quantify prediction reliability using Monte

Carlo–based uncertainty estimations. The biggest challenge that an image-based fire detection system must overcome is the inability to precisely determine the contours and shapes of flames. F2M is a compact encoder–decoder model that integrates outputs from five state-of-the-art CNN segmenters, distilling them into a single network that improves segmentation mask quality while reducing parameter count and inference cost. Despite the lack of scientific literature addressing the semantic segmentation of indoor fires in industrial environments, we have integrated established techniques from other deep neural network architectures to enhance model performance. We incorporate skip connections and Squeeze-and-Excitation (SE) blocks in our approach, as both have demonstrated their effectiveness in various contexts. Specifically, skip connections are implemented in the F2M model to maintain the spatial integrity of critical features by directly transmitting information from the encoder to the decoder. Skip connections help recover spatial details lost during downsampling, ultimately improving the model's ability to preserve crucial feature information. Additionally, the SE Block [112] ensures the retention of the most relevant and optimal features within the feature map, allowing the model to prioritize the most significant information for accurate semantic segmentation. Therefore, our research objectives are:

1. To benchmark existing state-of-the-art segmentation models for fire segmentation on custom indoor fire dataset.
2. To propose a novel F2M model that combines best-performing models to enhance fire segmentation accuracy.
3. To introduce an uncertainty estimation in fire segmentation through Monte Carlo dropout.
4. To evaluate and compare the novel F2M model performance with the best-performing model on 128 × 128, 256 × 256, and 512 × 512 resolutions.

3.2 Fire Segmentation Model Benchmark

This section provides a detailed dataset description that is implemented in the benchmark, as well as an overview of the network training procedure and implementation requirements. Additionally, we provide a benchmark analysis of six state-of-the-art semantic segmentation models: FPN, U-Net, U-Net++, MANet, DeepLabV3+, and SegFormer. For each model, 45 unique model configurations were generated by training them with three different input image resolutions and three distinct backbone architectures. The figure 3.1 shows the F2M research process. Data augmentation is applied as the first step to improve the performance of state-of-the-art semantic segmentation models. After each training phase, every model is evaluated using various encoder backbones and sets of hyperparameters. Models that achieved the highest Dice score on the validation dataset were selected for the final F2M model. Finally, uncertainty estimation was conducted during the inference stage for each selected model.

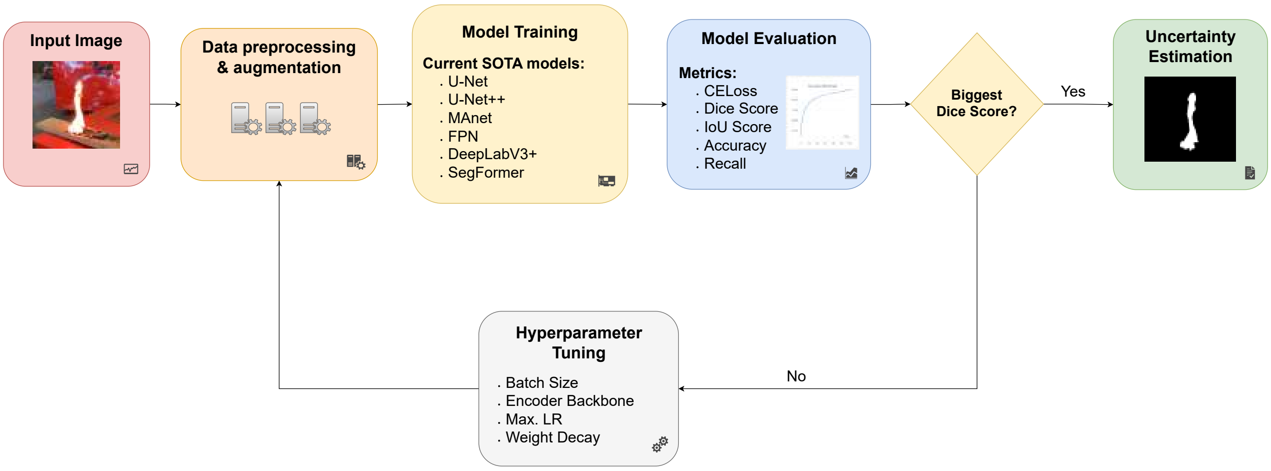


Figure 3.1: Flowchart showing the study workflow: data preparation, model development, performance evaluation, and final assembly with uncertainty analysis. Image source: *Arlovic et al.* [11].

3.2.1 Utilized Dataset

An extensive literature survey shows that the majority of related research is on the detection of outdoor fires. Outdoor fires have attracted significant interest from scientists due to the availability of data, the ease of surveying large areas, compelling applications like forest fires, and the importance of early fire detection due to the high environmental hazards. In contrast, the detection of indoor fires has largely relied on sensor systems. The volume of digitized data regarding indoor fires is minimal compared to outdoor fires, indicating significant potential for development, research, and enhancement. Therefore, we made a dataset for this study that primarily depicts flames and fires in a variety of indoor environments, such as residential structures, office buildings, shopping malls, warehouses, and industrial facilities. The primary objective was to create an extensive and comprehensive dataset for detecting indoor fires. Additionally, in order to improve the dataset's realism and variability, we applied a variety of secondary light sources, including reflections, various types of artificial lighting (such as incandescent, fluorescent, and neon), and natural illumination from sunrises and sunsets. Numerous images feature obscured flames, camera movement around the fire, motion blur, and flames of differing sizes and distances from the camera. Furthermore, to enhance the diversity of our dataset, we incorporated images of external flames, including those on rooftops and visible from windows. The images in the dataset were collected from multiple sources, mainly from FM Global and fire departments located in Los Angeles (USA), Washington D.C. (USA), Tokyo (Japan), and Zagreb (Croatia). These sources allowed us to utilize their resources, primarily images depicting real-life situations. In the annotation process of indoor fires, particular attention was directed toward scenarios involving smoke, varying visibility levels, and surrounding obstacles. Factors including image resolution, image scale, motion blur, and occlusion that may influence the detection of pixels representing fire were also taken into account. Table Table 3.1 presents the results of the detailed dataset analysis following a comprehensive annotation process.

Table 3.1: Dataset statistics grouped by metadata collected from image annotation.

	Training Dataset	**Validation Dataset**	**Test Dataset**
Annotated Images	5289	650	683
Average Image Size	1642 × 976	1640 × 968	1640 × 968
Number of Annotations	27745	3583	3833
Flame Annotations	23874	3100	3334
Smoke Annotations	3517	439	454
Small Flames	2167	263	276
Medium Flames	2123	258	268
Large Flames	595	73	84
No Flame	404	56	55
Warehouse Space	1190	239	256
Office Space	138	22	24
Indoor building	1946	238	241
Outdoor Building	580	79	74
Mall Space	5	0	2
Open Space	437	52	56
Unknown Space	180	20	26
Daytime	1861	254	228
Nighttime	611	66	95
Sunrise/Sunset	69	7	5
Unknown Time	2748	323	355
Containing Watermark	1270	154	152
Blurred Images	1836	218	234
Containing Other Light Sources	4248	515	553
Containing People	1721	222	216

3.2.2 Dataset preparation

To ensure high-quality data for our deep learning models, we developed a custom image annotation tool designed to centralize dataset management and optimize the labeling process. This system uses a server-hosted architecture in which annotators access their batch assignments via a web browser. A major advantage of this approach is that administrators maintain full oversight of the dataset. They can effectively manage the dataset by inspecting images and performing necessary actions, such as uploading or deleting samples. Efficiency is further improved through automated assignment and session persistence. When an annotator logs into the tool, a new set of images is automatically provided for processing. If a user leaves a session before finishing a batch, the system reloads all unsaved images so they can continue their work exactly where they left off.

The tool incorporates a randomized delivery mechanism to support a double-blind review process. To reduce bias in the initial labeling, experts manually review each annotation to confirm its accuracy. During this review stage, the experts make necessary corrections and adjustments to the labels. This ensures that the final ground truth is both precise and consistent throughout the whole dataset. Once annotations are finalized, the data can be exported in standardized formats such as COCO or YOLO. This enables seamless integration with various deep learning frameworks and training pipelines. This structured workflow ensures the final dataset is both reliable and consistent for the subsequent training phases. A brief preview of the dataset is shown in Figure 3.2.

Determining the presence of fire or smoke in an image can be challenging in certain scenarios. Fire and smoke exhibit diverse shapes, colors, and transparency characteristics, all influencing the area detected. In addition to the essential characteristics of flame and smoke, various factors, including low image quality, scale, motion blur, and occlusion, may obscure this assessment. This decision depends on human judgment due to the absence of an objective criterion for including or excluding a fire or smoke zone. Isolated small flame regions may occasionally emerge from the flickering of the primary fire source. The precise contour of each flame is nearly impossible to annotate due to its tendency to flicker between 2 and 10 Hz [19]. Before beginning the annotation process, guidelines were established to guarantee the quality and uniformity of annotations. The rules are as follows:

1. The polygon must match the flame's shape exactly, without straight lines or rectangles.
2. No new polygons should be made if minor objects (like window or door grills) are in front of the flame and/or smoke. Additional polygons must be added if a large object is in front of the smoke or flame.
3. The annotated polygon should include as many fire and/or smoke pixels as possible. It is recommended to annotate pixels inside the flames rather than annotating transparent border pixels or those outside the flames.
4. Segment overlaps should be avoided because it is impossible to determine which segment is in front.
5. Smoke is marked regardless of its transparency unless it covers the entire indoor environment.

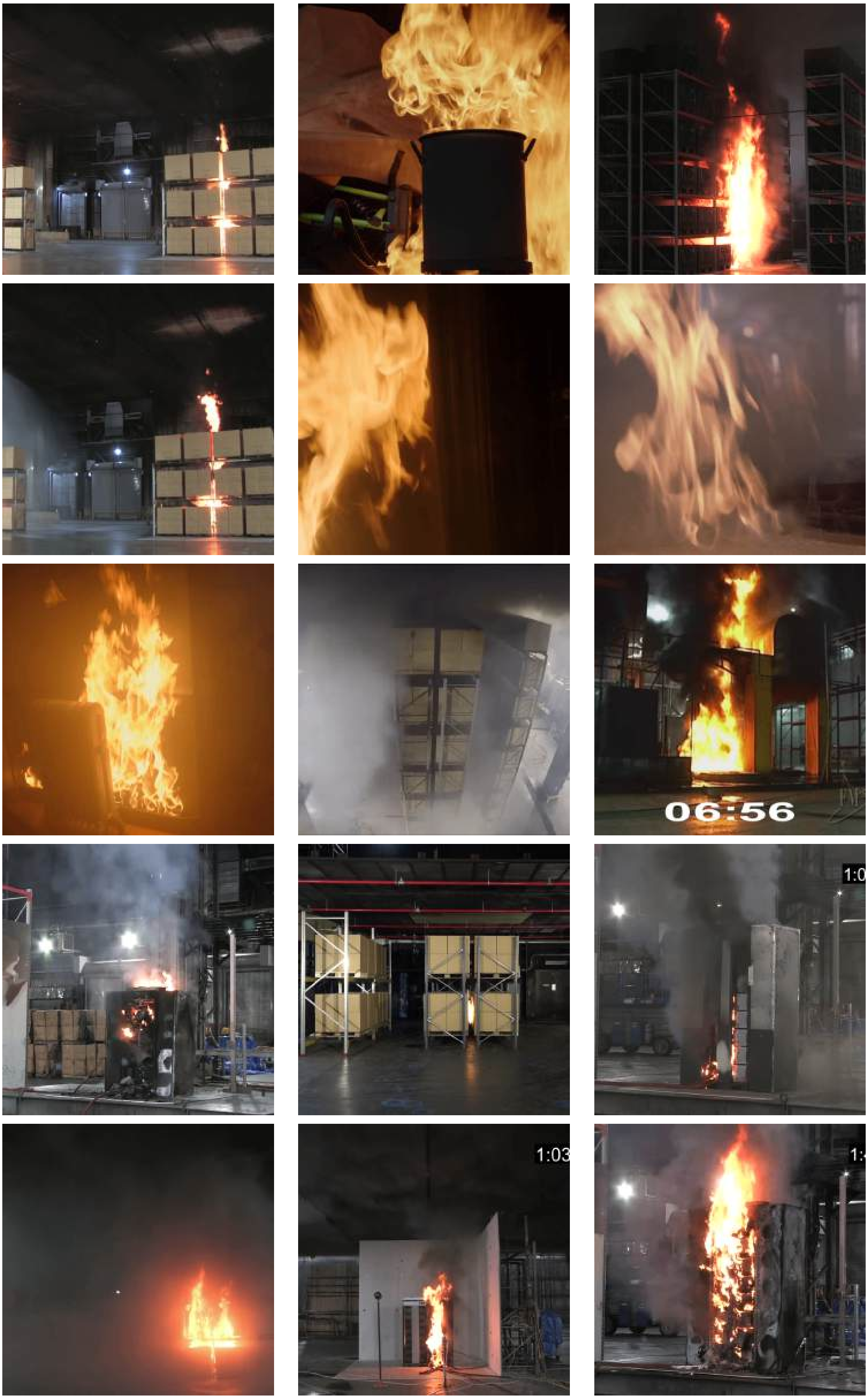


Figure 3.2: The visualization of RGB images in the utilized dataset. Image source: *Arlovic et al.* [11].

6. A polygon does not represent a reflected flame but rather one with the appropriate texture and color.
7. Reflections of flames and other light, lasers, light bulbs, lamps, and the glow of other light sources should not be annotated.

An additional metadata component is added to each image, and the goal is to evaluate how the model performs by adjusting and selecting specific parameters. During the image annotation process, annotators answer questions about the image context. Capturing more metadata alongside the usual image annotations helped us actually understand what is happening in the dataset. The collected information covers straightforward details about each image, making later analysis cleaner and more reliable. The informations obtained from annotators regarding a specific image include answers to the following questions:

- Verification of fire or smoke presence in the image.
- Analysis of the spatial context depicted in the image.
- Identification of visible flames in the image.
- A qualitative evaluation of the most significant flame in the image.
- Examination of the ambient lighting conditions depicted in the image.
- Assess for the presence of watermarks, such as timestamps or text that has been artificially inserted.
- Assessment whether image is computer-generated.
- Assessment of image blurriness, particularly about the clarity of fire contours.
- Assessment of the presence of additional light sources.
- Assess whether individuals are present in the image.

3.2.3 Model Training

The models were selected for their distinctive mechanisms and strong performance in different segmentation tasks. These include pyramidal feature extraction, traditional encoder-decoder architectures, attention-based enhancements, advanced spatial feature aggregation, and transformer-based representations. Our objective was to establish a comprehensive benchmark that simplifies the selection of suitable models for fire segmentation by integrating a diverse set of deep learning architectures. Each model was trained with images in three different input resolutions (256×256, 640×640, and 800×800 pixels) and three encoder backbones (EfficientNet-B4, EfficientNet-B7, and

ResNeXt-50 32×4d), resulting in 45 unique model configurations. These backbones were chosen for their proven effectiveness in prior studies, providing a balanced mix of architectures well-suited for fire segmentation tasks. Training was conducted on a high-performance workstation equipped with dual NVIDIA RTX A6000 GPUs, a Ryzen 9 5900X processor, and 128 GB of RAM. Ubuntu 24.04 LTS was utilized tohether with PyTorch deep learning framework.

The dataset was divided into three non-overlapping subsets: 80% for training, 10% for validation, and 10% for testing. Since the original images had higher resolutions than supported model input resolutions, each image was resized to preserve its aspect ratio and then zero-padded to match the training resolution. To increase data diversity and improve generalization, we applied extensive augmentation. Geometric transformations included rotation by ±5°, horizontal flipping, coarse dropout, and perspective warping. Color and noise transformations included color jitter, random brightness and contrast adjustments, Gaussian and ISO noise, and Gaussian blur. These augmentations were designed to represent realistic fire scenarios. Training was performed for a maximum of 200 epochs, with early stopping triggered if the validation loss failed to improve for 30 consecutive epochs. We used the AdamW optimizer [113] with a weight decay of 10^{-3}, which decouples weight decay from gradient updates. For learning rate scheduling, we adopted the OneCycleLR scheduler [114]. This scheduler increases the learning rate from an initial value to a predefined maximum, then gradually reduces it back to the initial value. Although the scheduler dynamically adapts the learning rate, it still requires an initial value to be specified. We experimentally tested initial learning rates ranging from 10^{-7} to 10^{-2} and selected 10^{-3} as optimal, with the maximum rate set to 10^{-2}. We compared Binary Cross-Entropy (BCE) and Mean Squared Error (MSE) for binary fire segmentation. In our experiments, BCE proved to be more suitable than MSE for modeling the discrepancy between predicted probabilities and ground-truth masks. We trained and evaluated at both high and low input resolutions. High resolution helped capture fine-grained fire patterns, while low resolution was used to assess applicability in robotics and IoT.

3.2.4 Benchmarking Models

Benchmark Metrics

The metrics employed in this study are widely recognized in image segmentation: Sørensen–Dice Score and Intersection over Union (IoU) score. The equation presented in Equation 3.1 shows the formulation of the Dice Score.

$$Dice = \frac{2TP}{2TP + FP + FN} \tag{3.1}$$

The IoU score is represented by the equation 3.2:

$$IoU = \frac{TP}{TP + FP + FN} \tag{3.2}$$

The number of true positives is represented by TP in both equations. The expression refers to the pixels that are classified as fire in the ground truth mask and those that are predicted as fire by the model. Conversely, FP denotes false positive, which occurs when the model inaccurately predicts that a pixel is on fire, and FN denotes false negative when the model fails to predict that the pixel is on fire. The IoU metric commonly penalizes single instances of misclassification much more than the Dice Score, even though both metrics measure a similar value. Consequently, the Dice Score provides a more comprehensive understanding of the average model performance. Nevertheless, both metrics encounter difficulties when the image only contains relatively small fire regions (a small number of true positives) and a model predicts a significant amount of false positives. In such instances, both metrics would yield mediocre values. Subsequently, we evaluated the Total error metric to determine the precise error of the employed models. The equation of the Total Error metric is shown in 3.3:

$$TotalError = \frac{FP + FN}{N_{px}} \tag{3.3}$$

The term N_{px} denotes the total number of pixels present in the image. Consequently, the total error is defined as the ratio of incorrectly predicted pixels to the overall pixel count in the image.

Evaluation of Benchmarked Models

To identify the best model, we adjusted the threshold that determines whether a pixel is classified as fire on the validation set. The final evaluation of the model performance was conducted on the test set, which contains images that were not included in the training or validation datasets. We examined a range of thresholds from 10% to 60% and selected the optimal threshold for each model. Results of each model are extensively analyzed in Table 3.2, where the best-performing models in each of the three resolutions are indicated in bold text.

When observing models, we initially considered the Dice Score, Total Error, and IoU score when selecting the best performing model. No strict guidelines exist on which metric is more important in image segmentation tasks. However, the Dice Score is more general in segmentation tasks, so we prioritized it over the IoU. The Total error measures absolute error and is placed second in priority, as it fundamentally measures a different type of error than the IoU and Dice Score. For the 256 × 256 resolution, the model with the best performance in terms of Dice Score, IoU Score, and Total Error was U-Net++. It achieved a Dice Score

Table 3.2: Benchmark results of the performance of each tested model on the validation and test datasets.

				Validation Dataset			**Test Dataset**		
Models	**Resolution**	**Encoder Backbone**	**Thres-hold**	**Dice Coeff.**	**IoU Score**	**Total Error**	**Dice Coeff.**	**IoU Score**	**Total Error**
U-Net	256x256	ResNeXt-50 32x4d	33%	0.811	0.858	0.00282	0.809	0.849	0.00304
U-Net++	256x256	EfficientNet B7	34%	**0.821**	**0.868**	0.00242	**0.820**	0.859	0.00250
Manet	256x256	EfficientNet B7	30%	0.810	**0.868**	0.00252	0.810	**0.861**	0.00252
FPN	256x256	ResNeXt-50 32x4d	30%	0.764	0.831	0.00252	0.766	0.829	0.00261
DeepLabV3+	256x256	EfficientNet B7	30%	0.774	0.811	0.00271	0.772	0.805	0.00278
SegFormer	256x256	MiT-B3	39%	0.688	0.746	**0.00209**	0.676	0.736	**0.00205**
U-Net	640x640	ResNeXt-50 32x4d	41%	**0.851**	0.878	0.00250	**0.852**	0.873	0.00276
U-Net++	640x640	ResNeXt-50 32x4d	41%	0.849	**0.882**	0.00236	0.847	**0.874**	0.00266
Manet	640x640	EfficientNet B7	40%	0.843	0.856	0.00287	0.839	0.846	0.00298
FPN	640x640	ResNeXt-50 32x4d	40%	0.846	0.872	0.00254	0.842	0.861	0.00280
DeepLabV3+	640x640	EfficientNet B7	41%	0.843	**0.882**	0.00239	0.838	0.870	0.00255
SegFormer	640x640	MiT-B3	46%	0.764	0.793	**0.00193**	0.759	0.790	**0.00173**
U-Net	800x800	ResNeXt-50 32x4d	40%	0.850	0.868	0.00279	0.848	0.855	0.00287
U-Net++	800x800	EfficientNet B7	50%	**0.855**	0.880	0.00240	**0.849**	0.862	0.00274
Manet	800x800	EfficientNet B7	50%	0.848	0.872	0.00235	0.844	0.862	0.00258
FPN	800x800	ResNeXt-50 32x4d	50%	0.845	0.858	0.00319	0.838	0.845	0.00328
DeepLabV3+	800x800	ResNeXt-50 32x4d	40%	0.842	**0.882**	0.00221	0.837	**0.870**	0.00244
SegFormer	800x800	MiT-B3	42%	0.771	0.810	**0.00162**	0.764	0.803	**0.00147**

of 0.820 on the test set, a Total Error of 0.00250, and an IoU Score of 0.859. Despite SegFormer achieving the lowest Total Error (0.00205) and MAnet achieving the highest IoU Score (0.0861), we selected U-Net++ because of its superior Dice Score and marginally lower IoU score compared to MAnet. At a resolution of 640 × 640, the optimal model is U-Net, achieving a Dice Score of 0.852 on the test set. The second-ranked model, U-Net++, obtained a Dice Score of 0.847 and a slightly superior IoU score of 0.874 compared to U-Net's score of 0.873. In the 800 × 800 image resolution, U-Net++ performed best, achieving a Dice Score of 0.849. DeepLabV3+ and SegFormer attained the greatest IoU score of 0.870 and the lowest total error of 0.00147, surpassing all other models at the same resolution. U-Net++ consistently exhibited superior performance across several resolutions, with the most notable improvement observed at 256x256 resolution. In summary, U-Net++ with the EfficientNetB7 encoder backbone performs the best at 256 × 256 resolution, followed by U-Net with ResNeXt-50 32x4d encoder backbone at 640 × 640 resolution, and U-Net++ with the EfficientNetB7 encoder backbone at 800 × 800. Although SegFormer's hierarchical Transformer encoder and efficient MLP decoder enabled it to achieve the lowest Total Error by minimizing absolute pixel-level mistakes, our benchmark prioritized the Dice Score as the primary evaluation metric, in which U-Net++ consistently outperformed SegFormer while also achieving competitive IoU results, thereby establishing U-Net++ as the superior model in this study.

3.3 F2M Model

The F2M model is designed to combine the five best-performing models, utilizing their complementary capabilities to improve segmentation performance. This research examined the assumption that an ensemble of the best-performing models will produce superior results compared to using only the best-performing model. This section offers an in-depth look at F2M architecture, bias-imputation strategy, and its influence on segmentation performance.

3.3.1 Architecture Overview

The network integrates various models to capture different feature representations, reduce individual model biases, and improve generalization across different contexts. Significant attention was placed on explainability, particularly uncertainty estimation, in developing the proposed model. Our main goal was to develop a compact and efficient model capable of learning from several segmentation masks, which also provide uncertainty in its decisions (through the Monte Carlo dropout). Estimating uncertainty is a crucial element of machine learning that enhances the reliability of the employed models and algorithms, facilitating improved interpretation of model confidence and identifying potential mistake areas [115]. Furthermore, this improves model robustness and guarantees dependable deployment in practical applications.

The F2M topology with the number of layers and layer sizes is shown in Figure 3.3, while the complete neural network can be expressed using Equation 3.4.

$$\hat{y} = CH(FEB(Y), \frac{\sum_{i=1}^{5} y_i}{5}) \tag{3.4}$$

where CH represents Concatenation Head, FEB is Feature Extraction Branch with respective 5-channel input denominated with Y, and the sum which represents Bias Branch over input data Y. The model was developed following the comprehensive research on various architectures, methods, and optimizations. Initially, we tried to build a single Feature Extraction Branch by combining the masks with 1 × 1 convolution layers, which yielded an inferior Dice Score. This concept was promptly discarded as a model reliant exclusively on 1 × 1 convolutions lacks sufficient power to derive new information from the input masks. Consequently, we initiated the experiment with 3 × 3 convolutions, which produced markedly superior outcomes. After evaluating several pooling methods, we selected Adaptive Max Pool 2D to reduce the dimensions of the layers. With Adaptive Max Pool 2D, the neural network was prompted to concentrate on the essential components of the input data. Inspired by the U-Net model, we incorporated skip connections within the Feature Extraction Branch (FEB) to mitigate spatial context loss during downsampling. Consequently, a large number of channels, some of which provided minimal to no valuable information, motivated us to integrate the Squeeze and Excitation block for adaptive recalibration of channel-wise features [112]. FEB's addition stabilized the neural network's training and improved its performance.

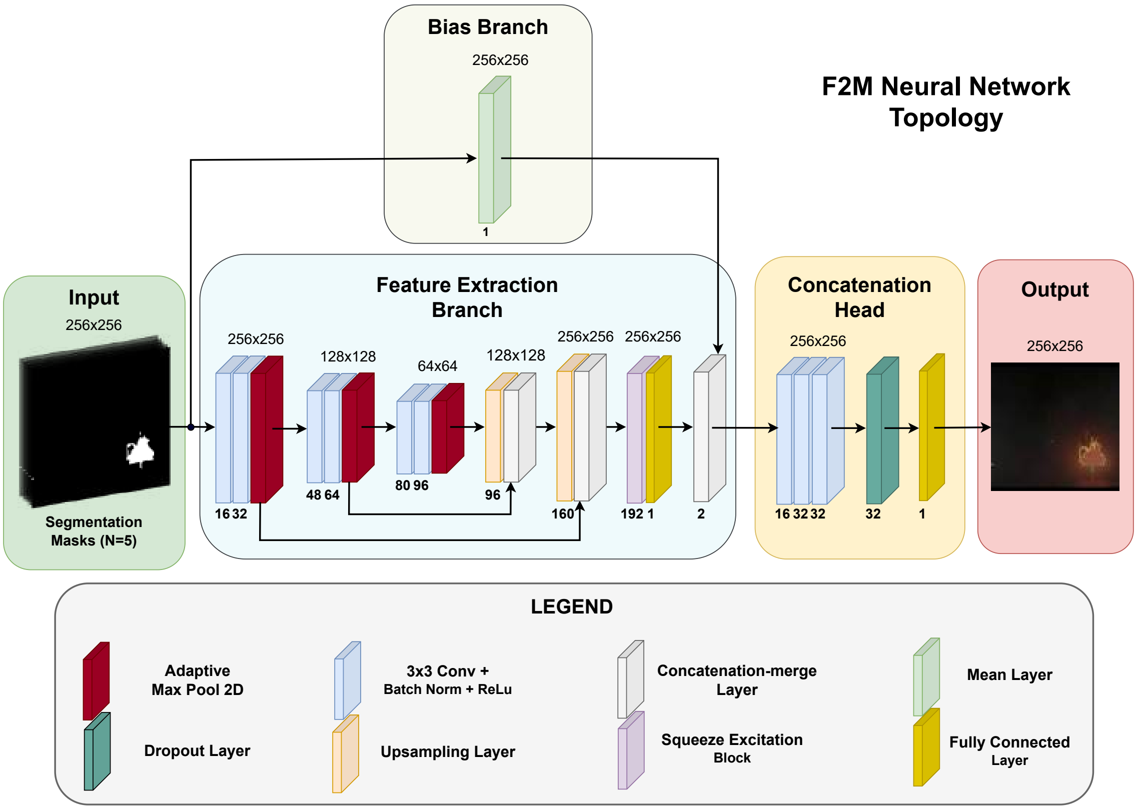


Figure 3.3: The architecture of the F2M network for an image size of 256x256 pixels, with layers adjusted for different resolutions. Image source: *Arlovic et al.* [11].

The Concatenation Head generated the final output of the F2M, which included dropout layers that functioned both as regularization and a foundation for Monte-Carlo dropout uncertainty estimation. Initially, we incorporated a dropout layer following each convolution layer to prevent the neural network from overfitting. Optimal results were achieved with a single dropout layer at the end of the neural network, with a dropout rate of 0.1. However, a range of dropout values from 0.1 to 0.5 were systematically evaluated. The most significant novelty of the F2M is the introduction of Bias branch, which regulates the bias of the extracted features. Applying the Shapely's [116] approach, which evaluated the features relative to the expected neural network output, we opted to employ the mean of the segmentation masks as the bias for the extracted features. With the implementation of the bias branch, the model converged approximately by the 60^{th} epoch and improved the Dice Score by approximately 1.5% for all evaluated networks. We utilized the same parameters employed for the training of segmentation models in the benchmark, except for the batch size. In the F2M training, we selected a batch size of 8. Batch sizes ranging from 4 to 64 were evaluated, and we found that training loss and validation accuracy remained stable across these settings. Since larger batch sizes offered no performance gain but increased memory requirements, we fixed the batch size at 8.

3.3.2 Ensemble Evaluation

We evaluated F2M performance across three resolutions against the best model for each resolution. The comparison is shown in Table 3.3, where the best results for each metric are highlighted in bold.

Table 3.3: The F2M model performance comparison against the best benchmarked model for each resolution.

			Validation Dataset			Test Dataset		
Models	**Resolution**	**Threshold**	**Dice Score**	**IoU Score**	**Total Error**	**Dice Score**	**IoU Score**	**Total Error**
U-Net++	256x256	34%	0.821	0.868	0.00242	0.820	0.859	0.00250
F2M	256x256	37%	**0.855**	**0.892**	**0.00179**	**0.853**	**0.883**	**0.00188**
U-Net	640x640	41%	0.851	0.878	0.00250	0.852	0.873	0.00276
F2M	640x640	41%	**0.862**	**0.899**	**0.00181**	**0.862**	**0.899**	**0.00198**
U-Net++	800x800	50%	0.855	0.880	0.00240	0.849	0.862	0.00274
F2M	800x800	37%	**0.860**	**0.914**	**0.00165**	**0.861**	**0.904**	**0.00182**

The F2M model outperformed other neural networks across all resolutions. The Dice Score on the test set exhibited a difference of 4.02% between U-Net++ and F2M, with the most substantial increase in performance observed at the lowest resolution (256 × 256). In addition, the Dice Score increased by 1.17% and 1.41% at resolutions of 640 × 640 and 800 × 800, respectively. It is important to note that the IoU score and Total Error metric exhibited greater improvements across all resolutions. F2M obtained improvements in IoU scores of 2.79%, 2.97%, and 4.87% at resolutions of 256 × 256, 640 × 640, and 800 × 800, respectively. The Total Error was reduced by 24.8% (from 0.00250 to 0.00188) at 256 × 256, by 28.3% (from 0.00276 to 0.00198) at 640 × 640, and by 33.57% (from 0.00274 to 0.00182) at 800 × 800. The results from the validation set displayed a comparable trend to those from the test set, suggesting that the model did not experience overfitting.

Table 3.4 presents the average inference time for each model in the research, including F2M and their corresponding parameter sizes. The results demonstrate that the computational cost imposed by the F2M network is minimal relative to the total inference duration of individual neural networks. On average, F2M experiences an additional computational cost of around 1.3ms compared to the slowest neural network tested in the benchmark. Furthermore, input image resolution substantially influences the inference time because higher resolutions result in extended processing times. Furthermore, F2M contains a much reduced quantity of parameters in comparison to other networks. SegFormer, an attention-based neural network, does not demonstrate a significant decrease in parameter count compared to convolutional designs.

Table 3.4: Comparison of model performance: Average Inference Time and Number of Parameters per sample.

Models	**Resolution**	**Encoder Backbone**	**Time [ms per sample]**	**Number of parameters**
U-Net	256x256	ResNeXt-50 32x4d	5.715 ± 0.359	31 992 977
U-Net++	256x256	EfficientNet B7	15.459 ± 0.745	112 220 049
MAnet	256x256	EfficientNet B7	29.923 ± 1.994	77 998 421
FPN	256x256	ResNeXt-50 32x4d	5.475 ± 0.558	25 587 905
DeepLabV3+	256x256	EfficientNet B7	26.663 ± 1.900	65 106 273
SegFormer	256x256	MiT-B3	36.684 ± 4.057	47 224 002
F2M	256x256	None	1.328 ± 10.128	333 467
U-Net	640x640	ResNeXt-50 32x4d	13.119 ± 0.048	31 992 977
U-Net++	640x640	ResNeXt-50 32x4d	35.923 ± 0.215	48 457 617
MAnet	640x640	EfficientNet B7	36.818 ± 0.406	77 998 421
FPN	640x640	ResNeXt-50 32x4d	10.671 ± 0.032	25 587 905
DeepLabV3+	640x640	EfficientNet B7	48.073 ± 0.433	65 106 273
SegFormer	640x640	MiT-B3	38.991 ± 3.102	47 224 002
F2M	640x640	None	1.358 ± 3.622	333 467
U-Net	800x800	ResNeXt-50 32x4d	18.877 ± 0.057	31 992 977
U-Net++	800x800	EfficientNet B7	69.021 ± 0.522	68 163 553
MAnet	800x800	EfficientNet B7	57.191 ± 0.482	77 998 421
FPN	800x800	ResNeXt-50 32x4d	15.074 ± 0.043	25 587 905
DeepLabV3+	800x800	ResNeXt-50 32x4d	17.965 ± 0.102	26 149 457
SegFormer	800x800	MiT-B3	52.672 ± 1.381	47 224 002
F2M	800x800	None	1.681 ± 4.671	333 467

F2M component evaluation

In this study, we introduced a novel bias branch, which we hypothesized increases model performance. To validate our hypothesis, we conducted a comparative analysis between two versions of F2M: without the Bias Branch (only with the Feature Extraction Branch) and the complete F2M network (Feature Extraction Branch + Bias Branch). The obtained results are summarized in Table 3.5. Across 256×256 resolution, complete F2M achieved higher Dice and IoU scores on both the validation and test datasets, with improvements ranging from 0.5% to 1.1%, respectively. However, the model without Bias Branch outperformed by a margin in Total Error. On the test dataset, complete F2M showed a substantial reduction in Total Error, with a decrease of 20.8% from 0.00250 to 0.00198. Notable increase ranging from 1.7% to 3.1% in Dice Score and IoU at a resolution of 640×640. At a resolution of 800×800, complete F2M showed a 0.7% increase in Dice Score, a 2.7% increase in IoU Score, and a decrease in Total Error by 25.4% (0.00244 to 0.00182) on the test dataset. Our experiments demonstrate that adding the Bias Branch improves the effectiveness of the F2M model and facilitates the model training process. Figure 3.4 presents the Dice score, IoU, and Total Error curves for the validation dataset at a resolution of 640×640. Compared to the model without a Bias Branch, the complete F2M graphs display significantly reduced fluctuations and are noticeably smoother. This pattern is uniform across all tested resolutions.

Table 3.5: Importance of Bias branch: Results of F2M with and without Bias branch

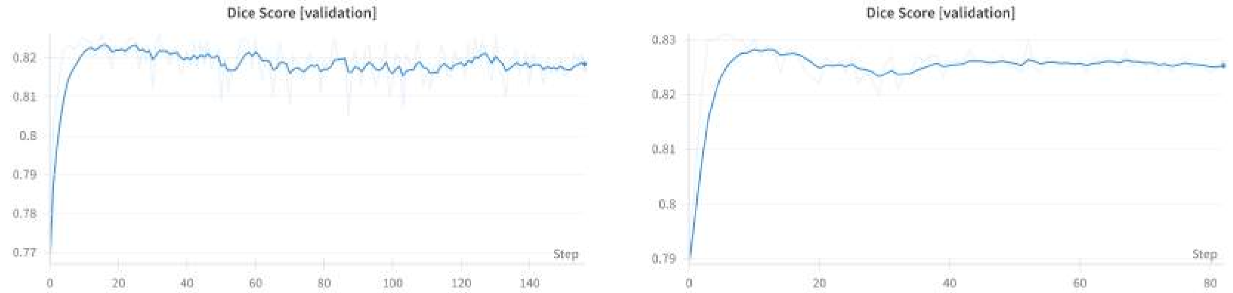
			Validation Dataset			**Test Dataset**		
Resolution	**Model**	**Thres-hold**	**Dice Score**	**IoU Score**	**Total Error**	**Dice Score**	**IoU Score**	**Total Error**
256x256	Feature Extraction Branch	33%	0.850	0.884	0.**00178**	0.848	0.874	**0.00186**
	F2M	37%	**0.855**	**0.892**	0.00179	**0.853**	**0.883**	0.00188
640x640	Feature Extraction Branch	38%	**0.863**	0.881	0.00228	0.856	0.868	0.00250
	F2M	41%	0.862	**0.899**	**0.00181**	**0.862**	**0.899**	**0.00198**
800x800	Feature Extraction Branch	33%	**0.860**	0.893	0.00213	0.854	0.877	0.00244
	F2M	37%	**0.860**	**0.914**	**0.00165**	**0.861**	**0.904**	**0.00182**

Uncertainty evaluation

To evaluate the uncertainty F2M predictions, we implemented Monte Carlo dropout by processing an identical input image through the model 100 times. The results are shown in Figure 3.6, with the final output of the F2M model presented in the last column. In the final output, the standard deviation (STD) of the average predicted mask from 100 F2M tests with dropout enabled is included. These values show the model's confidence and prediction at each pixel. White areas in the STD mask indicate substantial uncertainty, while black areas indicate that the model is entirely certain of its decision. To illustrate the standard deviation of F2M prediction alongside the mask for the user examining the final results, we have introduced a 3D graph representing the predicted mask on the X and Y axes. In contrast, the standard deviation derived from the uncertainty estimation for each pixel is positioned on the Z-axis. Figure 3.5 shows the Monte Carlo dropout uncertainty assessment over 100 iterations on the same input image. Uncertainty is higher at the borders of predicted regions, suggesting that the model is confident about the main fire areas but less certain about where the fire boundaries end.

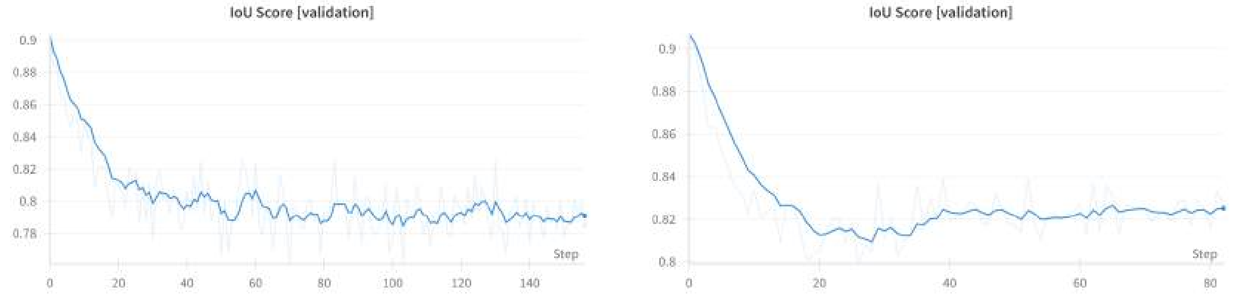
3.3.3 Discussion

In contrast to previous fire detection studies that mostly employ U-Net segmentation models (e.g., [4]), our study expands upon these methods. The U-Net++ architecture was utilized in this research because it outperformed not only the U-Net model but also the other benchmarked models in this study, particularly in the context of complex segmentation tasks. This performance gain, while modest, can be attributed to U-Net++'s redesigned skip connections and nested architecture, which provided more effective feature propagation. As a result, it achieved slightly higher Dice Score results, as well as improvements in Total Error and IoU, compared to the other benchmarked models. Although these modifications are claimed to improve model performance, their impact on our indoor fire detection dataset remains unclear. Our benchmarking experiments yielded varied outcomes, with some models surpassing those documented in prior studies, while others did not reach comparable performance. Thus, identifying the "optimal" model is a challenging task that requires extensive experimentation across several datasets, a process impeded by the restricted availability of publicly accessible fire segmentation datasets. Our novel strategy presents F2M, a methodology aimed at improving existing research and capitalizing on forthcoming advancements in segmentation models. The justification for selecting and evaluating the models presented in this study is based on their demonstrated



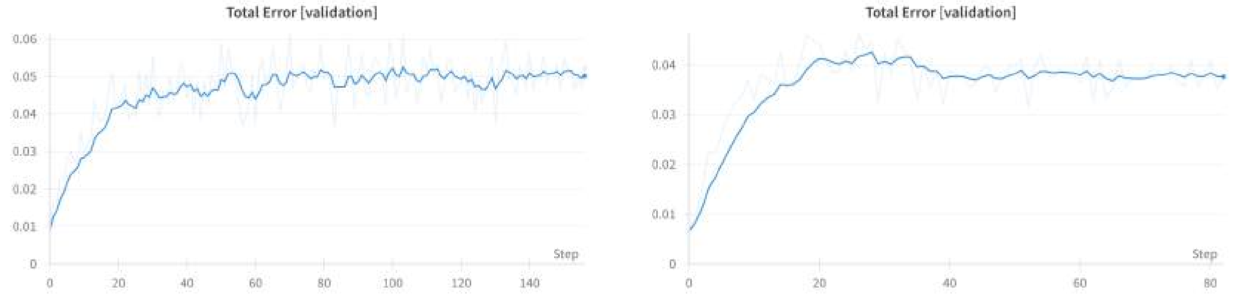
Without Bias
Branch 640x640 (Dice Score)

F2M 640x640
(Dice Score)



Without Bias
Branch 640x640 (IoU Score)

F2M 640x640
(IoU Score)



Without Bias
Branch 640x640 (Total Error)

F2M 640x640
(Total Error)

Figure 3.4: Dice score progression during the validation phase for F2M and the Feature Extraction Branch at a resolution of 640 × 640. The dark blue line represents the smoothed outcome (60% smoothing), whereas the light blue line illustrates the recorded metrics. Image source: *Arlovic et al.* [11].

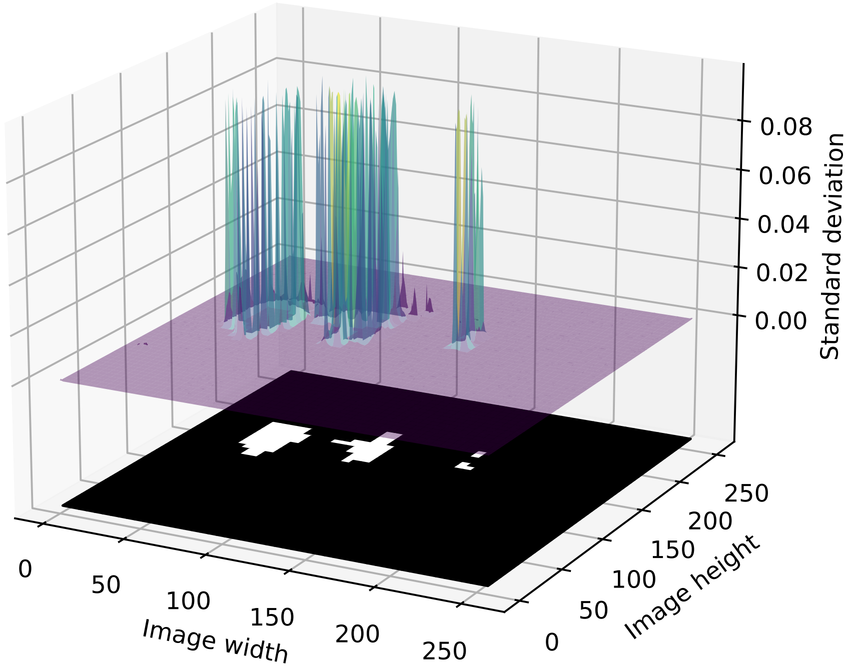


Figure 3.5: Graph showing the Monte Carlo dropout uncertainty across 100 iterations of the same input image. Image source: *Arlovic et al.* [11].

efficacy in fire detection and associated segmentation tasks, including applications in domains such as autonomous driving, remote sensing, and medicine. Introducing an innovative neural network that can leverage existing models and replace them with superior segmentation models would provide a sustainable framework for future fire detection applications.

The F2M model created in this research can substantially improve fire detection and its segmentation. The proposed Bias Branch enhances performance and stabilizes the training process. The uncertainty estimation mechanism allows the development of an automated picture annotation tool, optimizing the annotation process for extensive datasets. However, a significant constraint of the model is its dependence on predictions of five distinct models, which necessitates considerable computing resources. The computational demands of this model would be substantial on an embedded device; nevertheless, there are two methods to address this challenge. The initial approach involves reducing the model size by pruning and by utilizing various optimization algorithms which maintain model performance while improving efficiency. The alternative approach involves migrating the models to the cloud, facilitating the transfer of images from sources such as cameras or robots via the internet. Cloud computing facilitates the utilization of numerous GPUs, delivering sufficient processing capacity to execute all models simultaneously. Employing one or both of these technologies

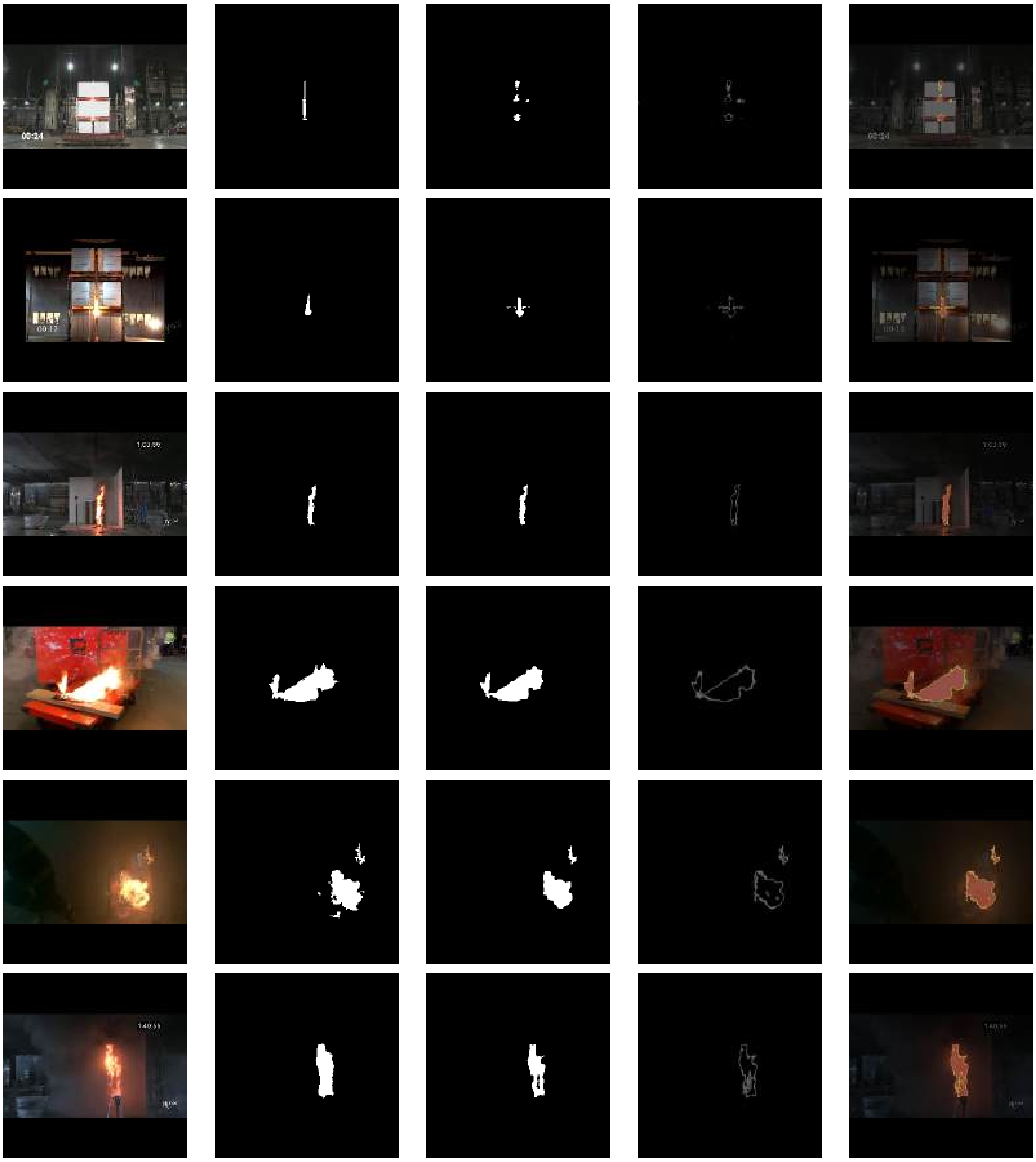


Figure 3.6: The visualization of fire detection results illustrates the RGB input image, the ground truth mask, the predicted mask at 33% threshold, the standard deviation (STD) mask derived via the Monte Carlo dropout uncertainty estimate, and the final amalgamated mask. The STD mask delineates regions of ambiguity in the forecasts. Image source: *Arlovic et al.* [11].

might make real-time fire detection viable. The additional computational burden imposed by F2M for real-time detection is negligible, as evidenced in table 3.4, requiring merely $333,467$ extra parameters.

3.4 Conclusion

Timely fire detection and fast alert dissemination are critical for minimizing risks to human life and property. In response, numerous fire monitoring systems have been developed by researchers, employing both sensor-based and image-based technologies. Image-based systems present considerable advantages compared to sensor-based systems, as they deliver more comprehensive information on fire more quickly - mainly location, intensity, and spread. Accurately identifying the shape and boundaries of flames in images presents challenges due to background interference, varying fire sizes, and objects that may resemble flames. This study presents extensive benchmarks utilizing CNNs for fire segmentation in indoor environments. We expanded upon previous studies by employing various advanced neural network architectures and presenting a new model called the Feature Merge Model, which integrates the outputs of multiple models to enhance accuracy and interpretability. The model's explainability was achieved using the Monte Carlo dropout mechanism. This technique applies dropout during inference to obtain multiple predictions and their associated uncertainties, thereby categorizing each pixel as fire or non-fire with a measure of model confidence.

The F2M model consistently outperformed other neural networks in all tested resolutions, showcasing its adaptability. F2M outperformed U-Net++ by 4.02% in the Dice Score at the narrowest resolution (256×256), resulting in the most substantial improvement. In addition, F2M demonstrated modest yet discernible enhancements in the 640×640 and 800×800 resolutions, with increases of 1.17% and 1.41%, respectively. The model also exhibited a significant decrease in error rates, with a 24.8% decrease at a resolution of 256×256 and even larger decreases at higher resolutions. It is important to note that the model's performance on the validation set was consistent with that of the test set, indicating that the model generalized well and did not overfit.

4

Impact of Synthetic Data on Fire Segmentation Models

This chapter introduces SYN-FIRE, a novel synthetic dataset focused on fire detection in industrial environments. Compared to conventional methods, deep neural networks have demonstrated superior performance in fire detection. However, their effectiveness relies on the use of gold-standard datasets, which are essential for developing robust semantic segmentation models. In Chapter 2, we discussed how the performance of deep learning models relies on the availability of high-quality annotated data. We also examined the challenges researchers face when creating such datasets. These include high annotation costs, legal restrictions, and the general scarcity of data across many scientific fields, especially in the field of indoor fire detection. Additionally, we reviewed several synthetic datasets that have demonstrated promising results and highlighted their potential to help overcome these limitations. To address these challenges, we developed SYN-FIRE, which contains 2,000 annotated synthetic fire images generated using NVIDIA Omniverse. This chapter begins by explaining the dataset generation process and reviewing current methods used to produce synthetic data. We then present two ablation studies that evaluate the impact of different synthetic-to-real data ratios on the performance of our segmentation model.

This chapter is structured as follows. Section 4.1 outlines the primary objectives of the research. Section 4.2 discusses the methods that can be employed for developing synthetic data, followed by Section 4.3, which discusses the synthetic dataset of indoor fires in the industrial environments, comprised of 2,000 labeled images. Section 4.4 describes the experimental configuration and explains the network training methodology. Section 4.5 outlines the ablation studies performed in this research, accompanied by the related results, statistical analysis, and an evaluation of model generalization using previously unseen real-world data. Section 4.6 presents research results for both ablation studies. Section 4.7 provides statistical analysis of results using the Paired T-Test. Section 4.8 outlines results of model generalization trained on synthetic data, and tested on real unseen data. Finally, Section 4.9 presents the final research observations.

4.1 Research Objectives

This research aims to create a synthetic dataset that replaces or supplements real data for the semantic segmentation of indoor fires in industrial environments. Deep learning models typically necessitate substantial datasets to attain effective generalization. The lack of annotated datasets constrains the training and evaluation process of deep learning models. To mitigate this challenge, we developed SYN-FIRE, an entirely synthetic dataset of indoor fires utilizing NVIDIA Omniverse. Employing U-Net++ as the baseline architecture, we trained the models on the SYN-FIRE synthetic dataset and evaluated their effectiveness using four publicly available datasets of real fire images. The study comprises two ablation experiments: one substitutes portions of real data with synthetic data, while the other integrates varying amounts of synthetic samples into real data. The following outlines our research objectives:

1. To generate a new synthetic dataset by incorporating a mixture of industrial environments using the NVIDIA Omniverse.
2. To determine whether synthetic data is a viable alternative to real data and whether it has a beneficial impact on the model training.
3. To assess whether the performance of segmentation models is improved when synthetic data is incorporated in conjunction with real data.

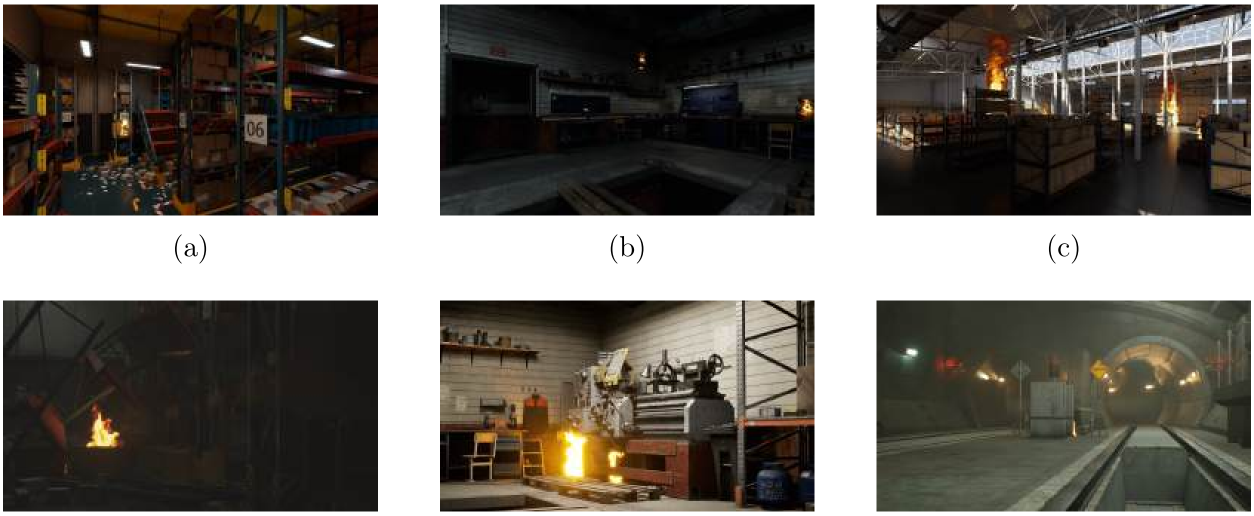
4.2 Existing methods for synthetic data generation

This section discusses methods for creating synthetic data for deep neural network training. Synthetic data is a preferred option for various applications because it accelerates the training, testing, and deployment stages, thereby enhancing the efficiency and effectiveness of deep learning model creation [117]. It also reduces the time required for image capture and labeling, preserving user privacy and security and reducing the risk of disclosing sensitive information [118]. Diverse techniques can be used for synthetic data generation, including 3D computer graphics engines such as Blender [119], Unreal Engine

5 [120], and NVIDIA Omniverse [121], as well as deep learning models such as Stable Diffusion [122], Generative Adversarial Networks (GANs) [123], and Variational Autoencoders (VAEs) [124]. The quality of the generated images is often influenced by multiple factors, including the ability to achieve realistic lighting and textures, the design of effective prompts, and the hyperparameter optimization.

4.2.1 Image Generation Using 3D Software

The increased advancement of GPUs has enhanced the realism of scenes produced in 3D applications like Blender, Unreal Engine 5 and NVIDIA Omniverse [125]. 3D tools require precisely made 3D models and environments to produce synthetic images that closely mimic real images. An in-depth understanding of scene lighting is essential for designers, as it directly influences the fidelity of the visual depiction of the given scenario [12]. Synthetic images generated by the NVIDIA Omniverse and Unreal Engine 5.3.2 are shown in Figure 4.1.



(a) (b) (c)

(d) (e) (f)

Figure 4.1: Comparison of rendered scenes in NVIDIA Omniverse (top row) and Unreal Engine 5.3 (bottom row). Image source: *Arlovic et al.* [12].

Blender

Blender is a free and open-source software extensively utilized to create 3D graphics [119]. Blender, mostly utilized in computer graphics, allows the creation of photorealistic renders, incorporating sophisticated features such as complex lighting, depth of field, volumetric effects (e.g., fog and mist), and physically accurate reflections and refractions [126]. Blender offers comprehensive support for the entire 3D pipeline which encompasses modeling, rigging, animation, simulation, rendering, compositing, motion tracking, video editing, and game development [119]. The software suite provides multiple rendering engines, each designed for different levels of realism and computational efficiency. Cycles [127] can produce highly photorealistic graphics, making it a favored engine for scenarios requiring precise simulations of lighting and material interactions. As a physically-based path-tracing renderer, Cycles simulates light behavior by tracing multiple rays per pixel throughout a scene. These rays interact with surfaces based on their material properties, producing realistic effects such as reflection, refraction, and light scattering [127]. The number of samples per pixel significantly influences the accuracy and clarity of an image generated by Cycles [127]. Low sample counts produce noise, especially in regions dominated by indirect light, such as within shadows, glossy reflections, and caustics. Meanwhile, more samples reduce noise and sharpen features. Due to its ability to accurately replicate complex light interactions, Cycles is widely used in visual effects, architectural visualization, and scientific simulations, where realism and physical accuracy are essential. EEVEE [128] was created as a real-time alternative for Cycles, providing improved versatility in its rendering pipeline. The main goal is to reduce frame rendering times, hence enhancing animation production efficiency. EEVEE prioritizes speed and interactivity while also accommodating high-quality physically based rendering (PBR) materials [128]. EEVEE uses rasterization and screen-space techniques instead of simulating individual light rays as path tracing does. Rather, it determines the visibility of surfaces from the camera's perspective and applies a variety of algorithms to approximate the interaction of light with these surfaces and materials. Moreover, experienced users can exploit the full Python programming capabilities available through the Blender API. This powerful capability enables the modification of the program using the specialized tools, which can also ease the integration with deep learning technologies [129, 130]. Using custom plugins such as the Blender Annotation Tool [129], researchers can automate the process of generating annotated synthetic datasets.

Károly et al. [131] improved the Blender Annotation Tool to create a novel annotation method for synthetic datasets. Their methodology automates the creation of multimodal annotations, encompassing segmentation masks, depth maps, surface normals, and optical flow across several scenarios. Consequently, they created the Synthetic Multimodal Video Benchmark dataset, comprising of 1585 images from seven distinct scenarios spanning various domains, including underwater environments, two-dimensional animation, and photorealistic scenes. Detecting vehicles using aerial and satellite imagery is necessary for traffic prediction, vehicle counting, and velocity assessment applications. In response to the absence of appropriate publicly available datasets, Orić et al. [132] created a synthetic dataset containing 5000 labeled images with a resolution of 2048×2048 pixels. To obtain realistic results, the authors created authentic road scenes in Blender using mapping services such as Google Maps. They then populated these environments with 3D car models. Džijan et al. [133] developed a modular approach to generate single-view synthetic depth images from 3D point clouds of indoor environments to train object detectors. Their research showed that object detectors trained exclusively on synthetic data exhibit poor performance compared to those trained on real data. Furthermore, they demonstrated that pretraining with synthetic data and later fine-tuning with real data slightly increases the network's performance.

Unreal Engine

Graphics and simulation capabilities of Unreal Engine 5 [120] improve synthetic data creation for deep-learning applications [134]. Unreal Engine's Nanite technology offers an internal mesh format and rendering technique that adeptly manages extremely complex geometry. This technology allows developers and researchers to create realistic and highly detailed 3D environments in real-time. In contrast to conventional rendering methods, Nanite does not depend on levels of detail (LOD) to regulate object complexity. It dynamically streams geometric data, loading exactly what is required at any moment. This method significantly streamlines the creation process while preserving superior visual fidelity and performance [135]. Unreal Engine 5's real-time rendering and dynamic lighting solutions, such as Lumen, enable the creation of synthetic images and videos that closely resemble real-life visuals [136]. Lumen is a real-time global illumination system that offers high-quality lighting in 3D environments [135]. Eliminating time-intensive pre-calculations enables developers and researchers to simulate indirect illumination promptly. The system relies on voxel cone tracing to compute this indirect lighting, offering real-time feedback during scene creation. It also supports advanced features such as accurate reflections, refractions, and soft shadows while dynamically adapting to any changes in lighting conditions. These features enable the building of immersive, lifelike settings without compromising performance [137]. Path tracing is an alternative lighting solution in Unreal Engine that uses a physically based rendering technique to simulate the complex interactions of light rays with surfaces and materials. This primarily offline rendering method is not limited by Lumen's geometric simplifications and lighting approximations, yielding more realistic and physically precise images, although at a somewhat higher computational cost [136]. Unreal Engine also supports the light baking option. This method includes precomputing lighting computations and storing them in specialized textures called lightmaps, significantly reducing rendering complexity during runtime. A significant development in this domain is GPU Lightmass (GPULM), which enhances the conventional CPU-based Lightmass Global Illumination through GPU acceleration. GPULM proficiently computes complex lighting interactions from stationary lights, saving the precomputed outcomes in created lightmap textures directly applied to the scene geometry. GPU Lightmass markedly decreases the creation and computation duration for lighting data compared to CPU-based alternatives, achieving velocities akin to distributed CPU-based builds. Furthermore, GPULM enables an interactive workflow, allowing developers to execute real-time scene modifications and swiftly recalculate lighting data, a capability unattainable with the conventional CPU-based Lightmass system [138].

Due to the scarcity of real-world fire datasets, researchers are increasingly developing synthetic datasets to facilitate the training of high-performing fire detection models. Hu et al. [139] introduced a synthetic dataset called FireFly, developed for ember detection in wildfires, produced using Unreal Engine 4. The dataset comprises of 19,273 images, including 16,904 positive samples containing embers and 2,369 negative samples missing embers. The FireFly dataset was evaluated using four object detection methods. Models trained on the FireFly dataset demonstrated an enhancement of up to 8.57% in mean average precision (mAP) for real wildfire scenarios, in contrast to models trained solely on a limited real-world dataset. Fernando et al. [140] used Unreal Engine 5 to develop a Simulated Wildfire Images for Fast Training (SWIFT) dataset. The dataset consists of 70,000 images and 15 videos of wildfire scenarios collected from multiple viewpoints. Moreover, the authors assessed the dataset on three deep-learning models for wildfire classification.

NVIDIA Omniverse

The NVIDIA Omniverse [121] is a versatile computer graphics platform designed to enhance collaborative operations in both industrial and creative domains. It can generate physically accurate synthetic 3D scenes that are automatically annotated and ready to use for learning machine-learning models. NVIDIA Omniverse features two primary rendering methods: RTX - Real-Time and RTX - Interactive (Path Tracing), which are designed to meet the diverse requirements of visual fidelity and performance. Real-Time Mode utilizes advanced real-time ray tracing techniques to handle more complex geometries and achieve improved material quality relative to conventional rasterization methods. This method involves separating the lighting calculations into separate passes, including ray-traced ambient occlusion, direct lighting with ray-traced shadows, indirect diffuse global illumination, reflections, translucency, and subsurface scattering. The renderer can eventually integrate these elements into the final image by cautiously denoising each pass. This method may introduce approximations and optimizations that significantly reduce physical realism, yet it delivers a responsive, high-frame-rate experience essential to interactive apps and real-time workflows. In contrast, the main objective of the interactive mode is to attain the highest level of visual accuracy and photorealism. By methodically tracing light paths across the scene, each frame is generated to encompass contributions from all possible interactions. After a single path-tracing run that gathers accumulated illumination data, the renderer utilizes an AI-accelerated denoiser, followed by post-processing procedures like bloom and tone mapping. This mode generally achieves more precise lighting and material representations; nevertheless, it requires increased computational resources, often resulting in lower framerates than RTX -Real-Time mode. Depending on their project needs, these rendering modes allow researchers to choose between interactive performance and unparalleled image quality. Researchers can use the

Replicator SDK to generate physically accurate, labeled 3D synthetic data to train and validate AI perception models. The Replicator SDK can generate synthetic data for multiple perceptual tasks, such as object detection, segmentation, pose estimation, and depth estimation, utilizing a physically correct camera placement [141].

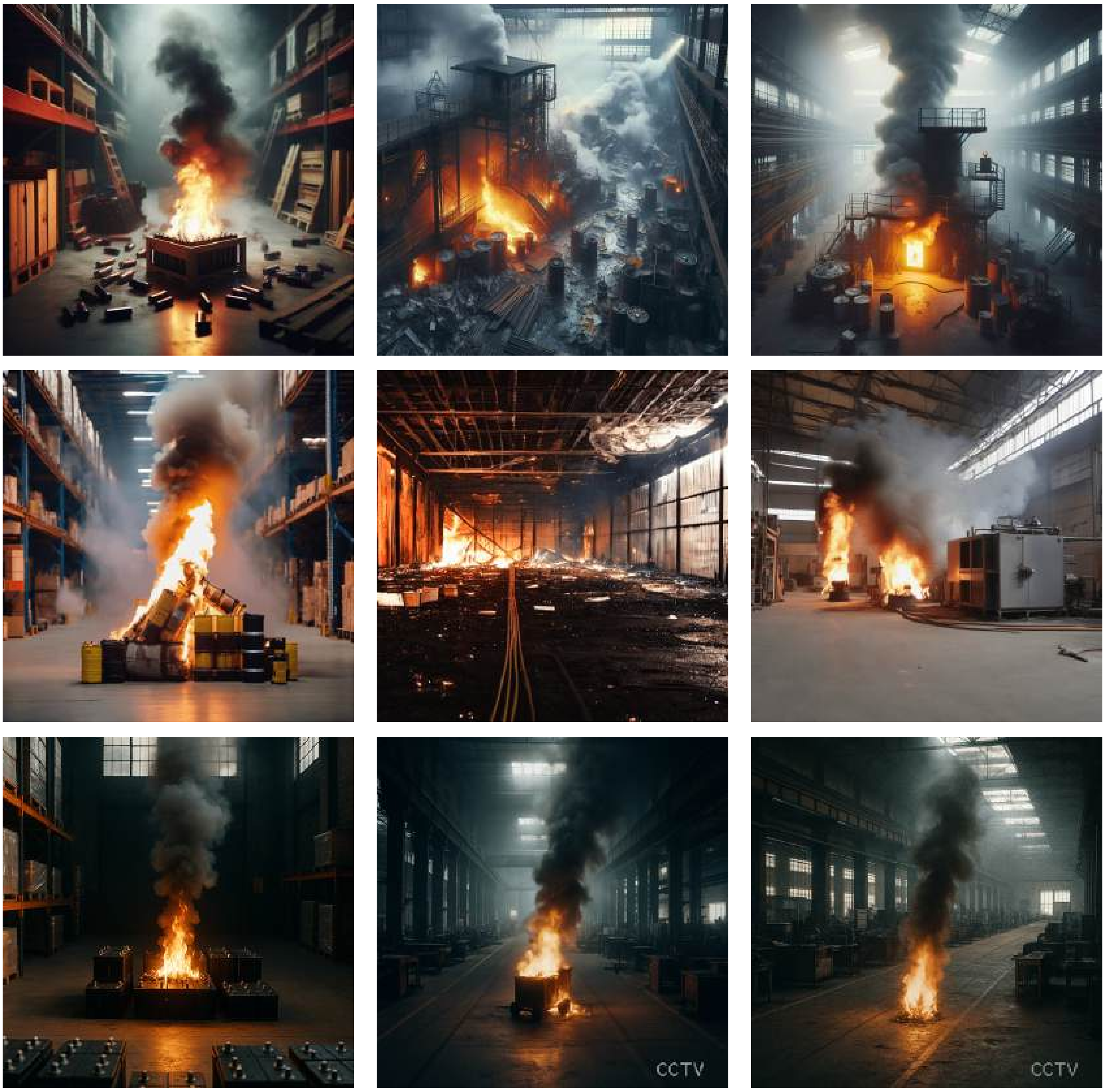
NVIDIA Omniverse is a new software, and a few datasets have been generated. Conde et al. [142] introduced a methodology executed within the NVIDIA Omniverse platform to generate and validate synthetic datasets based on real-world scenarios. The authors employed Replicator SDK to produce RGB images accompanied by labels for semantic segmentation or object detection. In our research, we used NVIDIA Omniverse to develop the SYN-FIRE dataset, the first synthetic dataset of indoor fires. Using this software, we created multiple industrial environments. We initiated fires with the NVIDIA PhysX physics engine, which produced realistic fire and smoke under physically grounded parameters, enabling high-quality imagery for future research and development of robust deep learning models for fire detection.

4.2.2 Image Generation Using Deep Learning Networks

A Generative Adversarial Network (GAN) is a deep learning framework composed of two neural networks, the generator and the discriminator, which are trained concurrently in an adversarial manner [123]. The generative model ($G(x)$) attempts to learn and replicate the distribution p_g over data x. The primary objective is to generate new data samples by transforming random noise $p_z(z)$ into meaningful outputs that are not direct duplicates of the authentic data but rather capture its key characteristics. The discriminator model ($D(x)$) receives the generator output and real images from the training dataset [123]. $D(x)$ represents the probability that x came from the training dataset rather than p_g. The discriminator model is trained to maximize the probability of assigning the correct label to training examples and samples from G [143].

Diffusion models are generative models designed to generate data mimicking training dataset samples. Their main goal is to estimate the underlying data distribution $p(x)$ by progressive denoising a normally distributed variable through a sequence of iterative enhancements [12]. This approach is mathematically analogous to acquiring the inverse of a predetermined Markov chain with a predefined length T [144]. Their capacity to produce high-quality and various outputs has led to widespread adoption in text-to-image generation, including advanced models like DALL·E and Stable Diffusion. In recent years, diffusion models have gained recognition as a viable alternative to GANs, mainly because of their ability to address specific limitations in GAN architectures. The primary challenge of GANs is mode collapse, a case where the model produces a restricted range of outputs, reducing its efficiency in practical applications [145]. Diffusion models fundamentally generate variations of samples via their iterative denoising process, rendering them a resilient and dependable option for various generating tasks.

In recent years, there has been a substantial increase in the use of artificial intelligence to generate synthetic images. Islam i Zhang [146] introduced a GAN-based approach for creating synthetic brain PET images. They addressed the challenge of restricted medical images for normal control (NC), mild cognitive impairment (MCI), and Alzheimer's disease. Visual inspection and quantitative metrics like peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) were used to evaluate synthetic images. Adding synthetic data to a diagnostic classifier enhanced Alzheimer's disease diagnosis from normal control cases. Abduljawad i Alsalmani [147] evaluated the ability of diffusion models to generate satellite, Synthetic Aperture Radar (SAR), and passive microwave images, which are essential but challenging to obtain through conventional methods. Dall-E 2 consistently generated the most realistic and accurate results among the AI models the authors tested, particularly in generating visible-band images. Nathanail [148] successfully generated 1200 realistic synthetic images across six distinct fossil categories by fine-tuning the Stable Diffusion model in conjunction with the DreamBooth technique. In the domain of geoscience, this dataset is a valuable resource for the training and evaluation of image classification and object detection models, facilitating tasks such as the automated interpretation of depositional environments. Figure 4.2 displays the synthetic images produced by the DALL-E 3 in the first row, the Stable Diffusion XL v1.0 in the second row, and the ChatGPT 4o in the third row. Each row corresponds to one model, and each column to a shared text prompt. **Prompt 1**: A small fire with smoke in a warehouse containing batteries, no people, high-quality, 8k, indoor, filmed on a Sony A7iii, 50mm, f2.8, realistic appearance similar to an iPhone photo. **Prompt 2**: A small fire with smoke in a factory scene, high-quality, 8k CCTV photo. **Prompt 3**: A small fire with smoke in a factory scene, CCTV, no people, high-quality, 8k, indoor.



Prompt 1 *Prompt 2* *Prompt 3*

Figure 4.2: Qualitative comparison of images generated by three diffusion-based models: DALL·E 3, SDXL 1.0, and ChatGPT 4o. Image source: *Arlovic et al.* [12].

4.3 SYN-FIRE Dataset

Due to the private nature of industrial environments and privacy concerns, the field of fire detection has a limited number of publicly available datasets. Most publicly available fire detection datasets [104, 105, 149, 150] are based on outdoor areas, but not on indoor industrial environments. In this study, we developed a new synthetic dataset of indoor fires in industrial environments, SYN-FIRE [5], using the NVIDIA Omniverse platform. Certain assets were developed exclusively in Omniverse, while others were obtained from the Unreal Marketplace and transformed into Omniverse using the NVIDIA Unreal Engine 5 connector. In these environments, a drone's route is simulated as it moves from a higher viewpoint to a lower one, with variable light conditions. Additionally, the camera angles are designed to simulate security camera viewpoints. The virtual camera used a perspective projection with an effective focal length of about 18.15 millimeters and sensor apertures of 20.955 by 15.2908 millimeters, which yields fields of view of approximately 60 degrees horizontally and 45.7 degrees vertically, with depth of field disabled. The camera was autonomously rotated and moved for each frame using Omniverse Replicator SDK, resulting in the capture of images and the assignment of semantic annotations to specific objects. The SYN-FIRE dataset comprises of 2,000 labeled images, each 1920 by 1080 pixels, depicting simulated indoor industrial fires across five distinct settings, with labels marking the fire in each image [5]. We used the PhysX fire particle system, which is configured to capture indoor fire behavior with realistic fidelity, using a temperature parameter near 0.15, a fuel fraction around 0.8, a temperature coupling of 10, and a moderate fuel coupling. To simulate indoor conditions, we enabled burn and lowered the smoke density, thereby aligning the plume behavior with confined-space dynamics. This setup produces sustained combustion, credible buoyant rise, visible smoke stratification beneath ceilings, and natural plume meandering with convincing light attenuation, which together yield sequences that align well with observed indoor fire dynamics. Collision primitives were introduced for the pallets and boxes to eliminate subsurface combustion and ensure flames evolve along the exterior boundaries of objects.

4.3.1 Environments

Five indoor industrial environments were constructed to span representative layouts and operating conditions. Each scene targets factors relevant to fire detection, including aisle geometry, occlusion from shelving and machinery, material diversity, reflective surfaces, clutter density, and the presence of moving equipment. Illumination varied by fixture type, intensity, and placement, and included simulated firelight, resulting in a broad range of contrasts and color temperatures. Camera viewpoints emulate surveillance and aerial inspection, with elevations ranging from overhead to eye level, and oblique angles that create long sight lines. These choices yield complementary difficulty profiles that support generalization across diverse indoor industrial sites.

The following subsections provide a detailed description of the five environments: Modular Warehouse, Sorting Warehouse with Conveyor Belts and Robots, Garage, Warehouse with Various Sections, and Metro Maintenance Station. Each environment is presented in its own subsection.

Modular Warehouse

A warehouse environment was created to replicate the characteristics of a real-world industrial environment. Utilizing a customized plugin within the NVIDIA Omniverse platform, this environment was entirely generated using procedural methods. The creation process involved the structured integration of NVIDIA's proprietary assets to maintain alignment with spatial and material characteristics commonly found in real-world warehouse facilities. Additionally, dynamic elements such as fire and smoke were also incorporated into the environment through procedural generation techniques. Each iteration of generating the environment introduced modifications to both lighting conditions and the visual and structural variations of the assets used. This iterative variability was essential for generating a diverse dataset of warehouse scenes under varying operational and atmospheric conditions. Approximately 20 distinct asset types were employed during the environment generation process. These assets included common warehouse components such as boxes, shelving units, and forklifts. Each asset was varied in geometry and placement across iterations to enhance the realism and complexity of the generated scenes. The procedural method ensured that every version of the warehouse environment presented a unique spatial arrangements and lighting setups. Figure 4.3 presents sample images contained in SYN-FIRE dataset.

Sorting Warehouse With Conveyor Belts & Robots

The sorting warehouse environment was created to simulate a realistic industrial logistics setting using NVIDIA Omniverse. Within this scene, shelves, robots, and conveyor belts are manually positioned to resemble the structure of an actual facility. Although these main components are arranged by hand, fire and smoke elements are generated procedurally. Each iteration looks different, with fire and smoke behaving and appearing in new ways, making the results feel more realistic. Lighting is deliberately changed each iteration in addition to the artificial sources (e.g., lighting from the robots). This variety helps create a more diverse dataset, which is useful for testing how models perform under different visual conditions. Sample visuals from SYN-FIRE's Sorting Warehouse with Conveyor Belts & Robots environment are shown in Figure 4.4.

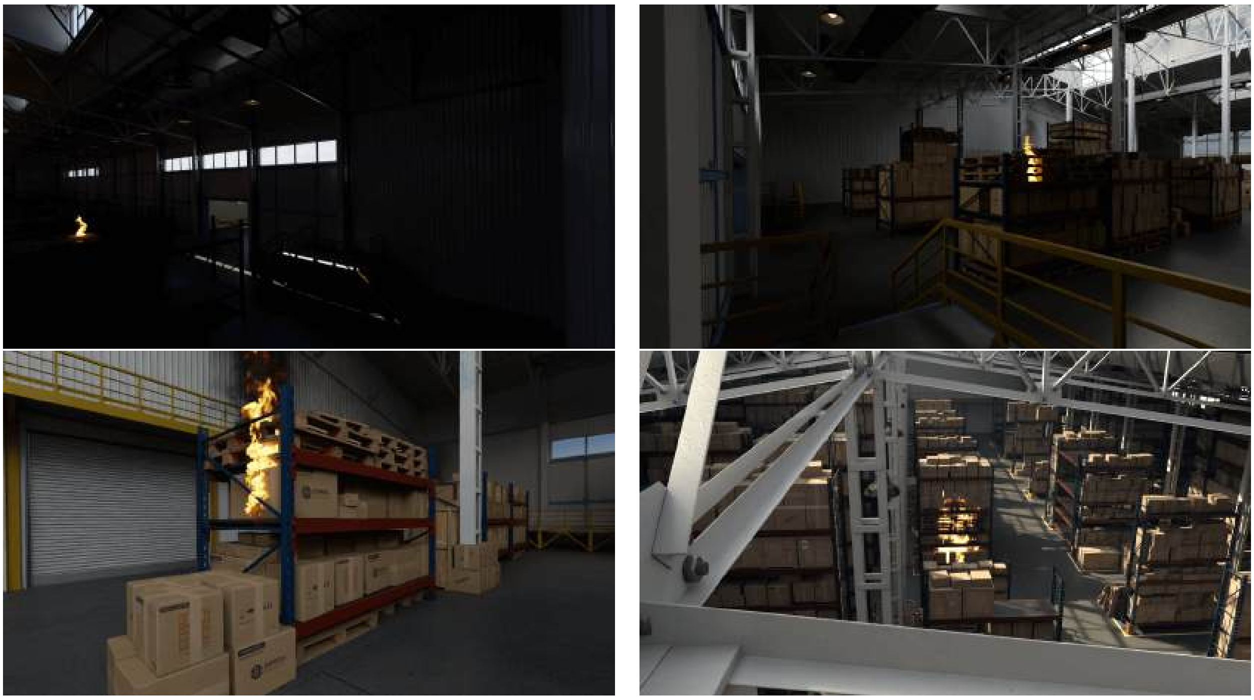


Figure 4.3: Synthetic image samples for the Modular Warehouse environment.

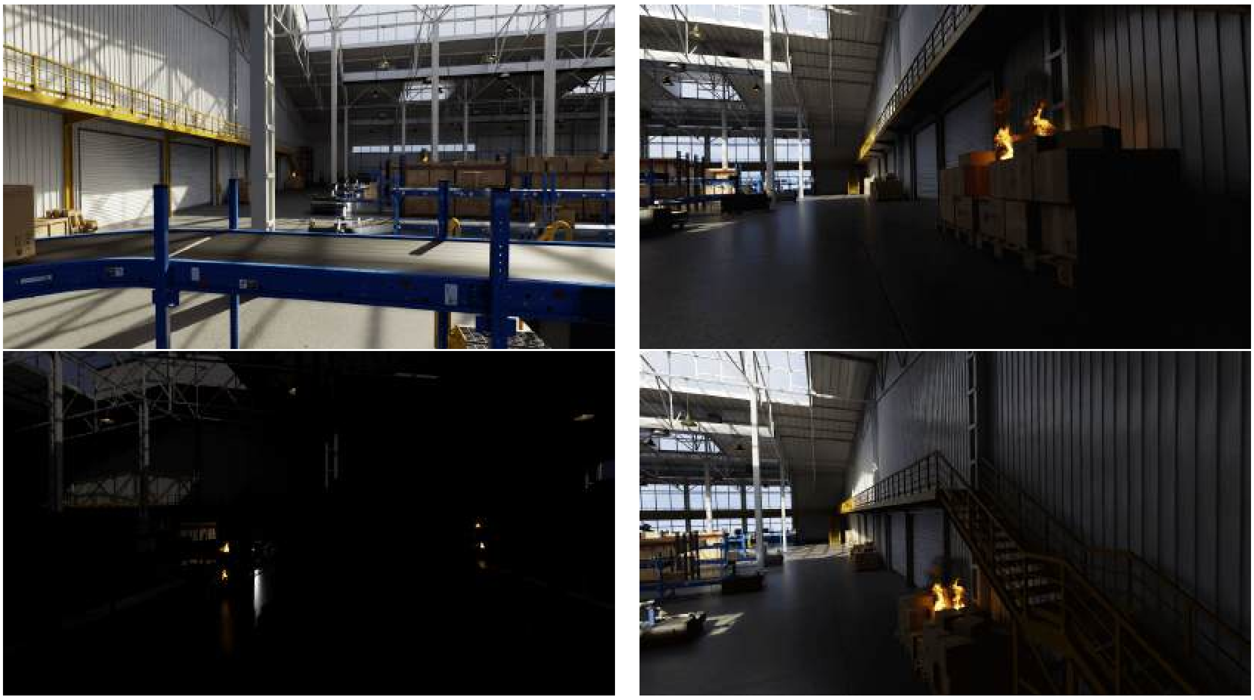


Figure 4.4: Synthetic image samples for the Sorting Warehouse With Conveyor Belts & Robots environment.

Garage

We used a garage environment obtained from the Epic Games Marketplace [151]. Although it was created for Unreal Engine 5, we successfully imported it into NVIDIA Omniverse using NVIDIA plugins. The garage contains storage cabinets, tools, and other essential items that are typically found in a functional garage. These objects consist of toolboxes, tires, paint cans, wall-mounted storage systems, workbenches, and smaller hand tools that are scattered across the garage. The arrangement of objects in the room is intentionally designed to resemble a realistic and lived-in workspace, with a particular emphasis on physical authenticity, wear, and clutter. All elements within the environment, including props and particle effects such as fire, were manually positioned. This approach enables a higher level of artistic control and ensures visual authenticity. The images in the dataset are rendered with and without interior lights. Figure 4.5 displays image samples from SYN-FIRE's Garage environment.

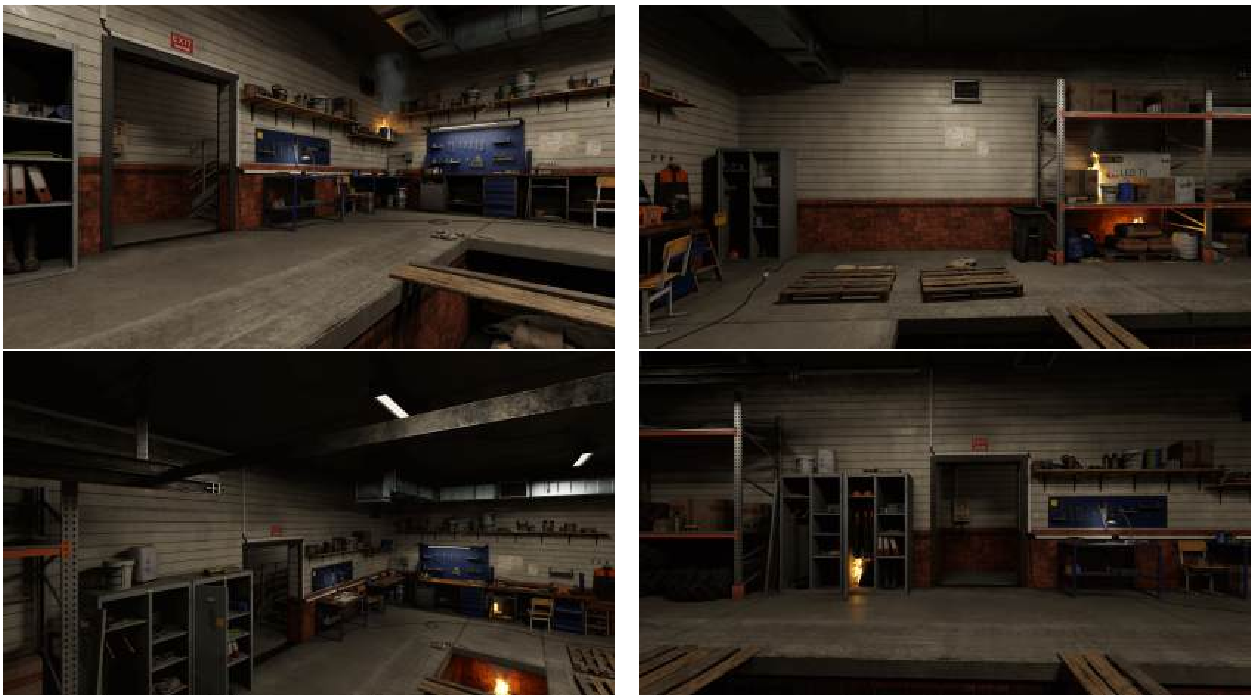


Figure 4.5: Synthetic image samples and their annotations for the Garage environment.

Warehouse With Various Sections

The multi-compartment warehouse environment is a customized scene with multiple interconnected sections, each with its unique asset arrangements and shelving configurations. This environment, purchased from the Epic Games Marketplace, offers a high degree of visual fidelity and realism. In contrast to the Modular Warehouse and Sorting Warehouse with Conveyor Belts & Robots, this warehouse employs customized warehouse assets rather than NVIDIA assets. Every compartment has been organized differently, with shelves positioned in various places and sizes. Some compartments appear organized with a regular warehouse layout, while others are disorganized, with boxes, papers, and other items scattered throughout the room. Every object in the scene, including shelves, boxes, crates, and ambient elements like fire, was placed manually. This deliberate positioning enabled greater control over the visual composition of each area, allowing each compartment to have a distinct character and function. Image samples from SYN-FIRE's Warehouse with Various Sections environment are shown in Figure 4.6.

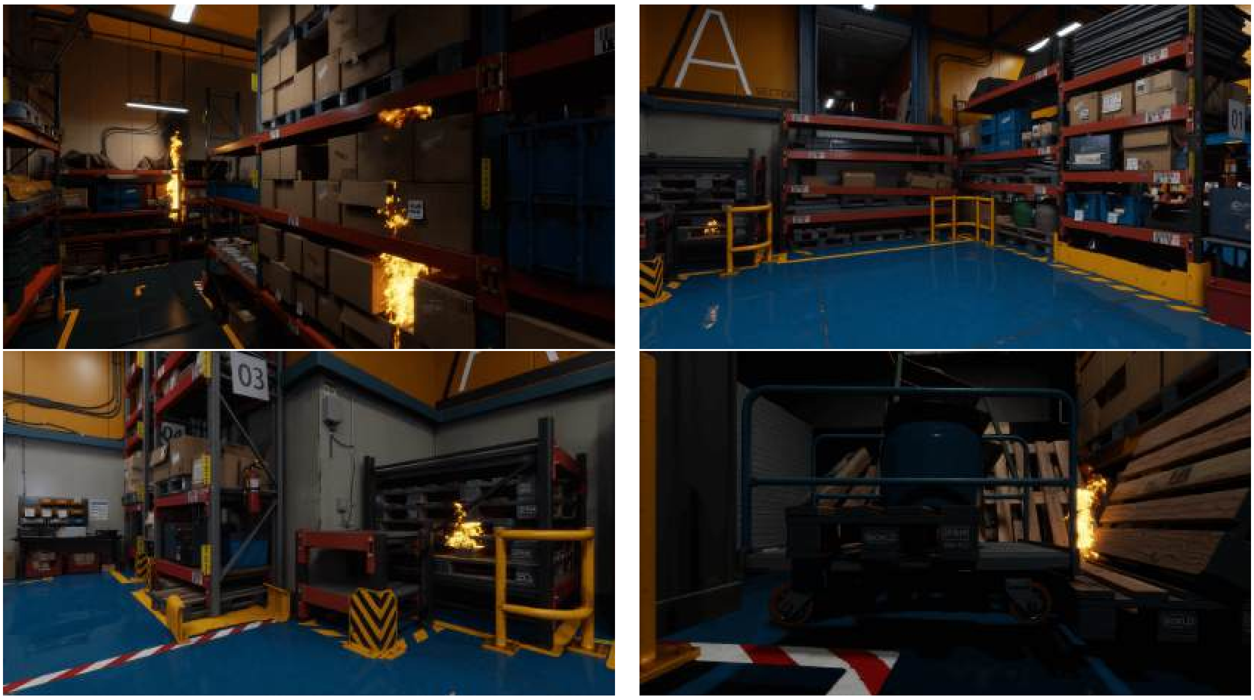


Figure 4.6: Synthetic image samples and their annotations for the Warehouse With Various Sections environment.

Metro Maintenance Station

The underground metro station features two parallel tunnels lit by warm orange lighting embedded inside the tubes, creating a soft, focused glow. The rest of the station remains largely in shadow, with minimal natural light entering through small upper windows. All environment objects were placed manually to ensure precise scene composition. Fire and smoke were then added as separate, clearly defined elements to simulate realistic combustion events. This setup is designed specifically for a synthetic fire dataset, allowing models to learn the visual distinction between ambient orange lighting and actual fire. Figure 4.7 illustrates sample images from SYN-FIRE's Metro Maintenance Station environment.

For annotating generated images, we employed the U-Net++ model trained on real images. A human-in-the-loop (HITL) approach was introduced to ensure accurate pixel-level annotations. By involving human intelligence in the annotation process, its input supported the system in the handling of uncertain or complex data, resulting in more precise annotations. Automation became more adaptable and reliable as it was enhanced by expert insight. The quality of annotated data has been improved by the integration of human expertise with machine learning. Although the HITL approach offers distinct benefits, such as flexibility, speed, and accuracy, it also has certain disadvantages. Relying

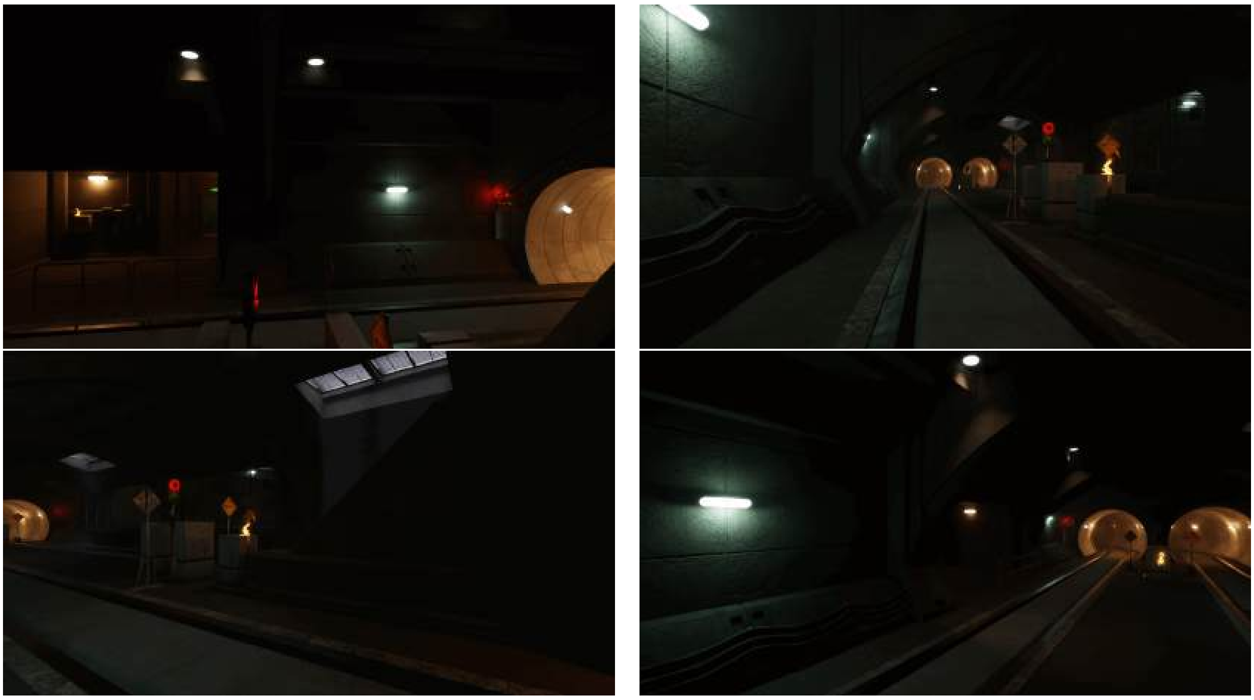


Figure 4.7: Synthetic image samples and their annotations for the Metro Maintenance Station environment.

on human input can result in inconsistencies and a slowdown of the annotation process, as different individuals may make different decisions. Furthermore, scaling HITL systems can be resource-intensive, particularly when manual evaluation of large quantities of data is necessary. The HITL approach in the annotation process, which involves a single human expert overseeing the procedure, substantially accelerates the annotation timeline despite these challenges.

4.3.2 Dataset Statistics

The SYN-FIRE dataset comprises of 2,030 labeled images that depict simulated indoor industrial fires in five distinct environments. All images in the dataset are 1920 × 1080 in resolution and have been split into two subsets: 1,402 for training and 601 for validation. The scene's complexities are diverse since every place varies in time of day, camera perspective, and fire characteristics. The dataset shows a substantial imbalance, with warehouse scenarios represented in 89.22% of the images, which is 74.36 times more frequent than the least common Metro Station scenario. Notably, the Modular Warehouse makes up 80.55% of the dataset, while the Metro Station contributes only 1.08%. To capture a variety of lighting conditions, 27.93% of the images were taken at night or during sunset. The analysis of fire occurrences reveals that flame sizes vary significantly, with an average diameter of 2,979.85 pixels and an average of 2.37 fire instances per image.The analysis of fire sizes and counts per image for all five scenarios and the total number of images per scenario are summarized in Table 4.1.

Table 4.1: Statistics for the SYN-FIRE dataset, highlighting the number of images per scenario.

Scenario	**N Images In Scenario**	**Avg. Flame Size**	**Avg. N Of Fires On Images**
Modular Warehouse	1636	2985.48 ± 2422.394	2.45 ± 2.166
Sorting Warehouse With Conveyor Belts & Robots	96	584.21 ± 170.939	1.42 ± 0.795
Garage	100	1645.96 ± 658.509	1.94 ± 1.398
Warehouse Divided Into Various Sections	176	1645.96 ± 658.509	1.94 ± 1.398
Metro Station	22	439.46 ± 65.357	1.65 ± 1.493

4.4 Implementation Details

In this section, we provide a brief overview of network implementation and training for the U-Net++ network used in our experiments. Following that, a description of the real dataset used to assess the impact of synthetic datasets on model performance was provided.

4.4.1 Experimental Setup

To evaluate the impact of synthetic data on model performance, we trained the U-Net++ network at resolutions of 128×128, 256×256, and 512×512 to enable seamless integration into the Internet of Things (IoT) networks for deployment on compact cameras. U-Net++ [94] is an encoder-decoder network featuring densely interconnected and hierarchically nested decoder sub-networks. These are designed to minimize discrepancies between the high-resolution representations of the decoder and the spatial features captured by the encoder. The resnext50_32x4d backbone was trained from scratch for 200 epochs in conjunction with an early stopping mechanism to mitigate model overfitting with a 30-epoch patience. We opted to train from scratch to avoid any biases that might arise from using pretrained weights. Automatic Mixed Precision was also used to optimize memory usage and speed training. Moreover, to mitigate the risk of overfitting and improve generalizability, the AdamW optimizer was implemented with a weight decay of 10^{-3}. Lastly, the Binary Cross Entropy loss function and backpropagation measured the divergence between predicted segmentation maps and their corresponding ground truth masks.

4.4.2 Real Data Datasets

To ensure the reproducibility and reliability of our research results, we tested models using publicly available fire datasets suitable for semantic segmentation. This cross-dataset evaluation allowed us to assess the practical utility of synthetic data for fire detection in real-world scenarios. A brief description of used datasets is given in the concluding part of the subsection. The main characteristics of the datasets are summarized in Table 4.2.

Dataset	Purpose	N Images in training set	N Images in validation set	N Images in test set
Corsican FireDB [104]	**Wildfire**	624	171	340
FLAME [107]	**Wildfire**	1402	300	301
FireBot [19]	**Indoor**	1402	601	1986
BowFire [105]	**Wildfire**	180	22	24
SYN-FIRE	**Indoor**	1402	601	-

Table 4.2: Statistics of real data datasets that utilized in the experiment.

In addition to the SYN-FIRE dataset, we also considered the FireBot dataset, since SYN-FIRE was designed with its structure and objectives in mind. However, the FireBot dataset is not publicly available due to copyright restrictions and was not used in our experiments. Instead, we relied on publicly available datasets, including Corsican FireDB, FLAME, and BowFire, which are commonly used for outdoor fire segmentation. We aimed to investigate whether a synthetic dataset such as SYN-FIRE could deliver comparable or improved performance in classification and segmentation tasks. The *FireBot* [19] dataset was developed to advance research on fire detection in indoor industrial environments, with a focus on image segmentation tasks. The dataset comprises 12,000 RGB images, all annotated at the pixel level, to enable comprehensive model training and evaluation. The *Corsican Fire Database* [104] comprises of 500 RGB images of fires, 100 RGB and NIR images captured simultaneously, and five sequences of RGB and NIR image pairs. The NIR images in this set are obtained with a longer exposure time, enhancing the brightness of fire regions and allowing segmentation through basic image processing algorithms. All images are annotated at the pixel level and appropriate for developing semantic segmentation models. The *FLAME dataset* [107] comprises of annotated images of regulated fires within a pine forest in Arizona. The dataset consists of images from RGB and IR cameras. The dataset is intended for image classification and segmentation applications. It consists of 39,375 labeled RGB images in the training subset and 8,617 in the test subset for image classification. Moreover, 2,003 images have pixel-level annotations, making them appropriate for training and evaluating image segmentation algorithms. The *BoWFire* [105] dataset was created to facilitate research in fire detection from stationary images, emphasizing the combination of color and texture data. The dataset has 226 images depicting real fire incidents, including structural fires, vehicular collisions, wildfires, and civil disturbances. The dataset is labeled to enable the classification and segmentation of fire images.

4.5 Experiments

To evaluate the potential advantages and disadvantages of utilizing synthetic data for fire detection, we trained various U-Net++ models on the previously described datasets. During the study, two research hypotheses were formulated to guide the experimental evaluation:

1. Synthetic data is a feasible alternative to real data and positively impacts model training.
2. The integration of synthetic data enhances the effectiveness of segmentation models when utilized with real data.

We evaluate these hypotheses with two ablation studies conducted on each dataset.

4.5.1 Ablation Study 1

The objective of this ablation study was to assess the impact of the correlation between synthetic and real data on model performance by replacing a portion of real data with synthetic images. The training and validation datasets comprise both real and synthetic data. The number of images in the observed dataset determines the number of real images N_r and synthetically generated images N_s. The ratio of images sampled from N_s and N_r in the training subset is determined by the parameter α. Consequently, the following equation is used to show the total number of images in the training subset N_t:

$$N_t = \alpha \times N_r + (1 - \alpha) \times N_s \tag{4.1}$$

With α set at 0.2, the training subset was generated by randomly picking 20% of images from real data and 80% from synthetic data. The subset contained a total of 1,402 images from the Flame dataset. Figure 4.8 shows the data allocation flowchart for the training subset.

4.5.2 Ablation Study 2

The objective of this ablation study was to evaluate the impact of synthetic data on the model's effectiveness when combined with real data. The following equation is used for calculating the number of images in the training subset:

$$N_t = N_r + \beta \times N_s \tag{4.2}$$

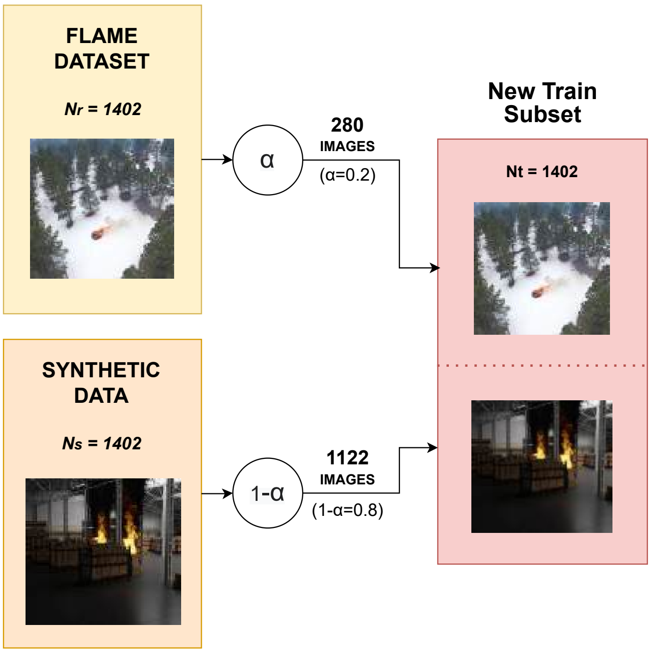


Figure 4.8: The flowchart depicts the process of data allocation for the training subset in the first ablation study when the $\alpha = 0.2$. Image source: *Arlovic et al.* [5].

The quantity of real data is denoted by N_r, the quantity of synthetic data is denoted by N_s, and the proportion of synthetic data in the training subset is determined by β. The training subset was generated by randomly selecting 10% of synthetic data and combining it with 100% of real data, with β set to 0.1. A total of 1,822 images comprised the training subset from the FireBot dataset. The data allocation flowchart for the training subset is depicted in Figure 4.9.

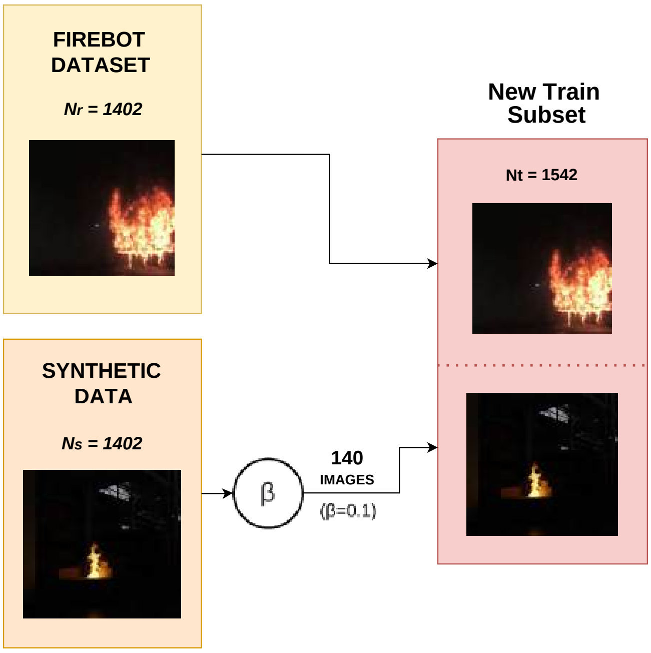


Figure 4.9: The flowchart depicts the process of data allocation for the training subset in the second ablation study when the $\beta = 0.1$. Image source: *Arlovic et al.* [5].

4.6 Results

To assess the performance of trained segmentation models across all ablation studies, we used the Dice Score, IoU Score, and Total Error as primary metrics. The models were trained using both publicly available datasets and the proposed SYN-FIRE dataset. The test subsets were evaluated using the thresholds that performed the best on their corresponding validation subsets. Figure 4.10 shows how synthetic data affects U-Net++ model training, using four randomly selected samples from the BowFire dataset to provide qualitative results.

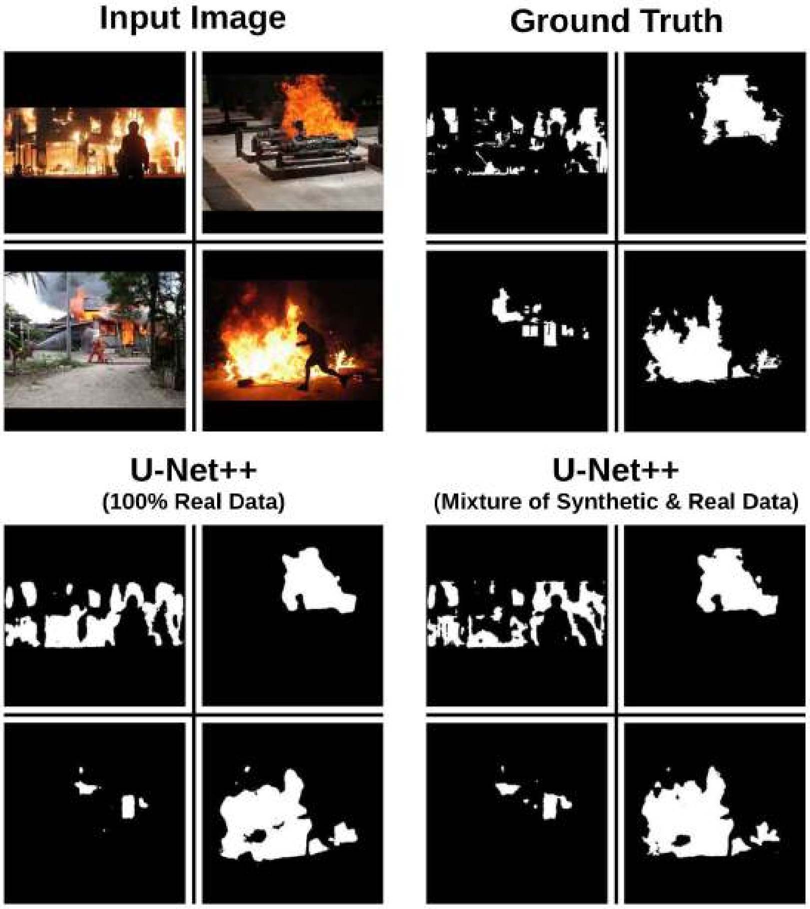


Figure 4.10: The U-Net++ model performance is demonstrated on four randomly selected samples from the BowFire dataset. Image source: *Arlovic et al.* [5].

The Tables 4.3 to 4.6 present experiment results that aimed to understand how substituting real with synthetic data during training impacts the performance of a semantic segmentation model. The *Real Data* parameter in the table indicates the percentage of real data α present in the training subset. The Tables 4.7 to 4.10 show the results of an experiment that investigated the influence of using synthetic data in model training alongside the maximum available real data. The parameter *Synthetic Data* in the table represents the percentage of synthetic data β added to the original dataset, consisting entirely of real data.

Table 4.3: Results of training the model on the validation and test subsets of the Corsican Fire Database dataset. The underlined results represent the baseline results from the model trained with $\alpha = 1$.

			Validation Dataset			**Test Dataset**		
Resolution	**Real Data**	**Thres-hold**	**Dice Score**	**IoU Score**	**Total Error**	**Dice Score**	**IoU Score**	**Total Error**
128 × 128	0%	10%	0.050	0.042	0.03156	0.096	0.081	0.05659
	10%	10%	0.179	0.230	0.02222	0.252	0.561	0.03931
	20%	10%	0.694	0.610	0.01135	0.606	0.693	0.03511
	30%	10%	0.811	0.819	0.00523	0.643	0.801	0.03125
	40%	20%	0.841	0.858	0.00422	0.728	0.800	0.03197
	50%	10%	0.838	0.868	0.00374	***0.736***	***0.818***	***0.03151***
	60%	10%	0.784	0.827	0.00449	0.675	0.787	0.03194
	70%	20%	0.820	0.813	0.00519	0.696	0.768	0.03254
	80%	20%	***0.858***	***0.862***	***0.00398***	0.731	0.821	0.03147
	90%	20%	0.772	0.796	0.00606	0.577	0.541	0.03797
	100%	30%	0.879	0.884	0.00412	0.810	0.827	0.03198
256 × 256	0%	10%	0.069	0.059	0.0298	0.144	0.114	0.05478
	10%	10%	0.725	0.698	0.00915	0.624	0.740	0.03392
	20%	10%	0.764	0.764	0.00721	0.719	0.750	0.03403
	30%	20%	0.834	0.854	0.00423	0.749	0.767	0.03306
	40%	20%	0.860	0.868	0.00365	0.755	0.825	0.03125
	50%	10%	0.854	0.901	0.00256	0.747	0.840	0.03062
	60%	20%	***0.891***	***0.906***	***0.0029***	***0.810***	***0.806***	***0.03186***
	70%	20%	0.851	0.851	0.00449	0.750	0.788	0.03246
	80%	20%	0.858	0.872	0.00397	0.752	0.756	0.03332
	90%	20%	0.875	0.879	0.00336	0.787	0.795	0.03195
	100%	30%	0.893	0.905	0.00318	0.835	0.840	0.03165
512 × 512	0%	10%	0.059	0.047	0.03084	0.155	0.114	0.05503
	10%	10%	0.679	0.606	0.01113	0.729	0.749	0.03436
	20%	20%	0.811	0.782	0.00676	0.779	0.795	0.03286
	30%	30%	0.921	0.923	0.00274	0.840	0.819	0.03206
	40%	20%	0.868	0.875	0.00363	0.787	0.777	0.03265
	50%	40%	0.930	0.932	0.00236	0.845	0.807	0.03218
	60%	40%	***0.933***	***0.927***	***0.00248***	0.853	0.819	0.03197
	70%	40%	0.927	0.920	0.00267	0.852	0.813	0.03208
	80%	30%	0.930	0.933	0.00209	***0.862***	***0.850***	***0.03096***
	90%	30%	0.931	0.917	0.00261	0.861	0.847	0.03125
	100%	40%	0.936	0.930	0.00239	0.866	0.837	0.03148

4.6.1 Corsican Fire Database

The model trained on the Corsican Fire Database exhibited the least effective performance on both the validation and test subsets when the dataset was partitioned into three segments with the following distribution: 55% for training, 15% for validation, and 30% for testing. After evaluating the impact of synthetic data on model training as a replacement for real data, it was observed that a model trained using synthetic data had a decrease in performance

Table 4.4: Results of training the model on the validation and test subsets of the FLAME dataset. The underlined results represent the baseline results from the model trained with $\alpha = 1$.

			Validation Dataset			**Test Dataset**		
Resolution	**Real Data**	**Thres-hold**	**Dice Score**	**IoU Score**	**Total Error**	**Dice Score**	**IoU Score**	**Total Error**
128 × 128	0%	100%	0.000	0.000	0.00305	0.000	0.000	0.00359
	10%	10%	0.215	0.204	0.00236	0.458	0.575	0.00164
	20%	10%	0.040	0.026	0.00293	0.257	0.200	0.00293
	30%	10%	0.495	0.501	0.00153	0.542	0.654	0.00131
	40%	20%	0.588	0.578	0.00129	0.585	0.631	0.00135
	50%	20%	***0.637***	***0.614***	***0.00120***	0.648	0.638	0.00132
	60%	10%	0.574	0.690	0.00095	0.571	0.711	0.00106
	70%	20%	0.622	0.631	0.00115	0.641	0.688	0.00111
	80%	20%	0.632	0.593	0.00128	0.660	0.694	0.00108
	90%	20%	0.604	0.564	0.00134	***0.678***	***0.693***	***0.00111***
	100%	30%	0.707	0.689	0.00096	0.703	0.702	0.00109
256 × 256	0%	10%	0.001	0.000	0.00295	0.000	0.000	0.00349
	10%	10%	0.537	0.467	0.00152	0.630	0.653	0.00125
	20%	10%	0.559	0.476	0.00157	0.672	0.672	0.00116
	30%	10%	0.605	0.575	0.00133	0.683	0.743	0.00088
	40%	20%	0.660	0.654	0.00109	0.729	0.743	0.00091
	50%	20%	0.681	0.671	0.00102	0.738	0.755	0.00085
	60%	20%	0.704	0.695	0.00095	0.744	0.729	0.00097
	70%	30%	0.725	0.702	0.00095	0.760	0.743	0.00086
	80%	20%	0.723	0.739	0.00085	0.746	0.772	0.00078
	90%	30%	***0.751***	***0.758***	***0.00074***	***0.771***	***0.788***	***0.00071***
	100%	30%	0.767	0.752	0.00073	0.781	0.764	0.00079
512 × 512	0%	10%	0.018	0.009	0.00301	0.012	0.006	0.00355
	10%	10%	0.693	0.648	0.00113	0.745	0.839	0.00056
	20%	20%	0.613	0.526	0.00139	0.572	0.480	0.00188
	30%	20%	0.748	0.694	0.00098	0.802	0.836	0.00059
	40%	20%	0.739	0.693	0.00102	0.808	0.854	0.00055
	50%	10%	0.749	0.774	0.00078	0.814	0.905	0.00034
	60%	30%	0.776	0.779	0.00074	0.812	0.892	0.00040
	70%	30%	0.783	0.753	0.00083	***0.838***	***0.855***	***0.00054***
	80%	20%	0.798	0.793	0.00072	0.835	0.873	0.00045
	90%	20%	***0.807***	***0.824***	***0.00060***	0.832	0.894	0.00037
	100%	30%	0.829	0.841	0.00050	0.831	0.890	0.00041

of 9.14% on a test subset with a resolution of 128 × 128 when 50% real data was employed. Utilizing 60% of the real data resulted in a performance loss of 2.99% at a resolution of 256 × 256. After increasing the resolution to 512 × 512, the model exhibited a marginal performance decline of 0.46%, utilizing 80% real data. The results from the ablation study aimed at evaluating the initial hypothesis, utilizing the Corsican FireDB dataset, are shown in Figure 4.11a and Table 4.3. Upon evaluating the impact of incorporating synthetic data

Table 4.5: Results of training the model on the validation and test subsets of the FireBot dataset. The underlined results represent the baseline results from the model trained with $\alpha = 1$.

			Validation Dataset			**Test Dataset**		
Resolution	**Real Data**	**Thres-hold**	**Dice Score**	**IoU Score**	**Total Error**	**Dice Score**	**IoU Score**	**Total Error**
128 × 128	0%	10%	0.443	0.586	0.0080	0.386	0.493	0.01188
	10%	10%	0.583	0.711	0.00778	0.507	0.603	0.01146
	20%	20%	0.597	0.708	0.00594	0.526	0.602	0.00908
	30%	20%	0.661	0.762	0.00548	0.577	0.658	0.0101
	40%	20%	0.691	0.788	0.00502	0.601	0.676	0.00935
	50%	30%	0.710	0.756	0.00546	0.611	0.641	0.00973
	60%	30%	0.634	0.666	0.00977	0.634	0.666	0.00977
	70%	30%	0.741	0.806	0.00377	0.642	0.700	0.00729
	80%	30%	0.740	0.797	0.00472	***0.658***	***0.700***	***0.00732***
	90%	***30%***	***0.753***	***0.795***	***0.00386***	0.657	0.693	0.00662
	100%	30%	0.764	0.823	0.00318	0.674	0.721	0.00535
256 × 256	0%	10%	0.498	0.568	0.00897	0.439	0.485	0.0128
	10%	20%	0.607	0.684	0.00835	0.516	0.579	0.01286
	20%	20%	0.708	0.804	0.00485	0.611	0.674	0.0094
	30%	20%	0.741	0.814	0.00427	0.642	0.699	0.00868
	40%	30%	0.775	0.809	0.00462	0.662	0.686	0.00862
	50%	30%	0.774	0.806	0.00390	0.660	0.680	0.00786
	60%	30%	0.789	0.822	0.00392	0.678	0.701	0.00804
	70%	30%	0.794	0.854	0.00296	0.690	0.732	0.00625
	80%	***30%***	***0.801***	***0.846***	***0.00313***	***0.709***	***0.736***	***0.00651***
	90%	30%	0.800	0.837	0.00278	0.703	0.729	0.00580
	100%	30%	0.808	0.831	0.00275	0.714	0.727	0.00554
512 × 512	0%	10%	0.586	0.655	0.00912	0.493	0.549	0.01256
	10%	20%	0.682	0.769	0.00783	0.598	0.649	0.01181
	20%	20%	0.742	0.838	0.00433	0.671	0.736	0.00433
	30%	20%	0.784	0.855	0.00342	0.671	0.736	0.00764
	40%	30%	0.823	0.837	0.00423	0.730	0.731	0.00742
	50%	30%	0.827	0.840	0.00365	0.736	0.746	0.00683
	60%	30%	0.840	0.860	0.00304	0.750	0.764	0.00643
	70%	30%	0.845	0.868	0.00288	0.763	0.777	0.00574
	80%	30%	0.831	0.871	0.0026	0.756	0.779	0.00547
	90%	***30%***	***0.847***	***0.890***	***0.00229***	***0.776***	***0.808***	***0.00447***
	100%	30%	0.853	0.884	0.00244	0.777	0.797	0.00484

with real data for model training, we observed that the Dice Score of the model trained with synthetic data remained constant. However, employing a resolution of 128 × 128 and an additional 60% of synthetic data did not improve the Dice Score metric. Conversely, the model's IoU Score on the test subset exhibited a decline of 0.7%.

In contrast, utilizing 100% synthetic data at a resolution of 256 × 256 resulted in a model performance increase of 0.59%. Furthermore, when the model was evaluated at a resolution

Table 4.6: Results of training the model on the validation and test subsets of the BowFire dataset. The underlined results represent the baseline results from the model trained with $\alpha = 1$.

			Validation Dataset			**Test Dataset**		
Resolution	**Real Data**	**Thres-hold**	**Dice Score**	**IoU Score**	**Total Error**	**Dice Score**	**IoU Score**	**Total Error**
128×128	0%	30%	0.354	0.427	0.0121	0.319	0.451	0.01325
	10%	30%	0.375	0.352	0.02433	0.364	0.362	0.02119
	20%	30%	0.343	0.399	0.01475	0.347	0.401	0.01711
	30%	20%	0.392	0.457	0.00881	0.386	0.527	0.00754
	40%	30%	0.312	0.389	0.0141	0.284	0.372	0.01471
	50%	***20%***	***0.415***	***0.486***	***0.00747***	***0.421***	***0.519***	***0.00731***
	60%	30%	0.307	0.300	0.02242	0.272	0.239	0.02355
	70%	20%	0.405	0.447	0.01077	0.382	0.472	0.0115
	80%	20%	0.357	0.362	0.01507	0.326	0.380	0.01582
	90%	40%	0.412	0.398	0.01967	0.379	0.355	0.02121
	100%	20%	0.413	0.480	0.00643	0.382	0.453	0.01085
256×256	0%	40%	0.395	0.450	0.00814	0.403	0.495	0.00897
	10%	20%	0.422	0.432	0.01379	0.418	0.457	0.01589
	20%	30%	0.443	0.468	0.00916	***0.422***	***0.439***	***0.01097***
	30%	20%	0.412	0.416	0.01414	0.379	0.391	0.01906
	40%	20%	0.417	0.438	0.01148	0.393	0.438	0.01244
	50%	20%	0.421	0.426	0.0103	0.382	0.407	0.01231
	60%	20%	0.440	0.499	0.0058	0.410	0.521	0.00772
	70%	30%	0.453	0.482	0.00858	0.412	0.498	0.00969
	80%	***30%***	***0.456***	***0.461***	***0.01081***	0.414	0.438	0.01207
	90%	20%	0.432	0.421	0.01079	0.406	0.449	0.0113
	100%	20%	0.462	0.498	0.00763	0.404	0.495	0.00979
512×512	0%	20%	0.421	0.509	0.00476	0.405	0.553	0.00538
	10%	20%	0.376	0.438	0.00907	0.363	0.492	0.01042
	20%	20%	0.442	0.456	0.00877	0.436	0.515	0.00936
	30%	20%	0.473	0.512	0.0054	0.445	0.430	0.01253
	40%	20%	0.452	0.520	0.00385	0.441	0.550	0.00703
	50%	20%	0.422	0.451	0.00918	0.412	0.502	0.00921
	60%	30%	0.474	0.501	0.00669	***0.491***	***0.546***	***0.00737***–
	70%	20%	0.447	0.480	0.00854	0.396	0.518	0.00913
	80%	10%	0.431	0.474	0.00733	0.422	0.513	0.01005
	90%	***30%***	***0.475***	***0.489***	***0.00587***	0.458	0.516	0.00662
	100%	30%	0.470	0.497	0.00676	0.461	0.522	0.00744

of 512 × 512 using 80% real data, it demonstrated a performance improvement of 0.80%. The outcomes of the ablation study, conducted to assess the second hypothesis utilizing the Corsican FireDB dataset, are presented in Figure 4.11b and Table 4.7.

Table 4.7: Results of training the model on the validation and test subsets of the Corsican Fire Database dataset. The underlined results represent the baseline results from the model trained with $\alpha = 1$.

			Validation Dataset			**Test Dataset**		
Resolution	**Synthetic Data**	**Thres-hold**	**Dice Score**	**IoU Score**	**Total Error**	**Dice Score**	**IoU Score**	**Total Error**
128 × 128	0%	30%	0.879	0.884	0.00412	0.810	0.827	0.03198
	10%	30%	0.871	0.878	0.00425	0.805	0.803	0.03238
	20%	30%	0.872	0.878	0.00395	0.804	0.820	0.03191
	30%	30%	***0.881***	***0.890***	***0.0038***	0.809	0.807	0.03232
	40%	40%	0.871	0.864	0.00468	0.808	0.787	0.03325
	50%	30%	0.880	0.889	0.00366	0.803	0.805	0.03207
	60%	30%	0.878	0.896	0.00350	***0.810***	***0.819***	***0.03195***
	70%	30%	0.871	0.880	0.00425	0.794	0.815	0.03233
	80%	30%	0.878	0.876	0.00439	0.801	0.808	0.03275
	90%	40%	0.865	0.836	0.00558	0.785	0.754	0.03470
	100%	30%	0.877	0.892	0.00373	0.809	0.823	0.03193
256 × 256	0%	30%	0.893	0.905	0.00318	0.835	0.840	0.03165
	10%	40%	0.900	0.890	0.00353	0.836	0.824	0.03188
	20%	30%	0.899	0.903	0.00314	0.837	0.836	0.03150
	30%	40%	0.898	0.880	0.00414	0.831	0.798	0.03274
	40%	40%	0.899	0.878	0.00410	0.815	0.810	0.03234
	50%	40%	0.903	0.894	0.00370	0.828	0.798	0.03277
	60%	30%	0.907	0.915	0.00280	0.838	0.843	0.03128
	70%	40%	0.908	0.892	0.00366	0.824	0.781	0.03320
	80%	40%	0.900	0.889	0.00390	0.828	0.800	0.03276
	90%	40%	0.822	0.784	0.00356	0.822	0.784	0.03307
	100%	40%	***0.908***	***0.898***	***0.00338***	***0.840***	***0.829***	***0.03184***
512 × 512	0%	40%	0.936	0.930	0.00239	0.866	0.837	0.03148
	10%	30%	0.933	0.939	0.00213	0.865	0.852	0.03105
	20%	40%	0.927	0.923	0.00268	0.861	0.830	0.03174
	30%	30%	0.932	0.925	0.00260	0.862	0.837	0.03143
	40%	30%	0.932	0.932	0.00220	0.863	0.851	0.03104
	50%	30%	0.933	0.942	0.00190	0.862	0.850	0.03089
	60%	40%	0.935	0.927	0.00255	0.859	0.822	0.03187
	70%	30%	0.932	0.937	0.00213	0.864	0.849	0.03111
	80%	30%	***0.937***	***0.946***	***0.00179***	***0.873***	***0.869***	***0.03052***
	90%	30%	0.934	0.938	0.00212	0.868	0.860	0.03087
	100%	30%	0.935	0.944	0.00212	0.868	0.848	0.03120

4.6.2 FLAME Dataset

When trained with the FLAME dataset, the model's performance decreased by 3.56% and 1.28% at resolutions of 128 × 128 and 256 × 256, respectively. However, its performance improved by 0.84% at a resolution of 512 × 512. To obtain comparable or superior performance in the 128 × 128, 256 × 256, and 512 × 512 training, the model required a high

Table 4.8: Results of training the model on the validation and test subsets of the FLAME dataset. The underlined results represent the baseline results from the model trained with $\alpha = 1$.

			Validation Dataset			**Test Dataset**		
Resolution	**Synthetic Data**	**Thres-hold**	**Dice Score**	**IoU Score**	**Total Error**	**Dice Score**	**IoU Score**	**Total Error**
128 × 128	0%	30%	0.707	0.689	0.00096	0.703	0.702	0.00109
	10%	20%	0.668	0.670	0.00100	0.684	0.678	0.00115
	20%	20%	0.702	0.734	0.00081	0.697	0.737	0.00093
	30%	30%	***0.722***	***0.750***	***0.00077***	0.725	0.711	0.00102
	40%	20%	0.720	0.757	0.00073	0.708	0.726	0.00096
	50%	20%	0.673	0.670	0.00103	0.684	0.758	0.00087
	60%	20%	0.718	0.750	0.00076	0.715	0.777	0.00077
	70%	20%	0.710	0.740	0.00079	0.714	0.727	0.00096
	80%	20%	0.690	0.708	0.00091	0.706	0.739	0.00093
	90%	20%	0.719	0.758	0.00075	***0.728***	***0.753***	***0.00084***
	100%	20%	0.675	0.706	0.00092	0.652	0.715	0.00106
256 × 256	0%	30%	0.767	0.752	0.00073	0.781	0.764	0.00079
	10%	20%	0.743	0.751	0.00077	0.758	0.780	0.00078
	20%	30%	0.754	0.750	0.00074	0.773	0.787	0.00072
	30%	30%	0.748	0.726	0.00083	0.771	0.789	0.00073
	40%	30%	0.734	0.730	0.00081	0.767	0.792	0.00070
	50%	30%	***0.763***	***0.749***	***0.00078***	***0.789***	***0.814***	***0.00064***
	60%	30%	0.756	0.763	0.00072	0.774	0.803	0.00067
	70%	20%	0.743	0.736	0.00079	0.748	0.734	0.00088
	80%	30%	0.741	0.699	0.00091	0.772	0.801	0.00073
	90%	30%	0.745	0.731	0.00083	0.750	0.709	0.00104
	100%	20%	0.748	0.746	0.00077	0.765	0.722	0.00096
512 × 512	0%	30%	0.829	0.841	0.00050	0.831	0.890	0.00041
	10%	20%	0.827	0.825	0.00057	0.838	0.877	0.00045
	20%	30%	0.834	0.830	0.00054	0.841	0.888	0.00041
	30%	30%	0.827	0.796	0.00065	0.851	0.866	0.00049
	40%	30%	0.829	0.796	0.00065	0.842	0.881	0.00044
	50%	30%	0.813	0.801	0.00065	0.830	0.839	0.00058
	60%	30%	0.837	0.818	0.00057	***0.851***	***0.877***	***0.00044***
	70%	30%	0.827	0.24	0.00057	0.818	0.910	0.00032
	80%	30%	0.836	0.813	0.00059	0.837	0.881	0.00045
	90%	20%	0.822	0.832	0.00055	0.845	0.901	0.00037
	100%	30%	***0.837***	***0.843***	***0.00050***	0.838	0.896	0.00039

proportion of real training data, incorporating 90%, 80%, and 90% of real data, respectively. The results of the first ablation study on the FLAME dataset are presented in 4.12a and Table 4.4. Integrating synthetic data during the training phase and real data in the second ablation session led to modest but consistent improvements in the Dice Score across all tested resolutions. Specifically, when 90% synthetic data was integrated, the model exhibited a 3.55% improvement at a resolution of 128 × 128. Incorporating 50%

Table 4.9: Results of training the model on the validation and test subsets of the FireBot dataset. The underlined results represent the baseline results from the model trained with $\alpha = 1$.

			Validation Dataset			**Test Dataset**		
Resolution	**Synthetic Data**	**Thres-hold**	**Dice Score**	**IoU Score**	**Total Error**	**Dice Score**	**IoU Score**	**Total Error**
128 × 128	0%	30%	0.764	0.823	0.00318	0.674	0.721	0.00535
	10%	20%	0.702	0.781	0.00439	0.674	0.765	0.00441
	20%	30%	0.698	0.722	0.0062	0.665	0.701	0.00594
	30%	30%	0.701	0.737	0.00587	0.674	0.721	0.00569
	40%	30%	0.709	0.742	0.00567	0.676	0.725	0.0052
	50%	20%	0.707	0.792	0.00461	0.682	0.783	0.00438
	60%	20%	0.706	0.782	0.00500	0.670	0.755	0.00505
	70%	20%	0.716	0.797	0.00489	0.685	0.778	0.00483
	80%	30%	0.717	0.749	0.00578	0.678	0.722	0.00552
	90%	20%	0.693	0.743	0.00576	0.659	0.720	0.00613
	100%	***30%***	***0.725***	***0.761***	***0.00542***	***0.696***	***0.742***	***0.00530***
256 × 256	0%	30%	0.808	0.831	0.00275	0.714	0.727	0.00554
	10%	30%	0.752	0.774	0.00512	0.726	0.754	0.00521
	20%	20%	0.762	0.829	0.00367	0.735	0.815	0.00383
	30%	30%	0.771	0.792	0.00476	0.746	0.775	0.00474
	40%	30%	0.755	0.773	0.00522	0.730	0.757	0.00525
	50%	20%	0.766	0.832	0.00356	0.742	0.822	0.00331
	60%	30%	0.765	0.793	0.00464	0.741	0.781	0.00442
	70%	20%	0.766	0.825	0.00382	0.744	0.813	0.00402
	80%	30%	0.756	0.783	0.00483	0.735	0.773	0.0047
	90%	***20%***	***0.776***	***0.828***	***0.00385***	***0.752***	***0.815***	***0.00402***
	100%	20%	0.761	0.823	0.00393	0.738	0.811	0.00398
512 × 512	0%	30%	0.853	0.884	0.00244	0.777	0.797	0.00484
	10%	20%	0.788	0.824	0.00415	0.767	0.812	0.00446
	20%	30%	0.804	0.820	0.00436	0.787	0.809	0.00431
	30%	20%	0.778	0.814	0.00407	0.763	0.806	0.00423
	40%	30%	0.798	0.814	0.00419	0.781	0.804	0.00412
	50%	20%	0.804	0.851	0.00363	0.777	0.836	0.00363
	60%	20%	0.807	0.847	0.00356	0.790	0.840	0.00354
	70%	30%	***0.812***	***0.829***	***0.00418***	***0.793***	***0.820***	***0.00417***
	80%	30%	0.809	0.823	0.00423	0.788	0.808	0.00416
	90%	20%	0.793	0.826	0.00431	0.781	0.823	0.00421
	100%	30%	0.810	0.839	0.00391	0.789	0.829	0.00386

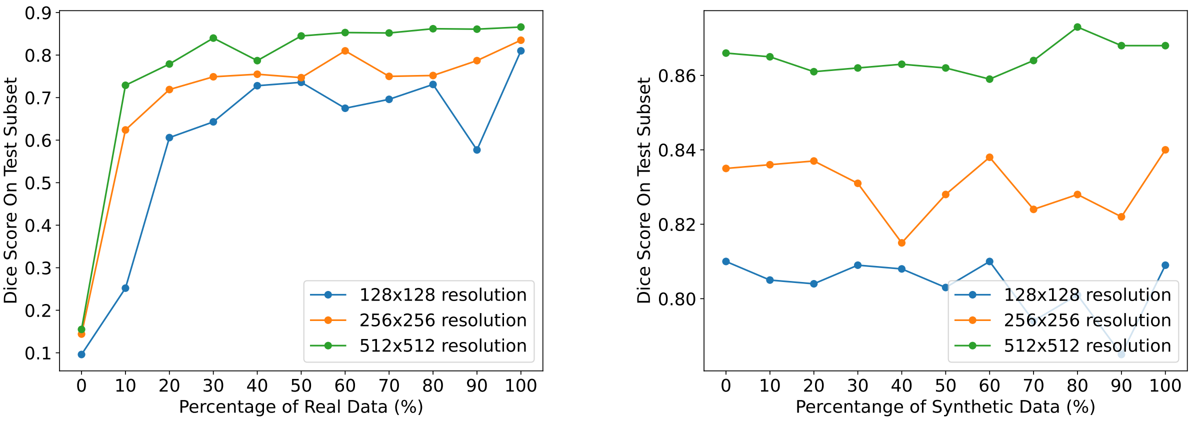
synthetic data at a resolution of 256 × 256 resulted in an improvement of 1.02%. Ultimately, at a resolution of 512 × 512, the model exhibited a moderate 2.41% improvement upon integrating 60% synthetic data. The results of the second ablation study on the FLAME dataset are presented in Figure 4.12b and Table 4.8. Overall, the results demonstrate that introducing synthetic data during training can enhance model performance on the FLAME dataset, particularly at higher resolutions.

Table 4.10: Results of training the model on the validation and test subsets of the BowFire dataset. The underlined results represent the baseline results from the model trained with $\alpha = 1$.

			Validation Dataset			**Test Dataset**		
Resolution	**Synthetic Data**	**Thres-hold**	**Dice Score**	**IoU Score**	**Total Error**	**Dice Score**	**IoU Score**	**Total Error**
128 × 128	0%	20%	0.413	0.480	0.00643	0.382	0.453	0.01085
	10%	20%	0.456	0.510	0.00468	0.419	0.561	0.00619
	20%	30%	0.397	0.435	0.00952	0.380	0.464	0.01228
	30%	20%	0.329	0.353	0.02187	0.278	0.320	0.02143
	40%	***20%***	***0.459***	***0.477***	***0.0056***	***0.435***	***0.492***	***0.00636***
	50%	20%	0.432	0.490	0.00503	0.389	0.460	0.01097
	60%	10%	0.394	0.447	0.01013	0.343	0.412	0.01367
	70%	30%	0.343	0.358	0.01756	0.281	0.285	0.02322
	80%	10%	0.390	0.450	0.01119	0.410	0.530	0.00793
	90%	30%	0.336	0.342	0.01802	0.280	0.272	0.02448
	100%	30%	0.453	0.482	0.00927	0.414	0.488	0.01068
256 × 256	0%	20%	0.462	0.498	0.00763	0.404	0.495	0.00979
	10%	40%	0.443	0.425	0.01279	0.419	0.426	0.01334
	20%	40%	0.363	0.329	0.01822	0.363	0.359	0.01779
	30%	20%	0.448	0.477	0.00908	0.405	0.502	0.01019
	40%	30%	0.479	0.489	0.00831	***0.469***	***0.515***	***0.00894***
	50%	20%	0.457	0.500	0.00653	0.442	0.452	0.01051
	60%	30%	0.483	0.493	0.00734	0.450	0.500	0.0077
	70%	10%	0.458	0.529	0.00453	0.423	0.558	0.00591
	80%	20%	0.349	0.362	0.01691	0.331	0.378	0.01681
	90%	20%	0.415	0.447	0.01035	0.374	0.429	0.01191
	100%	***20%***	***0.484***	***0.513***	***0.00592***	0.434	0.507	0.01077
512 × 512	0%	30%	0.470	0.497	0.00676	0.461	0.522	0.00744
	10%	20%	0.463	0.471	0.00764	0.435	0.400	0.01192
	20%	20%	0.468	0.503	0.00528	0.442	0.527	0.00559
	30%	40%	0.475	0.489	0.00697	0.435	0.491	0.00974
	40%	30%	0.515	0.535	0.00393	0.492	0.499	0.01102
	50%	30%	0.515	0.529	0.00406	0.486	0.486	0.0122
	60%	50%	0.519	0.516	0.00521	0.474	0.450	0.01449
	70%	20%	0.430	0.498	0.00535	0.410	0.557	0.00694
	80%	20%	0.471	0.490	0.00625	0.459	0.498	0.00878
	90%	***30%***	***0.520***	***0.538***	***0.00372***	0.462	0.483	0.01172
	100%	40%	0.505	0.520	0.00537	***0.499***	***0.509***	***0.0101***

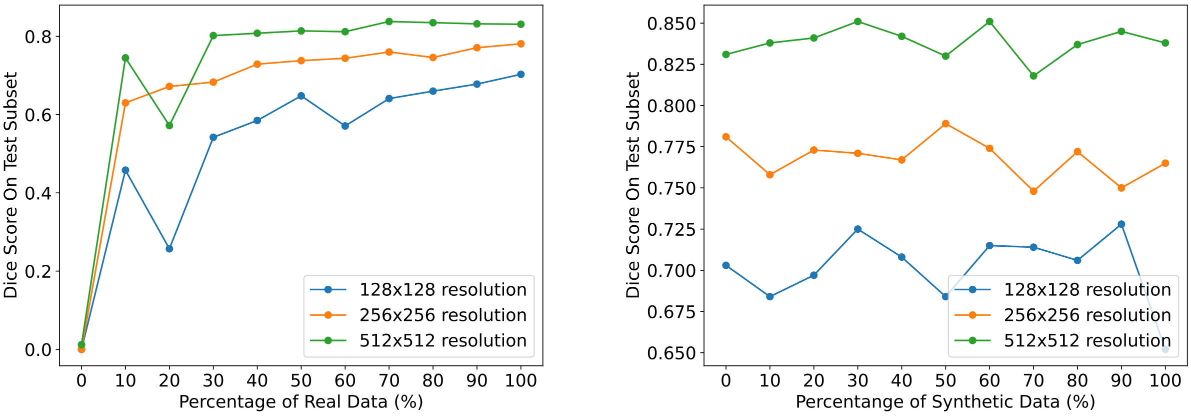
4.6.3 FireBot Dataset

The models trained using both the FireBot dataset and synthetic data demonstrated performance declines of 2.37%, 0.7%, and 0.12% at input resolutions of 128 × 128, 256 × 256, and 512 × 512, respectively. The results were derived utilizing 80% of real data at 128 × 128 and 256 × 256 resolutions and 90% at 512 × 512. The outcomes of this initial



(a) (b)

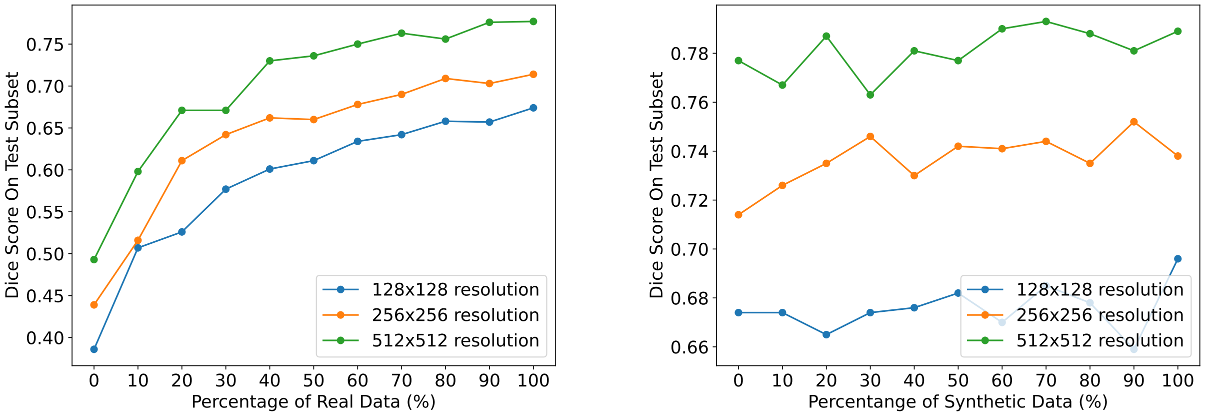
Figure 4.11: Model evaluation on the test subset using the SYN-FIRE and Corsican FireDB datasets. (a) Dice Score when real data is substituted with synthetic data, and (b) Dice Score when synthetic data is incorporated with real data. Image source: *Arlovic et al.* [5].



(a) (b)

Figure 4.12: Evaluation of the model on the test subset utilizing the SYN-FIRE and Flame datasets. (a) Dice Score when actual data is substituted with synthetic data, and (b) Dice Score when synthetic data is integrated with real data. Image source: *Arlovic et al.* [5].

ablation study are presented in 4.13a and Table 4.5. Furthermore, the second study, which integrated synthetic and real data, significantly improved performance across all resolutions. Integrating 128×128 with 100% synthetic data with 100% real data yielded a 3.26% enhancement. At 256×256, the incorporation of 90% synthetic data resulted in a 5.32% improvement, whereas at 512×512, a 2.06% improvement was attained by integrating 70% synthetic data. The results of the second ablation study are shown in in 4.13b and Table 4.9.



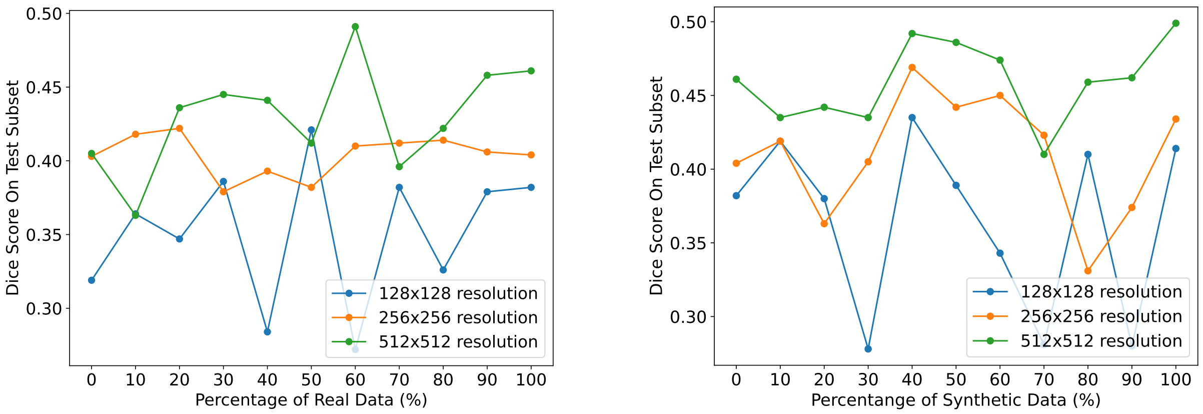
(a)

(b)

Figure 4.13: Evaluation of the model on the test subset utilizing the SYN-FIRE and FireBot datasets. (a) Dice Score when actual data is substituted with synthetic data, and (b) Dice Score with the integration of synthetic data with real data. Image source: *Arlovic et al.* [5].

4.6.4 BowFire Dataset

The model, trained using the BowFire dataset and synthetic data, exhibited substantial improvements in Dice Score at all resolutions. It achieved improvements of 10.21% at 128 × 128, 4.46% at 256 × 256, and 6.51% at 512 × 512 by integrating synthetic data with 50%, 80%, and 60% of real data, respectively. The results are shown in in 4.14a and Table 4.6. In the second ablation study, the use of synthetic data in the training process resulted in significant performance enhancements across all evaluated resolutions. At a resolution of 128 × 128, the Dice Score improved by 13.87% using 40% synthetic data. At 256 × 256, a 16.09% improvement was observed, while at 512 × 512, performance increased by 8.24%. The results of this second study are shown in 4.14b and detailed in Table 4.10.



(a)

(b)

Figure 4.14: Evaluation of models on the test subset, trained using the SYN-FIRE and BowFire datasets. (a) Displays the Dice Score when real data is replaced with synthetic data, and (b) Displays the Dice Score when integrating synthetic data alongside real data. Image source: *Arlovic et al.* [5].

4.7 Statistical Analysis of Results

Statistical analysis was conducted to determine whether the observed improvements in segmentation accuracy were statistically significant or attributable to random variation. A paired T-test was employed to evaluate the impact of synthetic data in two ablation studies across all datasets at resolutions of 128 × 128, 256 × 256, and 512 × 512 pixels. The paired T-test was selected as the appropriate statistical method because it compares two related samples, specifically the performance metrics of the same model architecture trained with and without synthetic data [152]. This approach controls for variability by treating each paired observation as a matched comparison, thereby isolating the effect of synthetic data on segmentation performance. All reported p-values were below the significance threshold of $\alpha = 0.05$, confirming that the observed improvements in Dice Scores were statistically significant [153].

The first ablation study demonstrated that the model performance can be maintained at a similar level by substituting a portion of the real images with synthetic ones. Statistically significant improvements in segmentation accuracy were observed by training on the BowFire dataset with the addition of SYN-FIRE data ($p = 0.031$, $p = 0.013$, $p = 0.028$). The Dice Score increased by 10.21%, 4.46%, and 6.51% at resolutions of 128 × 128, 256 × 256, and 512 × 512, respectively. The second ablation study revealed that additional synthetic data had improved the model performance ($p = 0.042$, $p = 0.030$, $p = 0.033$). Adding synthetic data to the FireBot dataset improved Dice Scores by 3.26%, 5.32%, and 2.06% at resolutions of 128×128, 256×256, and 512×512, respectively. The second ablation study on the BowFire dataset demonstrated the impact of synthetic data on segmentation performance, achieving increases of 13.87%, 16.09%, and 8.24% in Dice Score across all resolutions.

4.7.1 Statistical Analysis of Ablation Study 1 Results

Models trained on real data from the Corsican Fire Database achieved statistically higher Dice Scores across all tested resolutions than models that used real and synthetic data. The Dice Score for real data at a resolution of 128 × 128 pixels was 0.809 ± 0.18, significantly higher ($t(339) = 12.991$, $p = 0.006$) than the 0.735 ± 0.199 achieved using 50% synthetic data. At a resolution of 256 × 256, training with real data resulted in a Dice Score of 0.835 ± 0.175, whereas training with 40% synthetic data achieved a score of 0.810 ± 0.196, $t(339) = 5.241$, $p = 0.005$. At a resolution of 512 × 512, real data produced a Dice Score of 0.866 ± 0.187, barely surpassing ($t(339) = 2.078$, $p = 0.002$) the 0.862 ± 0.187 obtained with 20% synthetic data. Models trained entirely on real data in the FLAME dataset consistently achieved higher Dice Scores across all resolutions than models trained with a mix of synthetic data. At 128 × 128 resolution, training with real data yielded a Dice Score of 0.703 ± 0.056, which was greater ($t(300) = 8.539$, $p = 0.003$) than the 0.678 ± 0.056

achieved with 10% synthetic data. Similarly, at 256 × 256 resolution, the Dice Score was 0.781 ± 0.039 with real data, beating the 0.770 ± 0.041 achieved when 10% of the data was synthetic ($t(300) = 6.336$, $p = 0.002$). At 512×512 resolution, combining 30% synthetic data slightly improved performance ($t(300) = 4.048$, $p = 0.002$) to a Dice Score of 0.838 ± 0.030, compared to 0.831 ± 0.037 for real data. In the FireBot dataset, models trained on real data consistently achieved higher or similar Dice Scores across all resolutions than models that used synthetic data to some extent. At a 128 × 128 resolution, models trained solely with real data produced a Dice Score of 0.674 ± 0.249, compared to a lower score of 0.658 ± 0.239 ($t(1985) = 6.135$, $p = 0.003$) when 20% of the data was synthetic. Similarly, at 256 × 256 resolution, the real-data-only model achieved a Dice Score of 0.714 ± 0.235, slightly exceeding the 0.709 ± 0.228 ($t(1985) = 2.124$, $p = 0.002$) score obtained with 20% synthetic data. At 512 × 512 resolution, model performance was consistent with a Dice Score of 0.776 ± 0.219 for real data and 0.776 ± 0.212 ($t(1985) = 0.275$, $p = 0.002$) when 10% synthetic data was utilized. When real data was substituted with synthetic data, the BowFire dataset's model performance was lower or equivalent at all resolutions. At a resolution of 128 × 128, models trained entirely on real data achieved a Dice Score of 0.382 ± 0.363, which is lower ($t(23) = 1.419$, $p = 0.013$) than the 0.421 ± 0.360 achieved when 50% of the data was synthetic, $t(23) = 1.252$, $p = 0.031$. Training using real data at a resolution of 256 × 256 resulted in a Dice Score of 0.404 ± 0.376, lower than the 0.422 ± 0.372 achieved with 40% synthetic data. At a resolution of 512 × 512, model performance decreased ($t(23) = 1.056$, $p = 0.028$), resulting in a Dice Score of 0.461 ± 0.381 for real data and 0.491 ± 0.386 when 40% of real data was substituted with synthetic data. Figure 4.15 shows the paired t-test of Dice Score changes in Ablation Study 1.

4.7.2 Results For Ablation Study 2

When synthetic data was combined with real data, the Corsican Fire Database dataset exhibited equivalent or slightly improved model performance ($t(339) = 0.0079$, $p = 0.002$) at all resolutions. At a resolution of 128 × 128, the Dice Score for real data was 0.810 ± 0.18, which coincided with the model's performance trained using 60% synthetic data containing 0.810 ± 0.182. At a resolution of 256 × 256, the model achieved a slightly higher ($t(339) = 1.741$, $p = 0.003$) Dice Score of 0.840 ± 0.181 with the inclusion of 100% synthetic data alongside real data in contrast to 0.835 ± 0.175 with only real data. At a resolution of 512×512, incorporating 80% synthetic data resulted in a slight improvement ($t(339) = 4.336$, $p = 0.001$) in performance, providing a Dice Score of 0.872 ± 0.187 compared to 0.866 ± 0.187 with only real data. The model's performance improved modestly across resolutions while combining synthetic and real data from the FLAME dataset. At a resolution of 128 × 128, models exclusively trained on real data achieved a Dice Score of 0.703 ± 0.056, which was equivalent ($t(300) = 0.806$, $p = 0.003$) to the score of 0.706 ± 0.054 when 90% synthetic data

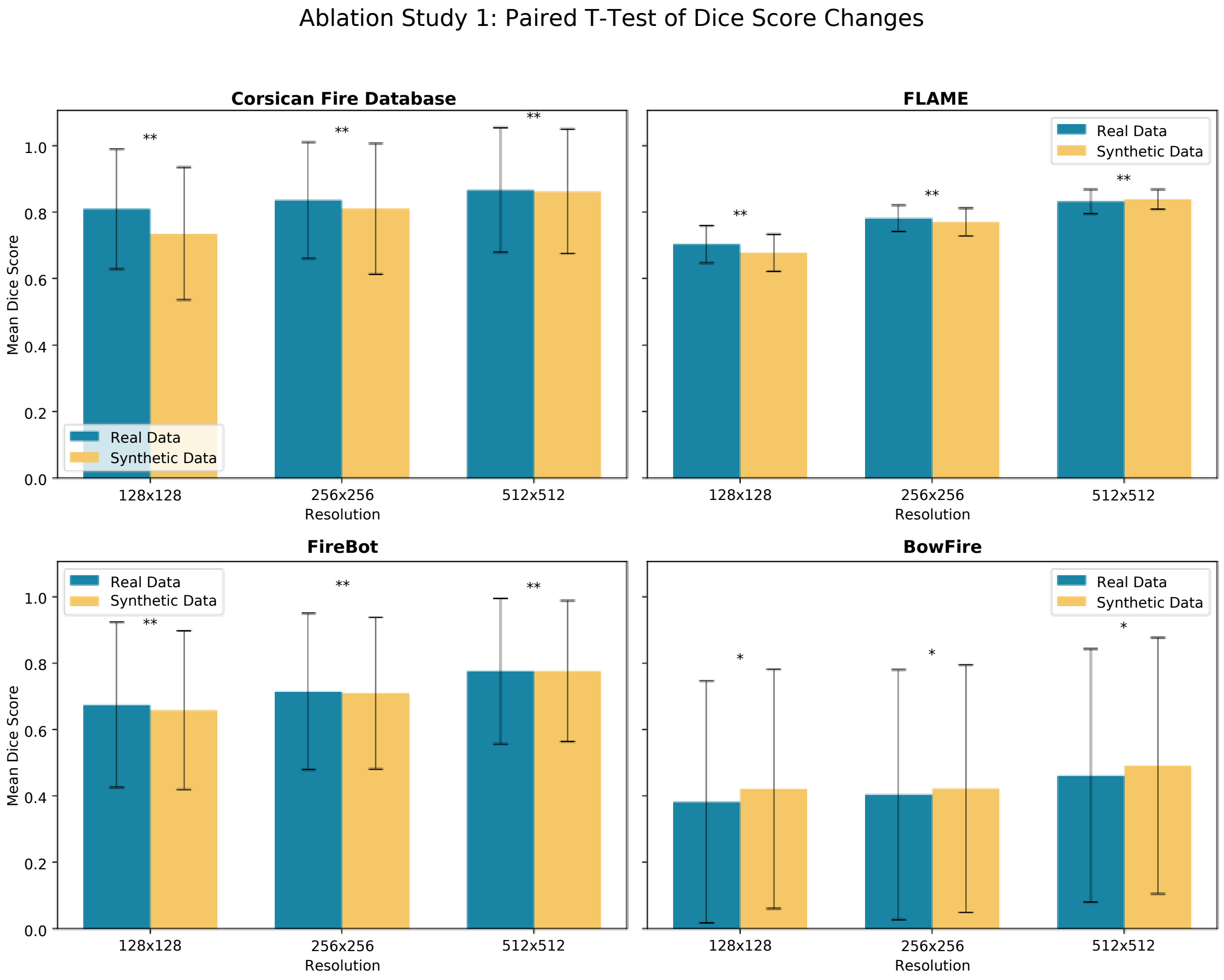


Figure 4.15: The paired T-Test statistical analysis of Dice Score changes in Ablation Study 1.

was included. At a resolution of 256 × 256, the model utilizing 50% synthetic data obtained a slightly superior ($t(300) = 4.8651$, $p = 0.002$) Dice Score of 0.789 ± 0.038, in contrast to 0.781 ± 0.039 with just real data. At a resolution of 512 × 512, the Dice Score of 0.851 ± 0.029 was obtained by adding 60% synthetic data. This was greater ($t(300) = 13.855$, $p = 0.001$) than the 0.831 ± 0.037 achieved with real data. When synthetic data was incorporated with real data in the FireBot dataset, the model's performance remained equivalent or slightly improved across resolutions. At a resolution of 128 × 128, models trained solely on real data achieved a Dice Score of 0.674 ± 0.249, whereas adding 100% synthetic data resulted in a higher score of 0.696 ± 0.222 ($t(1985) = 9.434$, $p = 0.002$). At a resolution of 256 × 256, training utilizing 90% of synthetic data resulted in a Dice Score of 0.752 ± 0.196, barely beating ($t(1985) = 15.006$, $p = 0.003$) the 0.714 ± 0.235 achieved utilizing just real data. At a resolution of 512×512, model performance exhibited a small improvement ($t(1985) = 7.131$, $p = 0.002$), achieving a Dice Score of 0.776 ± 0.219 for real data and 0.793 ± 0.193 when integrating 70% synthetic data. When synthetic data was used to train the BowFire dataset, the model's performance improved across all resolutions. At a resolution of 128×128, models trained with an additional 90% synthetic data achieved an improved ($t(23) = 1.261$, $p = 0.042$) Dice Score (0.435 ± 0.395) relative to those trained exclusively on real data (0.382 ± 0.363). At a resolution of 256×256, combining 40% synthetic data with the real data provided an improved ($t(23) = 2.165$, $p = 0.030$) Dice Score of 0.469 ± 0.383, in contrast to 0.404 ± 0.376 obtained with only real data. At a resolution of 512×512, integrating 90% synthetic data increased the Dice Score to 0.507 ± 0.403, surpassing ($t(23) = 1.393$, $p = 0.033$) the 0.461 ± 0.381 obtained with only real data. Figure 4.16 shows the paired t-test of Dice Score changes in Ablation Study 2.

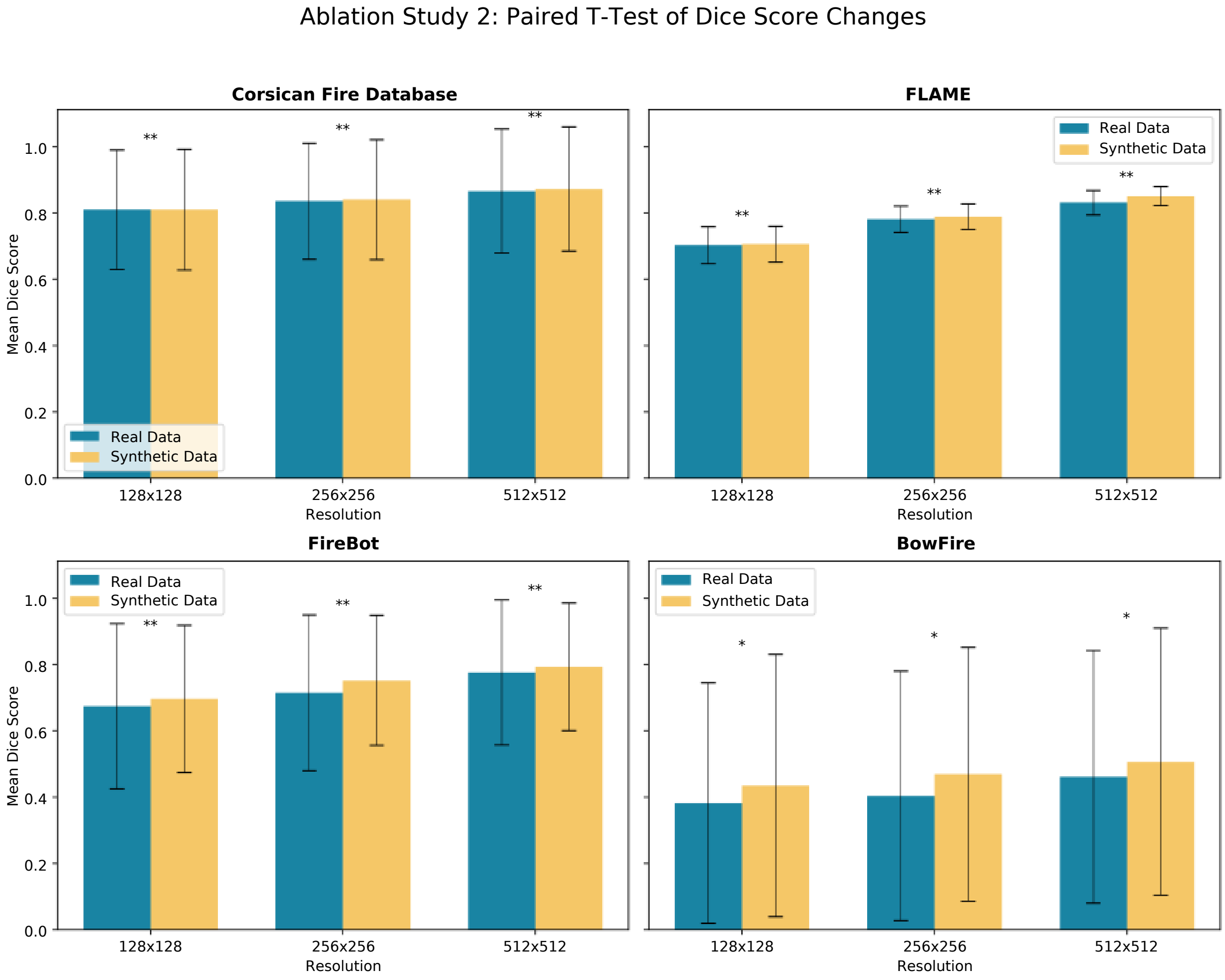


Figure 4.16: The paired T-Test statistical analysis of Dice Score changes in Ablation Study 2.

4.8 Model Generalization on Unseen Real-Life Data

When a model is trained on synthetic data, it may fail to capture the full complexity, variability, and noise of real-world scenarios. Because synthetic datasets are generated under assumptions and rules, they often contain biases or artifacts that models may exploit instead of learning the true underlying structure of the problem. This raises the risk of overfitting to artificial patterns, making it essential to test model generalization to ensure that learned representations extend beyond the synthetic environment and remain valid in more realistic or diverse conditions. In addition to validating our methodology using publicly available fire datasets, we further assessed its practical applicability by evaluating the ablation-selected top models on a separate dataset of 40 real indoor fire images. These images are part of a newly developed dataset of real fire scenarios. The dataset was created through a series of controlled fire experiments conducted at the State Firefighting School (DVŠ) in Zagreb. Preparation for the experiment, the journey to Zagreb, and capturing real fire images in two different indoor scenes took two full working days. In contrast, creating two synthetic environments using NVIDIA Omniverse required only one working day and did not involve travel, preparation, or risk of incident. All tests were conducted under the guidance of experienced firefighters, ensuring that safety procedures were followed. We ignited various materials commonly found in industrial and residential spaces to create realistic indoor fire scenarios. During these controlled fire scenarios, we recorded high-resolution RGB + thermal (visible + infrared) imagery, creating a robust dataset that captures the complexity of indoor fires. This dataset is designed to be a reference point for real-world indoor fire detection. Figure 4.17 shows that replacing real data with SYN-FIRE in Ablation Study 1 maintains or improves the Dice Score across multiple datasets. Incorporating SYN-FIRE data alongside real data in Ablation Study 2 yields further gains, indicating it is an effective complement, with only a modest dip on BowFire.

While the SYN-FIRE results show consistent improvements, they do not reveal where the model is focusing. To check that the improvements reflect plausible fire evidence rather than synthetic artifacts, we examine model attention using Gradient-weighted Class Activation Mapping (Grad-CAM). Grad-CAM is a post-hoc, class-discriminative visualization method that computes the gradient of a target class score (logit) with respect to a chosen convolutional layer feature maps, spatially averages those gradients to get per-channel importance weights, forms a weighted sum of the feature maps, applies ReLU, and upsamples the result to produce a coarse localization heatmap [154]. The resulting saliency maps in Figure 4.18 highlight image regions that most influenced each prediction. In our case, the highlighted areas align with flames and smoke, indicating that the model distinguishes flame from non-flame pixels at a local scale. The activations are spatially coherent rather than scattered, which is consistent with accurate pixel-wise segmentation. These observations suggest that the model is learning structured and relevant features for distinguishing between fire and non-fire areas, a crucial property for safety-critical applications such as fire detection.

Our results show that synthetic datasets can help models generalize better, but they are not a substitute for real-world data. It is also important to note that the dataset

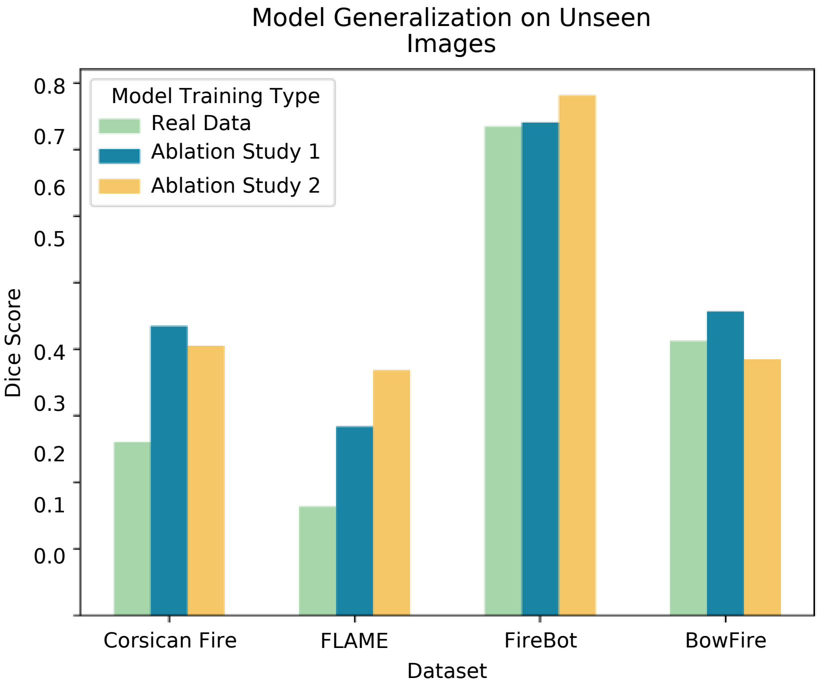


Figure 4.17: Evaluation of the model's performance using 40 real-world scenario images left out from the training dataset.

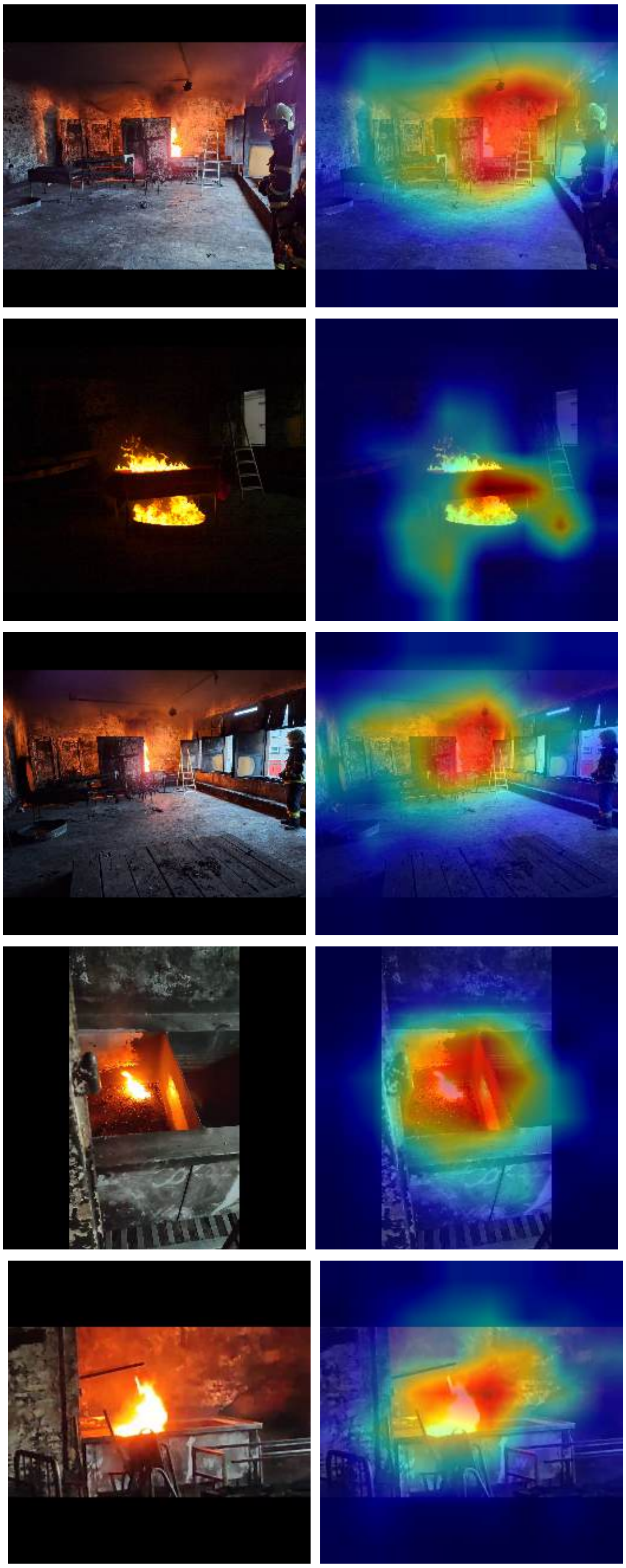
we assembled is particularly challenging and notably different from the datasets used for model training. Lastly, since we based our experiments on a single architecture, U-Net++, the effects of synthetic data could differ from those of other models. Adding more data typically leads to better outcomes, but synthetic data may negatively impact specific model architectures.

4.9 Conclusion

The effectiveness of deep neural networks in fire detection has been demonstrated to surpass traditional techniques. Nevertheless, they require high-quality datasets with large amounts of annotated data, which are time-consuming and highly expensive to gather. To address this challenge, this research study introduced the SYN-FIRE dataset, an entirely synthetic, publicly available dataset of indoor fires in industrial environments developed using NVIDIA Omniverse. The dataset was designed to address key challenges associated with training semantic segmentation models for fire detection in scenarios where real-world data is limited, costly, or difficult to obtain.

The impact of SYN-FIRE was evaluated through benchmark experiments on multiple publicly available datasets. It was observed that models trained exclusively on synthetic data underperformed compared to those trained on real data. However, when synthetic data was incorporated alongside real images, a consistent improvement in segmentation accuracy was achieved. This effect was particularly pronounced in the case of small datasets, such as BowFire, where performance improvements ranged from 4% to 10% depending on the resolution. Synthetic data was also an effective partial substitute for real data, reducing the need for extensive manual annotation while maintaining model performance. Additional advantages of synthetic data were identified regarding practical feasibility and development time. For instance, it took approximately three days to collect and annotate 40 real images, whereas a comparable quantity of synthetic images was generated in the span of 30 minutes. Procedural generation further facilitated the rapid production of large image sets with controlled variation in fire scenarios, object placement, and environmental conditions.

Considering these benefits, several drawbacks of the SYN-FIRE dataset were recognized. The current version inadequately represents varied environmental factors, realistic occlusions, human presence, or the full complexity of smoke dynamics. These missing variables may limit the generalizability of models trained solely on synthetic data. To address these limitations, future initiatives will aim to improve the dataset by including a wider variety of textures, lighting conditions, occlusion patterns, and fire-related behaviors. Moreover, more research will be conducted to examine the impact of synthetic data on various state-of-the-art deep learning architectures, providing extensive insights into its applicability for pixel-wise fire segmentation tasks.



RGB Image Grad-CAM

Figure 4.18: Comparison of original images and Grad-CAM overlays showing model attention on fire regions.

5

Conclusion

Fire is a complex phenomenon encompassing flame, heat, smoke, and combustion gases. The visible flame and smoke arise from an exothermic reaction between a fuel and an oxidizer, most often oxygen. These signals span multiple spatial and temporal scales, which makes fire a strong target for dependable visual detection and segmentation. Recent work has focused on large neural networks trained on large-scale datasets. These models can generalize across domains, yet deployment in safety-critical industrial interiors must conform to limited computation capabilities, low latency, and scarcity of labeled data. Traditional sensor-based systems, such as smoke detectors or gas sensors often react slowly, are sensitive to their placement, and suffer from airflow issues, frequently generating false alarms due to dust and steam, which offer limited scene coverage. In contrast, image-based deep learning methods can monitor wide areas, reason about context, and detect early visual cues.

This thesis introduces F2M, a fusion model that combines outputs from multiple advanced neural networks to enhance fire segmentation in complex indoor scenes. F2M integrates Monte Carlo dropout during inference to estimate predictive uncertainty, so each pixel is labeled as fire or non-fire and receives an associated confidence score. We evaluated six representative semantic segmentation architectures: FPN, U-Net, U-Net++, MAnet, DeepLabV3+, and SegFormer. This set spans multi-scale encoder–decoder CNNs and a modern transformer-based design, providing architectural diversity. Our goal was to integrate models with complementary inductive biases, allowing F2M to combine diverse features and patterns for robust fire segmentation. Across experiments, F2M consistently surpassed the evaluated convolutional networks. In this research, we tested the hypothesis that combining the best-performing models as an ensemble would yield better results than using only the invididual best-performing model. It achieves higher Dice Scores and lower Total Error at several target resolutions. At lower resolutions, F2M delivers a 4.02% improvement over U-Net++ while maintaining robust generalization without overfitting. Through this combination of accuracy, adaptability, and interpretability, F2M provides a solid foundation for next-generation fire monitoring systems.

F2M enhances segmentation by combining multiple models, but its performance still relies on the quality and diversity of the training data. Pixel-level fire annotations are rare and costly to produce, especially for indoor scenes. To address this challenge, we investigated how synthetic data can help train deep networks for pixel-level fire segmentation when real data is limited. We introduced the SYN-FIRE dataset to study indoor fires in industrial environments, comprising five distinct scenarios that vary in time of day, camera viewpoint, and fire characteristics. The 3D scenes were created in NVIDIA Omniverse, and the resulting images were collected and annotated at the pixel-level to provide accurate ground truth masks for each image. SYN-FIRE dataset advances fire detection by providing a fully synthetic and publicly available dataset of indoor industrial fire scenes created with NVIDIA Omniverse, designed to train deep neural networks for pixel-level fire segmentation when real data is scarce. Benchmarks showed that models trained only on synthetic images performed worse than those trained on real images. However, combining synthetic data with real data consistently improved accuracy, with gains ranging from 4% to 10% on small datasets, such as BowFire, depending on the resolution. The dataset also reduces annotation effort and development time, as generating a comparable set of synthetic images took approximately 30 minutes, while collecting and labeling 40 real images required about three days. Procedural generation enables the rapid creation of large image sets with controlled variation in fire behavior, object placement, and environmental conditions. Current limitations in environmental diversity, realistic occlusions, human presence, and smoke complexity may constrain generalization if these synthetic datasets are used alone. We plan future versions with richer textures, lighting conditions, occlusion patterns, and fire dynamics, along with broader evaluations across modern deep neural network architectures.

Future work will continue to build on the foundations laid out in this doctoral thesis by exploring new directions for improving fire detection with limited real-world data. To achieve this, we will investigate methods that can replicate the complex visual characteristics of smoke and flames under varying conditions. This approach supports the broader aim of creating data-rich training pipelines that reduce the need for time-consuming manual collection and labeling, while still allowing for the creation of varied and realistic examples for model development. Building on this idea, semi-supervised learning provides a method for combining labeled and unlabeled data during training, thereby reducing reliance on large annotated datasets. At the same time, it enables the model to retain strong accuracy and generalization, making it more practical for real-world use, where labeled fire data may be limited or difficult to obtain. We will incorporate temporal information into our models to enable video-based fire detection. This will enable the algorithm to account for motion and temporal variations, which is particularly beneficial for detecting early-stage fires or reducing false positives. These future directions offer strong potential for advancing fire detection systems that are more adaptable and reliable. By combining synthetic data generation, reduced dependence on labels, and temporal awareness, we aim to address some of the core limitations outlined in this thesis and support safer and more responsive fire monitoring solutions across various real-world use cases.

Bibliography

[1] U. Wickström. Fire physics and chemistry. Springer. URL `https://urn.kb.se/resolve?urn=urn:nbn:se:ri:diva-646`.

[2] H. Sharma i N. Kanwal. Intelligent video-based fire detection: A novel dataset and real-time multi-stage classification approach. *Expert Systems with Applications*, 271: 126655, May 2025.

[3] T. Neale, A. Zahara i W. Smith. An Eternal Flame: The Elemental Governance of Wildfire's Pasts, Presents and Futures. 25(2).

[4] M. Shahid, J. J. Virtusio, Y.-H. Wu, Y.-Y. Chen, M. Tanveer, K. Muhammad i K.-L. Hua. Spatio-Temporal Self-Attention Network for Fire Detection and Segmentation in Video Surveillance. *Ieee Access*, 10:1259–1275, 2022.

[5] M. Arlovic, F. Hrzic, M. Patel, T. Bednarz i J. Balen. Evaluation of synthetic data impact on fire segmentation models performance. *Scientific Reports*, 15(1):16759, May 2025.

[6] F. Hržić, S. Tschauner, E. Sorantin i I. Štajduhar. Fracture recognition in paediatric wrist radiographs: An object detection approach. *Mathematics*, 10(16), 2022.

[7] A. George, C. Ecabert, H. O. Shahreza, K. Kotwal i S. Marcel. EdgeFace: Efficient Face Recognition Model for Edge Devices. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 6(2):158–168, Apr. 2024.

[8] H. Qiu, B. Yu, D. Gong, Z. Li, W. Liu i D. Tao. SynFace: Face Recognition With Synthetic Data. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, stranice 10880–10890, 2021.

[9] S. Patil, V. Varadarajan, S. Mahadevkar, R. Athawade, L. Maheshwari, S. Kumbhare, Y. Garg, D. Dharrao, P. Kamat i K. Kotecha. Enhancing Optical Character Recognition on Images with Mixed Text Using Semantic Segmentation. *Journal of Sensor and Actuator Networks*, 11(4):63, Dec. 2022.

[10] H.-S. Choi, M. Jeon, K. Song i M. Kang. Semantic Fire Segmentation Model Based on Convolutional Neural Network for Outdoor Image. *Fire Technology*, 57(6):3005–3019, Nov. 2021.

[11] M. Arlović, M. Patel, J. Balen i F. Hržić. F2M: Ensemble-based uncertainty estimation model for fire detection in indoor environments. *Engineering Applications of Artificial Intelligence*, 133:108428, July 2024.

[12] M. Arlovic, D. Damjanovic, F. Hrzic i J. Balen. Synthetic Dataset Generation Methods for Computer Vision Application. *2024 International Conference on Smart Systems and Technologies (SST)*, stranice 69–74, Oct. 2024.

[13] D. Drysdale. *An Introduction to Fire Dynamics.* John Wiley & Sons, Aug. 2011.

[14] J. G. Quintiere. *Fundamentals of Fire Phenomena.* John Wiley, Chichester, 2006.

[15] J. Yang, Z. Li, X. Liu, X. Ren, J. Wu, X. Xu, X. Bao, L. Jiang i J. Fang. Characteristics and toxicity of burning smoke released from non-metallic materials of ships in a closed environment. *Journal of Hazardous Materials*, 480:136109, Dec. 2024.

[16] J. Wang, R. Zhang, Y. Wang, L. Shi, S. Zhang, C. Li, Y. Zhang i Q. Zhang. Smoke filling and entrainment behaviors of fire in a sealed ship engine room. *Ocean Engineering*, 245:110521, Feb. 2022.

[17] Z. Gao, C. Li, W. Yan, Y. Fan i L. Jiang. Experimental study on the flame characteristics of ceiling jet with various air entrainment conditions and vertical fire positions. *International Journal of Heat and Mass Transfer*, 238:126467, Mar. 2025.

[18] T. Celik, H. Demirel, H. Ozkaramanli i M. Uyguroglu. Fire detection using statistical color model in video sequences. *Journal of Visual Communication and Image Representation*, 18(2):176–185, Apr. 2007.

[19] P. Maric, M. Arlovic, J. Balen, K. Vdovjak i D. Damjanovic. A Large Scale Dataset For Fire Detection and Segmentation in Indoor Spaces. *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, stranice 1–8, Nov. 2022. IEEE.

[20] R. L. Alpert. Calculation of response time of ceiling-mounted fire detectors. *Fire Technology*, 8(3):181–195, Aug. 1972.

[21] G. V. Kuznetsov, R. S. Volkov, A. S. Sviridenko i P. A. Strizhak. Reduction of response time of fire detection and containment systems in compartments. *Fire Safety Journal*, 144:104089, Mar. 2024.

[22] P. Foggia, A. Saggese i M. Vento. Real-Time Fire Detection for Video-Surveillance Applications Using a Combination of Experts Based on Color, Shape, and Motion. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(9):1545–1556, Sept. 2015.

[23] B. Butler, S. Quarles, C. Standohar-Alfano, M. Morrison, D. Jimenez, P. Sopko, C. Wold, L. Bradshaw, L. Atwood, J. Landon, J. O'Brien, B. Hornsby, N. Wagenbrenner i W. Page. Exploring fire response to high wind speeds: Fire rate of spread, energy release and flame residence time from fires burned in pine needle beds under winds up to 27 m s-1. *International Journal of Wildland Fire*, 29(1):81–92, Nov. 2019.

[24] Z. Liu i A. K. Kim. Review of Recent Developments in Fire Detection Technologies. *Journal of Fire Protection Engineering*, 13(2):129–151, May 2003.

[25] A. Jadon, M. Omama, A. Varshney, M. S. Ansari i R. Sharma. FireNet: A Specialized Lightweight Fire & Smoke Detection Model for Real-Time IoT Applications, Sept. 2019.

[26] A. Antunovic, M. Arlovic, J. Balen i L. Seric. Advances in Fire Detection and Suppression: A Review of Contemporary Methods and Technologies. *IEEE Access*, stranice 1–1, 2025.

[27] F. Khan, Z. Xu, J. Sun, F. M. Khan, A. Ahmed i Y. Zhao. Recent Advances in Sensors for Fire Detection. 22(9):3310.

[28] Majid Ghassemi i Azadeh Shahidian. *Nano and Bio Heat Transfer and Fluid Flow*. Academic Press, 2017.

[29] H. Bordbar, F. Alinejad, K. Conley, T. Ala-Nissila i S. Hostikka. Flame detection by heat from the infrared spectrum: Optimization and sensitivity analysis. *Fire Safety Journal*, 133:103673, Oct. 2022.

[30] J. Sidey, M. , E. i R. L. and Gordon. Simulations of Autoignition and Laminar Premixed Flames in Methane/Air Mixtures Diluted with Hot Products. *Combustion Science and Technology*, 186(4-5):453–465, May 2014.

[31] S. F. Sulthana, C. T. A. Wise, C. V. Ravikumar, R. Anbazhagan, G. Idayachandran i G. Pau. Review Study on Recent Developments in Fire Sensing Methods. *IEEE Access*, 11:90269–90282, 2023.

[32] J. Fonollosa, A. Solórzano i S. Marco. Chemical Sensor Systems and Associated Algorithms for Fire Detection: A Review. *Sensors*, 18(2):553, Feb. 2018.

[33] M. Yu, H. Yuan, K. Li i J. Wang. Research on multi-detector real-time fire alarm technology based on signal similarity. *Fire Safety Journal*, 136:103724, Apr. 2023.

[34] Q. Wu, Z. Ding i W. Zhang. Research progress on electrochemical gas sensors for fire detection. *International Journal of Electrochemical Science*, 20(7):101043, July 2025.

[35] A. Gaur, A. Singh, A. Kumar, K. S. Kulkarni, S. Lala, K. Kapoor, V. Srivastava, A. Kumar i S. C. Mukhopadhyay. Fire Sensing Technologies: A Review. *IEEE Sensors Journal*, 19(9):3191–3202, May 2019.

[36] H. Hoff. Using Distributed Fibre Optic Sensors for Detecting Fires and Hot Rollers on Conveyor Belts. *2017 2nd International Conference for Fibre-optic and Photonic Sensors for Industrial and Safety Applications (OFSIS)*, stranice 70–76, Jan. 2017.

[37] C.-F. Cao, B. Yu, Z.-Y. Chen, Y.-X. Qu, Y.-T. Li, Y.-Q. Shi, Z.-W. Ma, F.-N. Sun, Q.-H. Pan, L.-C. Tang, P. Song i H. Wang. Fire Intumescent, High-Temperature Resistant, Mechanically Flexible Graphene Oxide Network for Exceptional Fire Shielding and Ultra-Fast Fire Warning. *Nano-Micro Letters*, 14(1):92, Apr. 2022.

[38] X. He, F. Xu, F.-F. Chen i Y. Yu. DNA-Modified Graphene Oxide Supported on a Fire-Resistant Hydroxyapatite Paper for Timely and Reliable Fire Warning. *ACS Applied Nano Materials*, 6(13):11612–11621, July 2023.

[39] D. R. Dreyer, S. Park, C. W. Bielawski i R. S. Ruoff. The chemistry of graphene oxide. *Chemical Society Reviews*, 39(1):228–240, Dec. 2009.

[40] A. Jiříčková, O. Jankovský, Z. Sofer i D. Sedmidubský. Synthesis and Applications of Graphene Oxide. *Materials*, 15(3):920, Jan. 2022.

[41] C.-F. Cao, B. Yu, B.-F. Guo, W.-J. Hu, F.-N. Sun, Z.-H. Zhang, S.-N. Li, W. Wu, L.-C. Tang, P. Song i H. Wang. Bio-inspired, sustainable and mechanically robust graphene oxide-based hybrid networks for efficient fire protection and warning. *Chemical Engineering Journal*, 439:134516, July 2022.

[42] X. Xu, J. Huang, G. Miao, B. Yan, Y. Chen, Y. Zhou, Y. Zhang, X. Zhang i W. Cai. Visualizing Thermal Reduction in Graphene Oxide. *Materials*, 18(10):2222, Jan. 2025.

[43] S. Prezioso, F. Perrozzi, L. Giancaterini, C. Cantalini, E. Treossi, V. Palermo, M. Nardone, S. Santucci i L. Ottaviano. Graphene Oxide as a Practical Solution to High Sensitivity Gas Sensing. *The Journal of Physical Chemistry C*, 117(20):10683–10690, May 2013.

[44] S. Z. N. Demon, A. I. Kamisan, N. Abdullah, S. A. M. Noor, O. K. Khim, N. A. M. Kasim, M. Z. A. Yahya, N. A. A. Manaf, A. F. M. Azmi i N. A. Halim. Graphene-based

Materials in Gas Sensor Applications: A Review. *Sensors and Materials*, 32(2):759, Feb. 2020.

[45] A. Chakraborthy, S. Nuthalapati, A. Nag, N. Afsarimanesh, M. E. E. Alahi i M. E. Altinsoy. A Critical Review of the Use of Graphene-Based Gas Sensors. *Chemosensors*, 10(9):355, Sept. 2022.

[46] R. Bose, A. Alanazi K., S. Bhowmik, S. Garai, M. Roy, B. Pakhira i T. Pramanik. Applications of Graphene and Graphene Oxide as Versatile Sensors: A Brief Review. *Biointerface Research in Applied Chemistry*, 13(5):457, Oct. 2023.

[47] E. Lee, D. Lee, J. Yoon, Y. Yin, Y. N. Lee, S. Uprety, Y. S. Yoon i D.-J. Kim. Enhanced Gas-Sensing Performance of GO/TiO2 Composite by Photocatalysis. *Sensors*, 18(10): 3334, Oct. 2018.

[48] K. Chen, Y. Cheng, H. Bai, C. Mou i Y. Zhang. Research on Image Fire Detection Based on Support Vector Machine. *2019 9th International Conference on Fire Science and Fire Protection Engineering (ICFSFPE)*, stranice 1–7, Oct. 2019.

[49] M. Ali, I. Ahmad, I. Geun, S. A. Hamza, U. Ijaz, Y. Jang, J. Koo, Y.-G. Kim i H.-D. Kim. A Comprehensive Review of Advanced Sensor Technologies for Fire Detection with a Focus on Gasistor-Based Sensors. *Chemosensors*, 13(7):230, July 2025.

[50] J. Fonollosa, A. Solórzano i S. Marco. Chemical Sensor Systems and Associated Algorithms for Fire Detection: A Review. *Sensors*, 18(2):553, Feb. 2018.

[51] S. P. Praveen, P. N. Srinivasu, J. Shafi, M. Wozniak i M. F. Ijaz. ResNet-32 and FastAI for diagnoses of ductal carcinoma from 2D tissue slides. *Scientific Reports*, 12 (1):20804, Dec. 2022.

[52] S. Li, P. Liu, Q. Yan i R. Qian. Optimized Deep Learning Model for Fire Semantic Segmentation. *Cmc-Computers Materials & Continua*, 72(3):4999–5013, 2022.

[53] J. Balen, D. Damjanovic, P. Maric, K. Vdovjak, M. Arlovic i G. Martinovic. FireBot - An Autonomous Surveillance Robot for Fire Prevention, Early Detection and Extinguishing. *2023 15th International Conference on Computer and Automation Engineering (ICCAE)*, stranice 400–405, Mar. 2023.

[54] K. Muhammad, J. Ahmad i S. W. Baik. Early fire detection using convolutional neural networks during surveillance for effective disaster management. *Neurocomputing*, 288: 30–42, May 2018.

[55] T.-H. Chen, P.-H. Wu i Y.-C. Chiou. An early fire-detection method based on image processing. *2004 International Conference on Image Processing, 2004. ICIP '04.*, svezak 3, stranice 1707–1710 Vol. 3, Oct. 2004.

[56] B. U. Toreyin, Y. Dedeoglu i A. E. Cetin. Contour based smoke detection in video using wavelets. *2006 14th European Signal Processing Conference*, stranice 1–5, Sept. 2006.

[57] S. Basar, M. Ali, G. Ochoa-Ruiz, M. Zareei, A. Waheed i A. Adnan. Unsupervised color image segmentation: A case of RGB histogram based K-means clustering initialization. *PLOS ONE*, 15(10):e0240015, Oct. 2020.

[58] N. Dhanachandra, K. Manglem i Y. J. Chanu. Image Segmentation Using K -means Clustering Algorithm and Subtractive Clustering Algorithm. *Procedia Computer Science*, 54:764–771, Jan. 2015.

[59] S. Rudz, K. Chetehouna, A. Hafiane, H. Laurent i O. Séro-Guillaume. Investigation of a novel image segmentation method dedicated to forest fire applications*. *Measurement Science and Technology*, 24(7):075403, June 2013.

[60] Y. Li, A. Vodacek i Y. Zhu. An automatic statistical segmentation algorithm for extraction of fire and smoke regions. 108(2):171–178.

[61] S. Anitha. An approach for identifying the forest fire using land surface imagery by locating the abnormal temperature distribution. *IOSR Journal of Computer Engineering*, 14(3):06–12, 2013.

[62] S. Beucher i F. Meyer. The Morphological Approach to Segmentation: The Watershed Transformation. *Mathematical Morphology in Image Processing*. CRC Press, 1992. ISBN 978-1-315-21461-0.

[63] L. Najman i M. Couprie. Watershed Algorithms and Contrast Preservation. I. Nyström, G. Sanniti di Baja i S. Svensson, editors, *Discrete Geometry for Computer Imagery*, stranice 62–71, 2003. Springer.

[64] Sakshi i V. Kukreja. Segmentation and Contour Detection for handwritten mathematical expressions using OpenCV. *2022 International Conference on Decision Aid Sciences and Applications (DASA)*, stranice 305–310, Mar. 2022.

[65] P. Arbeláez, M. Maire, C. Fowlkes i J. Malik. Contour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, May 2011.

[66] B. U. Töreyin, Y. Dedeoğlu, U. Güdükbay i A. E. Çetin. Computer vision based method for real-time fire and flame detection. *Pattern Recognition Letters*, 27(1): 49–58, Jan. 2006.

[67] Z. Zhang, J. Zhao, D. Zhang, C. Qu, Y. Ke i B. Cai. Contour Based Forest Fire Detection Using FFT and Wavelet. *2008 International Conference on Computer Science and Software Engineering*, svezak 1, stranice 760–763, Dec. 2008.

[68] A. Voulodimos, N. Doulamis, A. Doulamis i E. Protopapadakis. Deep Learning for Computer Vision: A Brief Review. *Computational Intelligence and Neuroscience*, 2018 (1):7068349, Jan. 2018.

[69] A. Krizhevsky, I. Sutskever i G. E. Hinton. ImageNet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, May 2017.

[70] R. Archana i P. S. E. Jeevaraj. Deep learning models for digital image processing: A review. *Artificial Intelligence Review*, 57(1):11, Jan. 2024.

[71] Ò. Lorente, I. Riera i A. Rana. Image Classification with Classic and Deep Learning Techniques, May 2021.

[72] H. Harkat, J. M. P. Nascimento, A. Bernardino i H. F. T. Ahmed. Fire images classification based on a handcraft approach. *Expert Systems with Applications*, 212: 118594, Feb. 2023.

[73] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho i S. W. Baik. Convolutional Neural Networks Based Fire Detection in Surveillance Videos. 6:18174–18183.

[74] E. Tsalera, A. Papadakis, I. Voyiatzis i M. Samarakou. CNN-based, contextualized, real-time fire detection in computational resource-constrained environments. *Energy Reports*, 9:247–257, Sept. 2023.

[75] F. Neha, D. Bhati, D. K. Shukla i M. Amiruzzaman. From classical techniques to convolution-based models: A review of object detection algorithms. URL `http://arxiv.org/abs/2412.05252`.

[76] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu i M. Pietikäinen. Deep Learning for Generic Object Detection: A Survey. *International Journal of Computer Vision*, 128(2):261–318, Feb. 2020.

[77] A. Islam i M. I. Habib. Fire Detection From Image and Video Using YOLOv5. URL `http://arxiv.org/abs/2310.06351`.

[78] L. He, Y. Zhou, L. Liu, Y. Zhang i J. Ma. Research and application of deep learning object detection methods for forest fire smoke recognition. 15(1):16328.

[79] P. Barmpoutis, K. Dimitropoulos, K. Kaza i N. Grammalidis. Fire Detection from Images Using Faster R-CNN and Multidimensional Texture Analysis. *ICASSP 2019*

- *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, stranice 8301–8305.

[80] Y. Liu, C. Bo i C. Feng. FB-YOLOv8s: A fire detection algorithm based on YOLOv8s. 5:240–248.

[81] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz i D. Terzopoulos. Image Segmentation Using Deep Learning: A Survey. URL `http://arxiv.org/abs/2001.05566`.

[82] F. Hou, X. Rui, Y. Chen i X. Fan. Flame and Smoke Semantic Dataset: Indoor Fire Detection with Deep Semantic Segmentation Model. *Electronics*, 12(18):3778, Jan. 2023.

[83] M. Li, Y. Zhang, L. Mu, J. Xin, Z. Yu, S. Jiao, H. Liu, G. Xie i Y. Yingmin. A Real-time Fire Segmentation Method Based on A Deep Learning Approach. *IFAC-PapersOnLine*, 55(6):145–150, Jan. 2022.

[84] H. Zhou, X. Wang, K. Xia, Y. Ma i G. Yuan. Transfer Learning-Based Hyperspectral Image Classification Using Residual Dense Connection Networks. *Sensors*, 24(9):2664, Jan. 2024.

[85] J. Sharma, O.-C. Granmo, M. Goodwin i J. T. Fidje. Deep Convolutional Neural Networks for Fire Detection in Images. G. Boracchi, L. Iliadis, C. Jayne i A. Likas, editors, *Engineering Applications of Neural Networks*, Communications in Computer and Information Science, stranice 183–193, 2017. Springer International Publishing.

[86] A. J. Dunnings i T. P. Breckon. Experimentally Defined Convolutional Neural Network Architecture Variants for Non-Temporal Real-Time Fire Detection. *2018 25th IEEE International Conference on Image Processing (ICIP)*, stranice 1558–1562, Oct. 2018.

[87] Y. Xie, J. Zhu, Y. Cao, Y. Zhang, D. Feng, Y. Zhang i M. Chen. Efficient Video Fire Detection Exploiting Motion-Flicker-Based Dynamic Features and Deep Static Features. *IEEE Access*, 8:81904–81917, 2020.

[88] F. Hou, X. Rui, Y. Chen i X. Fan. Flame and Smoke Semantic Dataset: Indoor Fire Detection with Deep Semantic Segmentation Model. *Electronics*, 12(18):3778, Jan. 2023. Number: 18 Publisher: Multidisciplinary Digital Publishing Institute.

[89] W. S. Mseddi, R. Ghali, M. Jmal i R. Attia. Fire Detection and Segmentation using YOLOv5 and U-NET. *2021 29th European Signal Processing Conference (EUSIPCO)*, stranice 741–745, Aug. 2021.

[90] B. Kim i J. Lee. A Video-Based Fire Detection Using Deep Learning Models. *Applied Sciences*, 9(14):2862, Jan. 2019.

[91] M. Niknejad i A. Bernardino. Attention on Classification for Fire Segmentation. *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, stranice 616–621, Dec. 2021.

[92] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan i S. Belongie. Feature Pyramid Networks for Object Detection, Apr. 2017.

[93] O. Ronneberger, P. Fischer i T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation, May 2015.

[94] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh i J. Liang. UNet++: A Nested U-Net Architecture for Medical Image Segmentation, July 2018.

[95] R. Li, S. Zheng, C. Duan, C. Zhang, J. Su i P. M. Atkinson. Multi-Attention-Network for Semantic Segmentation of Fine Resolution Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022.

[96] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff i H. Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation, Aug. 2018.

[97] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez i P. Luo. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers, Oct. 2021.

[98] A. Bauer, S. Trapp, M. Stenger, R. Leppich, S. Kounev, M. Leznik, K. Chard i I. Foster. Comprehensive Exploration of Synthetic Data Generation: A Survey, Feb. 2024.

[99] P. P. Khaing i M. T. Yu. A Survey in Deep Learning Model for Image Annotation. *International Journal of Computer (IJC)*, 32(1):54–63, Mar. 2019.

[100] CVAT.ai Corporation. Computer Vision Annotation Tool (CVAT), Nov. 2023. URL `https://github.com/cvat-ai/cvat`.

[101] P. Skalski. Make Sense. `https://github.com/SkalskiP/make-sense/`, 2019.

[102] B. C. Russell, A. Torralba, K. P. Murphy i W. T. Freeman. LabelMe: A Database and Web-Based Tool for Image Annotation. 77(1):157–173.

[103] Labelbox | The data factory for AI teams. URL `https://labelbox.com`.

[104] T. Toulouse, L. Rossi, A. Campana, T. Celik i M. A. Akhloufi. Computer vision for wildfire research: An evolving image dataset for processing and analysis. *Fire Safety Journal*, 92:188–194, Sept. 2017.

[105] D. Y. T. Chino, L. P. S. Avalhais, J. F. Rodrigues i A. J. M. Traina. BoWFire: Detection of Fire in Still Images by Integrating Pixel Color and Texture Analysis. *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*, stranice 95–102, Aug. 2015.

[106] P. V. A. B. de Venâncio, A. C. Lisboa i A. V. Barbosa. An automatic fire detection system based on deep convolutional neural networks for low-power, resource-constrained devices. *Neural Computing and Applications*, 34(18):15349–15368, Sept. 2022.

[107] A. Shamsoshoara, F. Afghah, A. Razi, L. Zheng, P. Z. Fulé i E. Blasch. Aerial Imagery Pile burn detection using Deep Learning: The FLAME dataset. *Computer Networks*, (arXiv:2012.14036), Dec. 2020.

[108] X. Chen, B. Hopkins, H. Wang, L. O'Neill, F. Afghah, A. Razi, P. Fulé, J. Coen, E. Rowell i A. Watts. Wildland Fire Detection and Monitoring Using a Drone-Collected RGB/IR Image Dataset. 10:121301–121317.

[109] S. Wu, X. Zhang, R. Liu i B. Li. A dataset for fire and smoke object detection. *Multimed Tools Appl*, 82(5):6707–6726, Feb. 2023.

[110] Y. Hu, X. Ye, Y. Liu, S. Kundu, G. Datta, S. Mutnuri, N. Asavisanu, N. Ayanian, K. Psounis i P. Beerel. FireFly A Synthetic Dataset for Ember Detection in Wildfire, Aug. 2023.

[111] L. Fernando, R. Ghali i M. A. Akhloufi. SWIFT: Simulated Wildfire Images for Fast Training Dataset. *Remote Sensing*, 16(9):1627, Jan. 2024.

[112] J. Hu, L. Shen, S. Albanie, G. Sun i E. Wu. Squeeze-and-Excitation Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8):2011–2023, Aug. 2020.

[113] I. Loshchilov i F. Hutter. Decoupled Weight Decay Regularization, Jan. 2019.

[114] L. N. Smith i N. Topin. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates, May 2018.

[115] F. Hržić, M. Janisch, I. Štajduhar, J. Lerga, E. Sorantin i S. Tschauner. Modeling uncertainty in fracture age estimation from pediatric wrist radiographs. *Mathematics*, 9(24), 2021.

[116] S. Lundberg i S.-I. Lee. A Unified Approach to Interpreting Model Predictions. URL `http://arxiv.org/abs/1705.07874`.

[117] Y. Lu, M. Shen, H. Wang, X. Wang, C. van Rechem, T. Fu i W. Wei. Machine Learning for Synthetic Data Generation: A Review, May 2024.

[118] F. Lucini. *The Real Deal About Synthetic Data.* MIT Sloan Management Review, 2021.

[119] Blender Development Team. Blender (Version 4.5 LTS). URL `https://www.blender.org/`.

[120] The most powerful real-time 3D creation tool - Unreal Engine. URL `https://www.unrealengine.com/en-US`.

[121] NVIDIA Omniverse. URL `https://www.nvidia.com/en-us/omniverse/`.

[122] R. Rombach, A. Blattmann, D. Lorenz, P. Esser i B. Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. URL `http://arxiv.org/abs/2112.10752`.

[123] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville i Y. Bengio. Generative Adversarial Networks, June 2014.

[124] D. P. Kingma i M. Welling. An Introduction to Variational Autoencoders. 12(4): 307–392.

[125] A. Kishore, T. E. Choe, J. Kwon, M. Park, P. Hao i A. Mittel. Synthetic Data Generation using Imitation Training. *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, stranice 3071–3079, Oct. 2021. IEEE.

[126] D. P. Rohe i E. M. C. Jones. Generation of Synthetic Digital Image Correlation Images Using the Open-Source Blender Software. *Experimental Techniques*, 46(4):615–631, Aug. 2022.

[127] G. R. Koteswara Rao, P. Vidya Sgar, T. Bikku, C. Prasad i N. Cherukuri. Comparing 3D Rendering Engines in Blender. *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, stranice 489–495, Oct. 2021.

[128] Blender Documentation Team. Blender 4.5 LTS Reference Manual. Eevee - Blender Developer Documentation. URL `https://developer.blender.org/docs/features/eevee/`.

[129] A. I. Károly, I. Nádas i P. Galambos. Synthetic Multimodal Video Benchmark (SMVB): Utilizing Blender for rich dataset generation. *2024 IEEE 22nd World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, stranice 000065–000070, Jan. 2024.

[130] A. I. Károly i P. Galambos. Automated Dataset Generation with Blender for Deep Learning-based Object Segmentation. *2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, stranice 000329–000334, Mar. 2022.

[131] A. I. Károly, I. Nádas i P. Galambos. Synthetic Multimodal Video Benchmark (SMVB): Utilizing Blender for rich dataset generation. *2024 IEEE 22nd World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, stranice 000065–000070, Jan. 2024.

[132] M. Orić, V. Galić i F. Novoselnik. Synthetic dataset generation system for vehicle detection. *Software Impacts*, 23:100735, Mar. 2025.

[133] M. Džijan, R. Grbić, I. Vidović i R. Cupec. Towards fully synthetic training of 3D indoor object detectors: Ablation study. *Expert Systems with Applications*, 232:120723, Dec. 2023.

[134] S. Arezoomandan, J. Klohoker i D. K. Han. Analyzing the Efficacy of Synthetic Images in Unmanned Aerial Vehicle Detection. *2024 IEEE International Conference on Consumer Electronics (ICCE)*, stranice 1–6, Jan. 2024.

[135] L. Orvalho. Using Unreal Engine 5 to realistic rendering of scenes.

[136] D. Agarwal, T. Kucukpinar, J. Fraser, J. Kerley, A. R. Buck, D. T. Anderson i K. Palaniappan. Simulating City-Scale Aerial Data Collection Using Unreal Engine. *2023 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, stranice 1–9, Sept. 2023.

[137] Lumen Technical Details in Unreal Engine | Unreal Engine 5.5 Documentation | Epic Developer Community. https://dev.epicgames.com/documentation/en-us/unreal-engine/lumen-technical-details-in-unreal-engine.

[138] GPU Lightmass Global Illumination in Unreal Engine | Unreal Engine 5.5 Documentation | Epic Developer Community. https://dev.epicgames.com/documentation/en-us/unreal-engine/gpu-lightmass-global-illumination-in-unreal-engine.

[139] Y. Hu, X. Ye, Y. Liu, S. Kundu, G. Datta, S. Mutnuri, N. Asavisanu, N. Ayanian, K. Psounis i P. Beerel. FireFly A Synthetic Dataset for Ember Detection in Wildfire, Aug. 2023.

[140] L. Fernando, R. Ghali i M. A. Akhloufi. SWIFT: Simulated Wildfire Images for Fast Training Dataset. *Remote Sensing*, 16(9):1627, Jan. 2024.

[141] D. Conde, J. Martínez, J. Balado, P. Arias i U. de Vigo. Generation of road zone synthetic data for training MOT models with the NVIDIA Omniverse platform. 2023.

[142] D. Conde, J. Martínez, J. Balado, P. Arias i U. de Vigo. Generation of road zone synthetic data for training MOT models with the NVIDIA Omniverse platform.

[143] Y. Kataoka, T. Matsubara i K. Uehara. Image generation using generative adversarial networks and attention mechanism. *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, stranice 1–6, June 2016.

[144] F.-A. Croitoru, V. Hondru, R. T. Ionescu i M. Shah. Diffusion Models in Vision: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9): 10850–10869, Sept. 2023.

[145] Karthika. S i M. Durgadevi. Generative Adversarial Network (GAN): A general review on different variants of GAN and applications. *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, stranice 1–8, July 2021.

[146] J. Islam i Y. Zhang. GAN-based synthetic brain PET image generation. *Brain Informatics*, 7(1):3, Mar. 2020.

[147] M. Abduljawad i A. Alsalmani. Towards Creating Exotic Remote Sensing Datasets using Image Generating AI. *2022 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, stranice 84–88, Nov. 2022.

[148] A. Nathanail. Geo Fossils-I: A synthetic dataset of 2D fossil images for computer vision applications on geology. *Data in Brief*, 48:109188, June 2023.

[149] P. V. A. B. de Venâncio, A. C. Lisboa i A. V. Barbosa. An automatic fire detection system based on deep convolutional neural networks for low-power, resource-constrained devices. *Neural Computing and Applications*, 34(18):15349–15368, Sept. 2022.

[150] K. He, X. Zhang, S. Ren i J. Sun. Deep Residual Learning for Image Recognition, Dec. 2015.

[151] Fab. URL `https://www.fab.com/`.

[152] G. Ding, I. Georgilas, A. Plummer, G. Ding, I. Georgilas i A. Plummer. A Deep Learning Model with a Self-Attention Mechanism for Leg Joint Angle Estimation across Varied Locomotion Modes. *Sensors*, 24(1), Dec. 2023.

[153] J. Åkesson, J. Töger i E. Heiberg. Random effects during training: Implications for deep learning-based medical image segmentation. *Computers in Biology and Medicine*, 180:108944, Sept. 2024.

[154] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh i D. Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision*, 128(2):336–359, Feb. 2020.

Curriculum Vitae

Matej Arlović was born on May 5, 1996, in Osijek, Croatia. He completed his primary education at Ljudevit Gaj Elementary School and his secondary education at the Electrical Engineering and Traffic School in Osijek. In 2017, he obtained a Bachelor's degree in Electrical Engineering with a specialization in Automation. After completing a bridging program, he obtained a Master's degree in Computer Science with a focus in Software Engineering in 2021. That same year, he began his PhD studies at the J. J. Strossmayer University of Osijek and is now employed as a Teaching and Research Assistant at the Faculty of Electrical Engineering, Computer Science, and Information Technology in Osijek.

U Osijeku, 2026.

Matej Arlović