Fire Detection Based on Surveillance Camera Using Transfer Learning

Tolera Tamiru Negasa

A Thesis Submitted to the Department of Computer Science and Engineering
School of Electrical Engineering and Computing
Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies
Adama Science and Technology University

June 2024

Adama, Ethiopia

Fire Detection Based on Surveillance Cameras Using Transfer Learning

Tolera Tamiru Negasa

Advisor: Bahiru Shifaw Yimer (Ph.D.)

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

June 2024

Adama, Ethiopia

DECLARATION

I hereby declare that this Master Thesis entitled "**Fire Detection Based on Surveillance Camera using Transfer Learning**" is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Tolera Tamiru	_____	_____
Name of student	Signature	Date

RECOMMENDATION

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled "**Fire Detection Based on Surveillance Camera using Transfer Learning**" prepared under my guidance by **Tolera Tamiru Negasa** submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering. Therefore, I recommend the submission of a revised version of the thesis to the department following the applicable procedures.

Dr. Bahiru Shifaw Yimer (Ph.D.) _____ _____

Major Advisor Signature Date

APPROVAL SHEET

I, the advisor of the thesis entitled "**Fire Detection Based on Surveillance Camera using Transfer Learning**" and developed by Tolera Tamiru Negasa, hereby certify that the recommendations and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Dr. Bahiru Shifaw Yimer (Ph.D.) _____ _____

Major Advisor Signature Date

We, the undersigned, members of the Board of Examiners of the thesis by **Tolera Tamiru Negasa** have read and evaluated the thesis entitled "**Fire Detection Based on Surveillance Camera using Transfer Learning**" and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science and Engineering.

_____ _____ _____

Chairperson Signature Date

_____ _____ _____

Internal Examiner Signature Date

_____ _____ _____

External Examiner Signature Date

Finally, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

_____ _____ _____

Department Head Signature Date

_____ _____ _____

School Dean Signature Date

_____ _____ _____

Office of Postgraduate Studies, Dean Signature Date

ACKNOWLEDGMENT

First and foremost, I would like to express my deepest gratitude to my faith, which has been a constant source of strength and encouragement throughout the challenging journey of completing this thesis. It has reminded me of the unwavering presence of a higher power that guides me.

I would like to extend my heartfelt gratitude to my esteemed advisor, Dr. Bahiru Shifaw Yimer, Ph.D. His unwavering support, guidance, and insightful suggestions played a pivotal role in my thesis journey. His expertise and dedication not only shaped the direction of my research but also helped me refine my writing and presentation skills.

I am deeply grateful to Dr. Mesfin Abebe, Ph.D., for his invaluable insights, comments, and recommendations throughout this study. His expertise and guidance have been instrumental in shaping and enhancing the quality of this work.

Lastly, I would like to express my deepest gratitude to the members of the Artificial Intelligence SIG, my family, and my friends for their unwavering support and motivation throughout my thesis journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

1D	One Dimensional
2D	Two Dimensional
CCTV	Closed-Circuit Television
CNN	Convolutional Neural Networks
CPU	Central Processing Unit
DFS	Fire and Smoke Detection
FN	False Negatives
FP	False Positives
GB	Giga Bytes
GPU	Graphics Processing Unit
HIS	Hue, Saturation, and Intensity
IoT	Internet-of-Things
LED	Light-Emitting Diode
RAM	Random Access Memory
RELU	Rectified Linear Unit
RGB	Red, Green, Blue
RoR	Rate of Rise
RTX	Ray Tracing Texel eXtreme
SGD	Stochastic Gradient Descent

SVM	Support Vector Machine
TN	True Negative
TP	True Positive
UHD	Ultra High Definition
UML	Unified Modeling Language
UV	UltraViolet
VGG	Visual Geometry Group

ABSTRACT

Fire detection is a crucial task within surveillance systems, holding significant importance in preventing fire-related incidents and protecting individuals and properties. However, the presence of uncertainties in surveillance, characterized by unclear, incomplete, or untrustworthy information obtained through surveillance systems, poses challenges to efficient fire detection processes. Additionally, existing fire detection models face several issues, including extensive computational requirements, large model sizes, and many parameters. The primary objective of this study is to develop fire detection systems that can accurately detect the existence of fires in uncertain surveillance environments using transfer learning. To achieve this goal, we utilized four pre-existing convolutional neural network models (VGG19, MobileNetV2, ResNet50, InceptionV3) to extract features specific to fires from images depicting fire incidents. The training phase involved the utilization of a dataset comprising 14,850 images, while the evaluation phase tested the models' performance using 1,650 images. These images were gathered from various benchmark research papers and online sources, chosen according to specific environmental condition criteria. Hyperparameter optimization further enhanced the system's performance. The outcomes of this study revealed a remarkable level of accuracy in the classification of fire detection, with rates of 98.36% accuracy, 99.27% recall, 98.37% F1-score, and 97.50% precision for VGG19; 95.75% accuracy, 92.37% recall, 95.61% F1-score, and 99.09% precision for MobileNetV2; 98.60% accuracy, 98.06% recall, 98.59% F1-score, and 99.14% precision for ResNet50; and 98.36% accuracy, 99.15% recall, 98.37% F1-score, and 97.61% precision for InceptionV3.These successful results using transfer learning for robust fire detection in uncertain environments suggest a promising future for improving fire safety through advanced surveillance systems.

Keywords: *fire detection, Transfer learning, pre-trained models, Surveillance, Uncertain Environments, Deep learning, Hyperparameter optimization.*

CHAPTER ONE

1 INTRODUCTION

1.1 Background of the Study

Fire is one of the dangers that endanger human lives and destroy property. Fire detection and prevention are crucial measures for protecting lives and property. According to the Addis Ababa Fire and Disaster Risk Management Commission's 2015 Ethiopian calendar report, 326 fire incidents occurred. These fires, encompassing residential structures, factories, and vehicles, resulted in 9 fatalities and 122 injuries. The report further highlights the significant economic impact of these fires, estimating property damage at 797,690,000 birrs.[1] Additionally, according to a report by (Hall & Evarts, 2022), fire departments across the United States responded to an estimated 1.35 million fires in 2021. These fires tragically caused around 3,800 civilian deaths, 14,700 injuries, and a staggering $15.9 billion in direct property damage. Earlier works on fire detection systems have proven effective, and advancements in technology have led to the development of more sophisticated and nuanced solutions. Previous studies of fire detection systems rely on fixed rules and threshold values to identify potential fires. These systems trigger an alarm by analyzing specific factors such as smoke concentration, temperature changes, and heat patterns(Courbat et al., 2011; Fonollosa et al., 2016; Hassan & Audu, 2022; F. Khan et al., 2022a; Perilla et al., 2018).

Uncertainty in surveillance environments refers to anything that can hinder the ability of a surveillance system to interpret the scene accurately and reliably it's seeing. These uncertainties can arise from various factors and can significantly affect the system's performance. Additionally, Uncertainty surveillance encompasses situations where the information acquired by surveillance systems is unclear, incomplete, or untrustworthy(Texier et al., 2017). Such circumstances challenge the system in accurately interpreting ongoing events and making well-informed decisions. Fire detection systems rely on accurately interpreting visual or sensor data to identify fires. However, various environmental factors and limitations can introduce uncertainty into this process, leading to missed detections or false alarms.

[1] (Addis Ababa Fire and Disaster Risk Management Commission annual report, 2015)

Environmental factors such as smoke, snow, fog, dust storms, rain, lighting situations, low-resolution or damaged cameras, poorly positioned cameras, and background clutter can significantly affect the accuracy and reliability of fire detection systems. Smoke, while a common byproduct of fire and beneficial for detection, can also originate from other sources like chimneys and industrial processes, creating false positives. Snow, fog, and dust storms can obscure the fire or mimic smoke, making distinguishing fire from background elements challenging(Chen, 2023). Heavy rain can extinguish small fires or obscure flames, leading to missed detections. Low light conditions can make the fire faint and difficult to detect by cameras relying on visible light. In contrast, bright sunlight can cause glare and reflections that might be mistaken for fire, leading to false positives. Fire detection often depends on identifying flame characteristics like flickering or specific colors; however, low-resolution cameras can blur these details, making it hard to differentiate fire from other light sources(Ivanova et al., 2024). Damaged cameras with malfunctioning pixels or blurry lenses can further compromise image quality, resulting in missed detections. Poorly positioned cameras, such as those placed in blind spots or at unsuitable angles, might miss crucial areas where fires are likely to start or spread. For instance, a camera pointed at a wall might not capture a fire behind it. Additionally, complex backgrounds with moving objects, such as swaying trees or people walking, or distracting elements like sunlight reflections and flickering lights, can overwhelm fire detection algorithms, which might mistake these elements for fire, leading to false positives(Singh et al., 2023).

This highlights the critical need for improved fire detection, especially in environments with uncertain surveillance conditions. Early and accurate fire detection saves lives and minimizes property loss. The previous methods, often reliant on manual monitoring, struggle to handle the complexities of real-world scenarios. Recently, these systems have garnered considerable attention and have proven instrumental in protecting both individuals and property from the dangers posed by fire incidents. These systems often result in delayed or missed recognitions, false warnings, and other related problems. (Yogesh et al., 2022). This is where advancements in fire detection for uncertain environments using a transfer learning approach come into play. This research investigates the potential of transfer learning to address these challenges and improve fire detection performance in real-world surveillance scenarios. Transfer learning is a machine learning technique where knowledge gained from a pre-trained model on a large, general dataset is applied to a new, specific task. Collecting large datasets with diverse fire

scenarios and uncertainties can be costly and time-consuming. By exploring the capabilities of transfer learning and its potential to overcome data scarcity and improve generalizability, this study aims to contribute to the advancement of robust fire detection systems for real-world applications(Zhuang et al., 2021).

1.2 The Motivation for the Study

To detect fires, researchers have combined or customized different sensors. However, these systems can have limitations such as being prone to false alarms or inability to detect fires in their initial stages(Chino et al., 2015). Vision-based fire detection systems using convolutional neural networks (CNNs) have shown promise in overcoming these limitations. However, existing CNN-based fire detection systems are often computationally expensive, making them difficult to implement in real-world surveillance scenarios.(Majid et al., 2022; Muhammad, Ahmad, et al., 2019). Research on fire detection in uncertain surveillance environments using transfer learning is driven by the urgent need to enhance safety measures in challenging scenarios. Accurate fire detection in places where visibility is poor or obstructed by several factors such as low lighting or obstacles is crucial for preventing potential harm to lives and property. Transfer learning emerges as a powerful tool for enhancing the accuracy and reliability of fire detection systems, especially within challenging surveillance environments. This study aims to optimize surveillance systems, enable quicker emergency responses, and overcome the unique challenges presented by uncertain surveillance conditions, ultimately striving for better safety, and reducing the impact of fire incidents.

1.3 Statement of the Problem

Fire detection in uncertain surveillance contexts using transfer learning is an active research topic. Recent research proposes effective CNN-based algorithms for fire detection in images/ videos taken in uncertain surveillance circumstances(Muhammad, Khan, et al., 2019). Accurate detection of fires in uncertain surveillance environments, such as low-light conditions, complex backgrounds, and unpredictable surveillance scenarios, is still a significant challenge. Existing fire detection systems often struggle to recognize fire indicators and non-fire incidents in challenging environments, leading to false alarms, and missed detection. In addition, the real-time performance of fire detection systems is crucial for prompt emergency responses.

The recent gaps in fire detection within uncertain surveillance environments have brought to attention the urgent requirement for deep learning models that are more specifically designed and efficient. These models should prioritize quicker inference time and resilience under different conditions, enabling early detection of fires. Earlier studies have identified drawbacks including extensive computational requirements, lengthy inference times, large model sizes, and a high number of parameters(Ahmad et al., 2023) (Muhammad, Khan, et al., 2019) This highlights the pressing need for more advanced and specialized models. Furthermore, it is a challenge to apply deep learning-based systems in real-time situations where there are diverse fire and environmental conditions. This highlights the importance of enhancing sensitivity and performance in complex real-world scenarios. Moreover, utilizing transfer learning in fire detection systems that are based on deep learning methods has been recognized as an area with potential for progress. It is crucial to make use of the knowledge acquired from existing data to enhance detection accuracy and efficiency. Other research findings highlight that employing transfer learning in the fire detection task leads to higher accuracy compared to using the base model alone. However, the performance of the identified transfer learning strategy was not compared to other state-of-the-art fire detection techniques. In addition, they do not specifically address potential external factors that could affect the performance of the proposed model in real-world monitoring Platforms. (Pednekar et al., 2023)

To overcome these challenges, we introduce a fire detection system that utilizes transfer learning to enhance both accuracy and robustness. Transfer learning involves using pre-trained models that have already learned to extract features from images or videos to solve a new problem. By fine-tuning these pre-trained models on data from uncertain surveillance environments, we develop fire detection systems that can effectively find and locate fires under these challenging conditions.

1.4 Research Questions

This study investigates the following research questions:

RQ1.How do pre-trained CNN architectures perform when applied to fire detection in surveillance environments using transfer learning?

RQ2 How can hyperparameter optimization be used to improve the performance of pre-trained models for fire detection in surveillance environments?

1.5 Objectives of the Study

1.5.1 General Objective of the Study

The general objective of this study is to develop fire detection systems that can detect the existence of fires in uncertain surveillance environments using transfer learning.

1.5.2 Specific Objectives of the Study

To fulfill the study's overall goal, we set up the following specific objectives:

- To analyze the ideas and related study of transfer learning approaches about fire detection systems and identify the gaps.
- To prepare and augment the dataset to enhance its diversity and robustness by introducing the data augmentation technique.
- To identify the suitable pre-trained CNN model for developing fire detection systems for uncertain surveillance environments.
- To fine-tune the pre-trained model parameters for optimal performance on the target dataset.
- To train the model with the prepared datasets and assess its performance using a test dataset and evaluation metrics.

1.6 Significance of the Study

Fire detection plays a crucial role in safeguarding lives and property in various locations including forests, industrial facilities, and residential buildings. This study provides an opportunity for a future with faster, more accurate, and reliable fire detection in a variety of contexts by overcoming the drawbacks of traditional methods. For many various individuals and communities, this can result in significant improvements in economic well-being, environmental protection, and public safety. The key role of this study is overcoming the difficulties associated with the development of a fire detection system that can operate effectively in uncertain environments.

1.7 Scope and Limitation of the Study

1.7.1 Scope of the Study

This research examines the integration of transfer learning with pre-trained convolutional neural networks (CNNs) to develop a robust fire detection system for uncertain surveillance

environments. Four pre-trained CNN architectures, VGG19, MobileNetV2, ResNet50, and InceptionV3 are compared and evaluated using metrics such as accuracy, precision, recall, and F1-score to identify the model that performs best for fire detection classification in these uncertain conditions.

1.7.2 Limitations of the Study

This research explored the efficacy of transfer learning in detecting fires within uncertain surveillance environments. While the findings provide valuable insights, several limitations should be noted. The system lacks functionalities for subsequent actions upon fire detection, such as activating fire suppression systems or sending notifications. Additionally, the proposed system may fail to detect fires if they are not prominently visible within the captured image frames.

1.8 Contribution of the Study

This study contributes to the development of more accurate fire detection systems in uncertain surveillance environments by comparing the performance of four modified pre-trained models VGG19, MobileNetV2, ResNet50, and InceptionV3, and identifying the optimal model for this challenging task. Additionally, we created a comprehensive dataset that reflects the complexities of real-world surveillance by drawing from various benchmarked research papers and online sources, selected according to environmental conditions criteria.

1.9 Organization of the Study

The organization of the research refers to the overall structure and logical flow of the research. This Study is divided into six chapters, each contributing to a comprehensive investigation of fire detection using transfer learning in uncertain surveillance environments.

Chapter One: This chapter sets up the research context by presenting the background, motivation, and significance of the study. It outlines the research problem statement, objectives, research questions, and scope, laying the groundwork for the investigation.

Chapter Two: This chapter explores existing research related to fire detection using transfer learning. It critically analyzes past studies, identifies research gaps, and establishes the theoretical foundation for this research.

Chapter Three: This chapter outlines the research methodology employed in the study. It details the data collection and preparation methods, the selection and training procedures for the pre-trained CNN models, and the evaluation metrics used to assess their performance.

Chapter Four: This section focuses on the design of the fire detection experiments and the selection of the optimal pre-trained CNN model for implementation. It details the specific data preprocessing techniques chosen for the selected model and justifies the chosen pre-trained architecture.

Chapter Five: This section provides a comprehensive overview of the implementation process for the fire detection system. It delineates the specific steps undertaken to configure the selected pre-trained model, encompassing the selection and fine-tuning of crucial hyperparameters. Furthermore, the chapter delves into the setup of the working environment, detailing any essential software or hardware configurations required for training the model. To elucidate the key aspects of the implementation process, relevant code snippets will be included throughout the chapter.

Chapter Six: This chapter exposes the key findings of the research, presenting the results of the preprocessing and data augmentation techniques, along with the performance of each pre-trained model on the fire detection task, including the chosen evaluation metrics and their outcomes. Following this, the discussion section delves deeper into the meaning of these results. It explores how the findings relate to the research questions initially posed and compares them to existing research in the field.

The last section summarizes the key findings of the research, reiterates the significance of the conclusions, and offers recommendations for upcoming study works.

CHAPTER TWO

2 LITERATURE REVIEW AND RELATED WORK

2.1 Conceptual Background

In the realm of artificial intelligence, computer vision empowers computers and systems to analyze visual data like digital images, videos, and other inputs, uncovering valuable insights.(Wiley & Lucas, 2018) It can also conduct or make recommendations grounded on that information. It is a multi-disciplinary field that deals with how computers can be made to gain a high-level understanding of digital images or videos. Machine learning is a type of artificial intelligence that allows computers to learn from data and accomplish activities that would normally need human knowledge(Mendonça et al., 2024).

The development of computer vision applications is aided by machine learning; yet, processing complex data, like images and videos, is a challenging task for machine learning algorithms. For this complex data analysis, we turned to a powerful subset of machine learning called deep learning. Deep learning, specifically through the use of CNNs, has transformed the field of computer vision, enabling us to extract meaningful insights from challenging datasets (Sarker, 2021). The ability of neural networks to learn and extract meaningful features from images has significantly improved object detection, image classification, and semantic segmentation. The key to unlocking the future of computer vision lies in the ongoing evolution of deep learning architectures. Deep learning allows computer models to learn and perform complex tasks directly from visual data, including images, videos, and even other sensory inputs(Talaei Khoei et al., 2023). These models are capable of achieving innovative accuracy, sometimes surpassing even human capabilities in specific areas. Training these models involves feeding them vast amounts of labeled data and leveraging neural network architectures with intricate, multi-layered structures. Many researchers have proposed fire detection methods based on traditional and computer vision methods(Hassan & Audu, 2022). Traditional sensor-based systems depend on fixed rules and threshold values to identify potential fires. These systems analyze specific factors such as smoke concentration, temperature changes, and heat patterns to trigger an alarm. However, traditional-based systems have limitations such as transport delays, conduction delays, and limited detection ranges. Traditional sensor-based fire detection methods use several types of sensors to detect fires.(F. Khan et al., 2022b) Those are:

Heat Sensors: - Heat Sensors use detectors sensitive to temperature changes. They initiate an alarm when the temperature rises above a predetermined level.

Smoke Sensors: - These sensors use light beams and photoelectric sensors to detect smoke particles that scatter the light, resulting in an alarm. And use radioactive materials to ionize air molecules, which then conduct electricity. Smoke disturbs this flow, setting off the alarm.

Flame Sensors: - These sensors use ultraviolet (UV) or infrared sensors to detect the flashing light patterns of fires, which are faster than smoke or heat sensors**.**

Gas Sensors: - This sensor detects invisible but dangerous carbon monoxide, a byproduct of incomplete combustion, and prevents carbon monoxide poisoning. Detecting specific combustible gases, such as propane or methane, can assist prevent explosions and fires.

Computer vision-based methods use surveillance cameras to record video or images in real-time, and a computer system processes the images to extract fire image features(Xie et al., 2021). These features are then applied to a suitable machine learning or deep learning algorithm to detect fire and trigger the alarm. Faster response times, precise detection, and better descriptive data are some benefits of computer vision-based fire detection techniques that can also address some of the drawbacks of traditional sensor-based techniques. There are two ways of applying computer vision for fire detection(O'Mahony et al., 2020). Those are traditional hand engineering feature-based and deep learning-based methods. Traditional Hand-Engineered Feature-Based Techniques: These techniques depend on manually creating features deemed crucial for identifying fire in images or videos. Features like edge detection (to locate flames), texture analysis (to inform fire from smoke), and color space analysis (to search for red, orange, and yellow hues) are a few examples. Then, to determine whether or not an image or video contains fire, these attributes are combined with machine learning techniques like Support Vector Machines (SVM) or Random Forests. Deep learning-based methods: This approach uses deep neural networks, namely CNNs, to automatically learn features from the data. Pre-trained models, such as VGG16 or ResNet, are commonly employed for transfer learning, with the pre-trained weights fine-tuned using fire detection data. These algorithms may learn complicated representations from images/videos, perhaps resulting in more robust fire detection than hand-crafted(Jin et al., 2023).

When comparing handcraft features based within deep learning-based methods the handcraft feature-based have limitations such as low accuracy, high complexity, and poor adaptability to different conditions.

Surveillance cameras, often called security cameras or CCTV (closed-circuit television) cameras, are video cameras that monitor and record activities in a given area for a variety of purposes, including security, safety, and surveillance(Welsh et al., 2020).

Uncertain surveillance encompasses situations where the information acquired by surveillance systems is unclear, incomplete, or untrustworthy.

Table 2. 1 The Contrast between traditional and vision-based fire detection systems

Parameter	**Traditional Sensor based system**	**Computer vision based system**
Detection method	Physical sensors (smoke, heat, flame, gas)	Machine learning algorithms
Data source	Sensor data	Images, videos, sensor data
Accuracy	Moderately accurate	Potentially more accurate
False alarm rate	High Prone to false alarms from dust, steam, cooking smoke	Less prone to false alarms
Detection range	Limited area	Wider area
Response time	Slower response	Faster response time
Cost of implementation	Relatively inexpensive	More expensive
Complexity	Simple and easy to use	Requires complex infrastructure, data collection, and training
Maintenance cost	High	Low

2.2 Deep Learning Models for Fire Detection

VGG19 is a powerful convolutional neural network (CNN) developed by researchers at Oxford University(Bansal et al., 2023). Unlike earlier CNNs that relied on large filters, VGG19 uses a unique approach. It uses a series of small, stacked convolutional filters that progressively extract features from images. These filters, arranged in increasing levels of complexity, analyze the image at various levels of detail. The first filters might detect basic edges and shapes, while deeper layers learn more intricate combinations of these features, leading to the identification of complex objects or patterns. VGG19 architecture has proven highly effective for computer vision tasks. One key strength of VGG19 is its ability to recognize fundamental visual concepts like edges, textures, and colors by training on a large dataset like ImageNet(Das et al., 2023). This knowledge establishes a strong base for applying transfer learning to broaden the scope of computer vision applications.

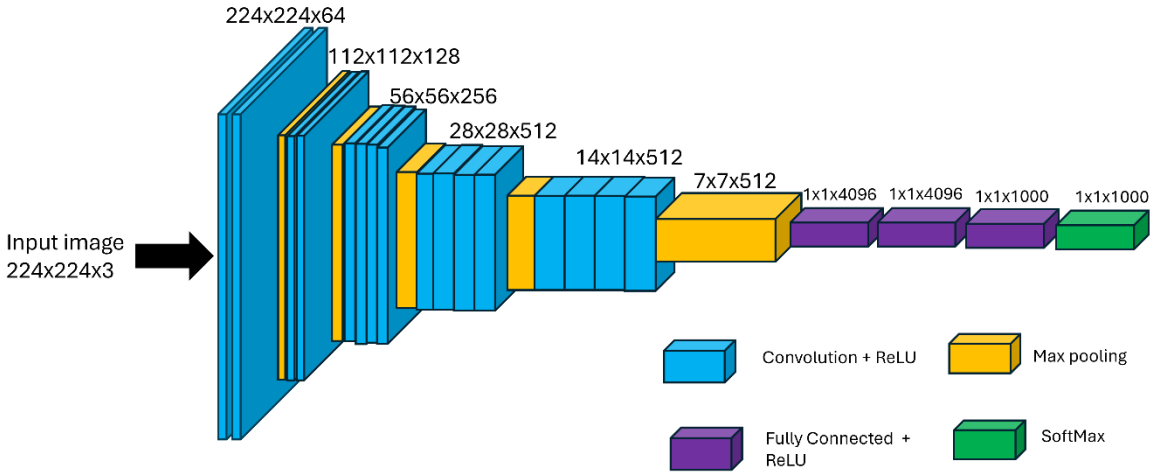


Figure 2. 1The pre-trained architecture of the VGG 19 model.

Adapted from(Ullah et al., 2022)

MobileNetV2 is an efficient convolutional neural network (CNN) specifically perfected for mobile and embedded devices(Dong et al., 2020). MobilenetV2 is a machine learning model that balances performance and the resources required to run. some key features make it unique-:

Lightweight Design: MobilenetV2 uses a technique called depth-wise separable- convolutions. This breaks down standard convolutions into two steps **depth wise- convolution** (which extracts features for each channel) and **pointwise convolution** (which combines features across channels). This approach significantly reduces the number of computations and parameters required compared to traditional convolutions. This makes it a perfect choice for applications on devices with limited resources, such as smartphones and embedded systems.

Efficient Bottlenecks: MobileNetV2 incorporates "linear bottlenecks" within its convolutional blocks. These bottlenecks use a low-dimensional linear layer between the depthwise and pointwise convolutions. This reduces the number of channels while still preserving valuable information, further enhancing the model's efficiency(Sandler et al., 2018).

Residual Connections: Like other modern convolutional neural networks (CNNs), MobilenetV2 uses residual connections, which help improve the model's performance. These connections allow the network to learn from the gradients of earlier layers more effectively, improving training speed and accuracy.

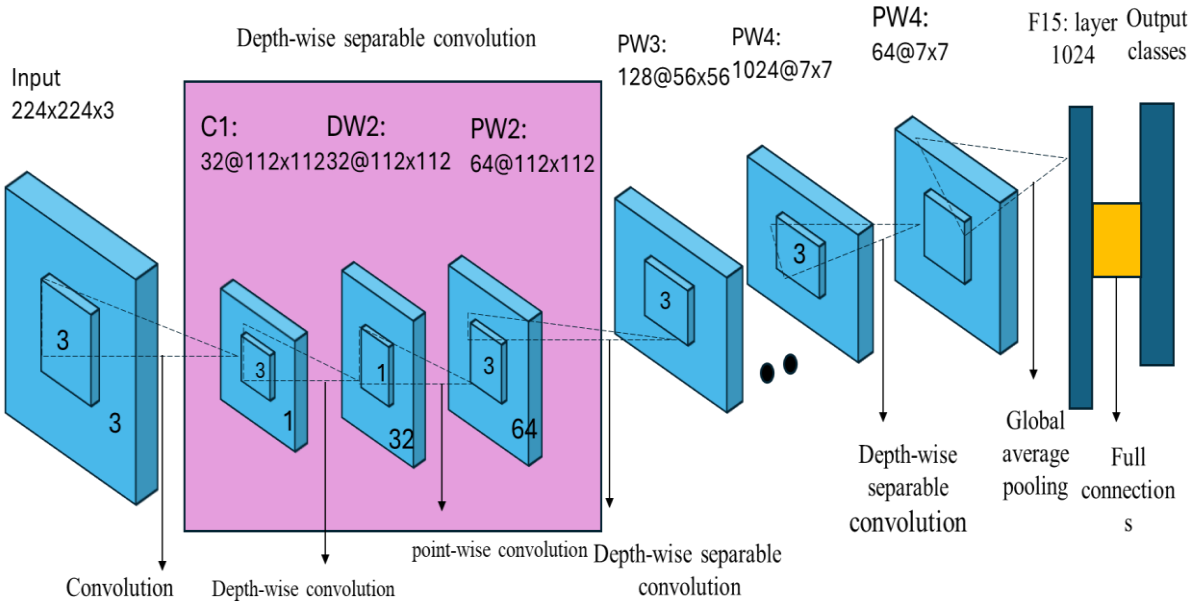


Figure 2. 2The pre-trained architecture of the MobileNet V2 model

Adapted from (M. U. Khan et al., 2022)

ResNet-50, a deep residual network architecture, has garnered acclaim for its ability to achieve state-of-the-art performance in image categorization tasks. ResNet-50 belongs to a class of CNNs, known as Residual Networks, which overcome the vanishing gradient problem(Mukti & Biswas, 2019). This problem hinders training in very deep networks by causing the gradients to get too small or too big when trying to backpropagate information through the deep layers during training. Here is why ResNet-50 overcomes it: To address the vanishing gradient problem that can hinder training in deep neural networks, ResNet incorporates a clever technique called residual connections. These connections act like shortcuts, skipping a few intermediate layers and directly adding a later layer's output to an earlier layer's output. This direct addition allows gradients to flow more easily throughout the network during backpropagation, facilitating the training of deeper models and improving their overall performance.

Bottleneck blocks: Because ResNet-50 is a block architecture, each block is a powerful feature extraction module with the name ‘bottleneck block’ that is computationally cheap(Reddy & Raju, 2023). The upper figure in the diagram illustrates a bottleneck block following a 3 x 3 convolution with 64 feature maps. A 1 x 1 convolution reduces the number of channels from 64 to 256. And, similarly, another 1 x 1 convolution is to recover the number of feature maps to 512. The aggressiveness of the ‘bottleneck block’ improves computation efficiency, by reducing the number of channels, without a decrease in accuracy.

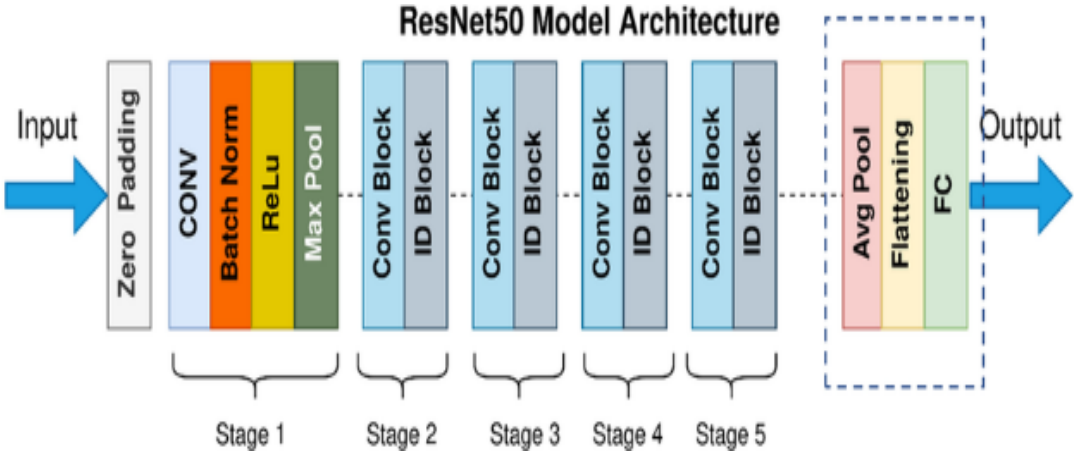


Figure 2. 3 The pre-trained architecture of the ResNet 50 model

(Source=Adapted from (Reddy & Raju, 2023))

Inception-v3, a deep learning model renowned for its image classification prowess, provides a powerful foundation for constructing a fire detection system using transfer learning. Transfer learning allows us to leverage the knowledge a pre-trained model has already acquired from a massive dataset (like ImageNet) and adapt it to a new, related task(Mahalle et al., 2024).

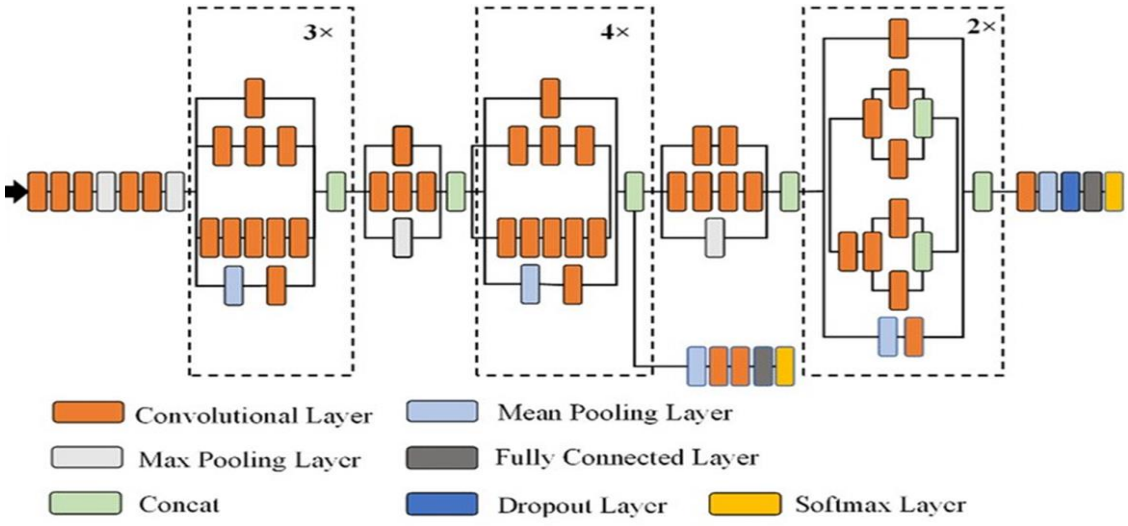


Figure 2. 4 The pre-trained architecture of the InceptionV3 model.

(Source = Adapted from(Bania, 2023))

2.3 Overview of Transfer Learning

Traditional machine-learning approaches have demonstrably achieved success across various real-world applications, but inherent limitations persist(Suyal et al., 2021). The ideal scenario hinges on an abundance of labeled training data that closely mirrors the test data distribution. However, acquiring such extensive labeled datasets can be impractical, often incurring significant costs and time investments.

One strategy to mitigate this data scarcity is semi-supervised learning, which leverages a combination of limited labeled data and a larger pool of unlabeled data(Yang et al., 2023). While this approach aims to enhance learning accuracy through the inclusion of unlabeled data, gathering sufficient unlabeled examples can be equally challenging, ultimately hindering the effectiveness of conventionally trained models.

Transfer learning emerges as a promising alternative machine learning technique that tackles these data-related hurdles by capitalizing on knowledge transfer across domains(Kim et al.,

2022). In essence, transfer learning involves leveraging a pre-trained model developed for one task as a foundation for a model tackling a various but related task. This proves particularly advantageous when the new task exhibits significant similarities to the original task, or when data constraints limit the training process for the new task. By utilizing the learned features from the pre-trained model, the new model can achieve faster and more efficient learning on the new task. This can also help prevent overfitting, as the model already possesses a base of generalizable features applicable to the new task(Rafiq & Albert, 2022).

In computer vision and natural language processing, where tasks often require substantial computational resources, transfer learning shines as a particularly attractive approach.

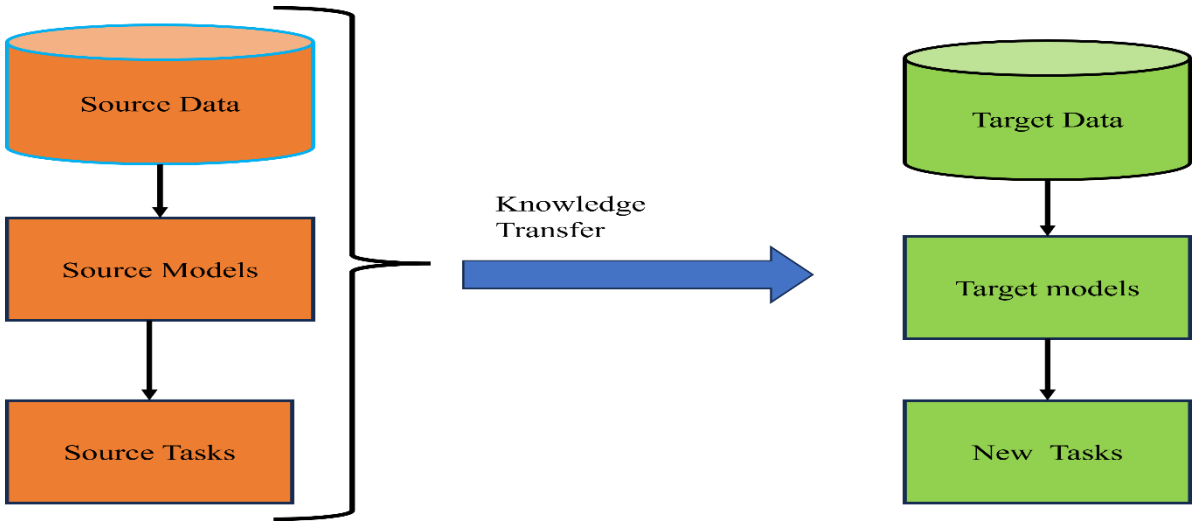


Figure 2. 5 Basic layout of the standard transfer learning process.

Simple illustrations of transfer learning are provided in Figure 2.6. (Zhuang et al., 2021) Inspired by how humans can apply knowledge from one area to another, transfer learning aims to

leverage knowledge gained in a related field (source domain) to improve learning performance or reduce the amount of labeled data required in a new target domain.

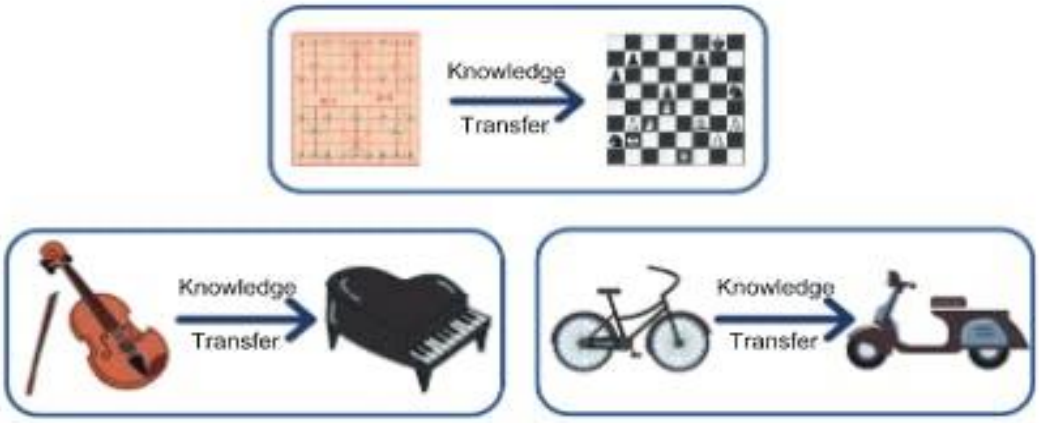


Figure 2. 6 Intuitive Examples of Transfer Learning.

Adapted from ((Zhuang et al., 2021)

2.4 Related Work

(Bari et al., 2021) This study assessed transfer learning's effectiveness in fire detection with MobileNetV2 and InceptionV3. Their study compared training a model from scratch with applying transfer learning on a similar dataset. The findings indicated that transfer learning attained superior performance, especially when dealing with limited datasets. Fine-tuning further improved the accuracy of both MobileNetV2 and InceptionV3 on the validation data. However, the research did not provide a comprehensive analysis of false positive and false negative rates, which are crucial metrics for evaluating the reliability and real-world applicability of fire detection systems.

(Mahmoud et al., 2022) developed a CNN-based fire detection system that prioritized computational efficiency, making it suitable for real-time applications. By employing transfer learning techniques, the proposed model achieved faster training and classification times compared to existing models in the literature. However, the study does not specifically address potential external factors that could affect the performance of the proposed model in real-world monitoring systems.

(Majid et al., 2022) investigated a different fire detection context that leverages transfer learning with state-of-the-art convolutional neural networks (CNNs). They compared the performance of VGG16, ResNet50, GoogleNet, and EfficientNetB0 architectures, all pre-trained on image datasets. Their findings revealed that EfficientNetB0 achieved the highest accuracy (95.40%) on a practical scenarios fire image dataset. Additionally, the system showed a high recall of 97.61%, indicating its effectiveness in minimizing false negatives (missed fire detections). These results suggest that EfficientNetB0 is a promising candidate for fire detection applications.

Pednekar and colleagues explored leveraging transfer learning with pre-trained convolutional neural networks like InceptionV3, MobileNet, VGG19, and VGG16 to enhance the accuracy of fire detection classification. The research findings highlight that employing transfer learning in the fire detection task leads to higher accuracy compared to using the base model alone. However, the performance of the identified transfer learning strategy was not compared to other state-of-the-art fire detection techniques. (Pednekar et al., 2023)

In their work, (Dogan et al., 2022) they have explored a deep learning method for fire detection using residual networks (ResNets). For this task, they investigated the performance of four pre-trained ResNet architectures (ResNet-18, ResNet-50, ResNet-101, and InceptionResNetV2). Notably, the study used the deep features extracted from the final layers of these models, pre-trained on the ImageNet dataset, to identify fire patterns. The authors further enhanced the approach by developing ensemble models that combined these deep features with Support Vector Machines (SVMs) and k-fold cross-validation. This strategy achieved a high classification accuracy of 96.91%, showing the capability of deep learning for fire detection.

In their study(Dua et al., 2020) explored the use of various transfer learning and deep learning techniques for fire detection in images, focusing on a specific task like smoke or flame detection. Their investigation compared the performance of different Convolutional Neural Network (CNN) architectures, including VGG and MobileNet. The outcomes presented that deeper CNN architectures succeeded in greater accuracy associated to shallower CNN models. While MobileNet offered the benefit of a smaller footprint and faster execution times compared to VGGNet, their accuracy levels were comparable (around 95%). However, the paper

acknowledges the potential for overfitting during training but does not explore details regarding mitigation strategies.

Li and Zhao investigated the performance of two convolutional neural network (CNN) architectures, AlexNet and GoogleNet, for fire detection in image data. Their findings revealed that both models achieved high accuracy in detecting fires. However, GoogleNet exhibited a slight edge over AlexNet, achieving an accuracy of 94.43% compared to 94.39%.(Li & Zhao, 2020)

(Sharma et al., 2017) investigated the fire detection capabilities of ResNet-50 and VGG16, two CNN architectures. Their research highlighted the limitations of basic CNNs in handling this task. To address this, they incorporated fully connected layers into both ResNet-50 and VGG16 architectures. While this approach increased training time, it led to significant improvements in accuracy. After adding fully connected layers, ResNet-50 achieved an accuracy of 92.15%, while VGG16 reached 91.18% accuracy for fire detection.

(Frizzi et al., 2016) investigated the use of a CNN for fire and smoke detection in image datasets. Their approach employed a simple in sequence CNN architecture similar to LeNet-5, designed to classify images into fire, smoke, or non-fire/smoke categories. The researchers reported a high testing accuracy of 95% with a minimal false positive rate.

(Muhammad, Khan, et al., 2019) proposed a fire detection system designed for mobile and embedded vision applications in uncertain Internet-of-Things (IoT) environments. Their system uses lightweight deep neural networks, specifically MobileNetV2, to achieve computational efficiency. This approach aims to balance accuracy, minimize false alarms, and provide a suitable solution for resource-constrained devices. However, the evaluation relied on benchmark fire image datasets, which might not fully characterize the difficulties of real-world uncertain surveillance scenarios. This highlights the need for more diverse and realistic datasets to accurately assess fire detection system performance under unpredictable conditions.

Table 2. 2 Summary of Related Work

Author(s) and Year	**Methods**	**Datasets**	**Contribution**	**Research gap**
Muhammad et al., (2019)	Transfer learning with MobileNetV2	Create two benchmark datasets	computationally inexpensive. By using Mobilenet models	accuracy is not sufficient.
Dua et al., (2020)	Transfer learning with VGG19 and Mobilenet.	They Use 5515 images downloaded from various resources on the internet. With the categories of fire and non-fire.	Addressing the issues of imbalanced datasets and real-world application. To improve the fire detection.	Computational Expense Imbalanced Datasets accuracy is not sufficient.
Bari et al., (2021)	Transfer learning with MobileNetV2 and InceptionV3	Foggia's dataset and the downloaded fire videos and images have diverse environments from the web. And uses the BoWFire public	Utilizes transfer learning to address the issue of limited training data for fire detection.	Full analysis of false positive and false negative rates,

		dataset to evaluate the models.		The result of Precision, Recall, and F1 measures are not enough
Mahmoud et al. (2022)	Transfer learning with AlexNet.	uses two benchmark datasets Dataset1 contains 370 video Dataset2 contains 600 images.	The study improves accuracy and reduces false alarm rates.	The study does not specifically address potential external factors that could affect the performance of the proposed model in real-world monitoring systems.
Pednekar et al., (2023)	Transfer learning withVGG16, VGG19, MobileNetv2 and InceptionV3.	They Use 850 images collected from various online sources.	The study explores the effectiveness of using pre-trained CNN models and transfer learning.	Tested on small datasets. Evaluate the model using Accuracy only

CHAPTER THREE

3. RESEARCH METHODOLOGY

3.1 Overview of the Methodology

This chapter describes how the study method was conducted, including the data collection procedures, preprocessing approaches, hardware tools, software tools, and evaluation measures utilized to complete the study effectively.

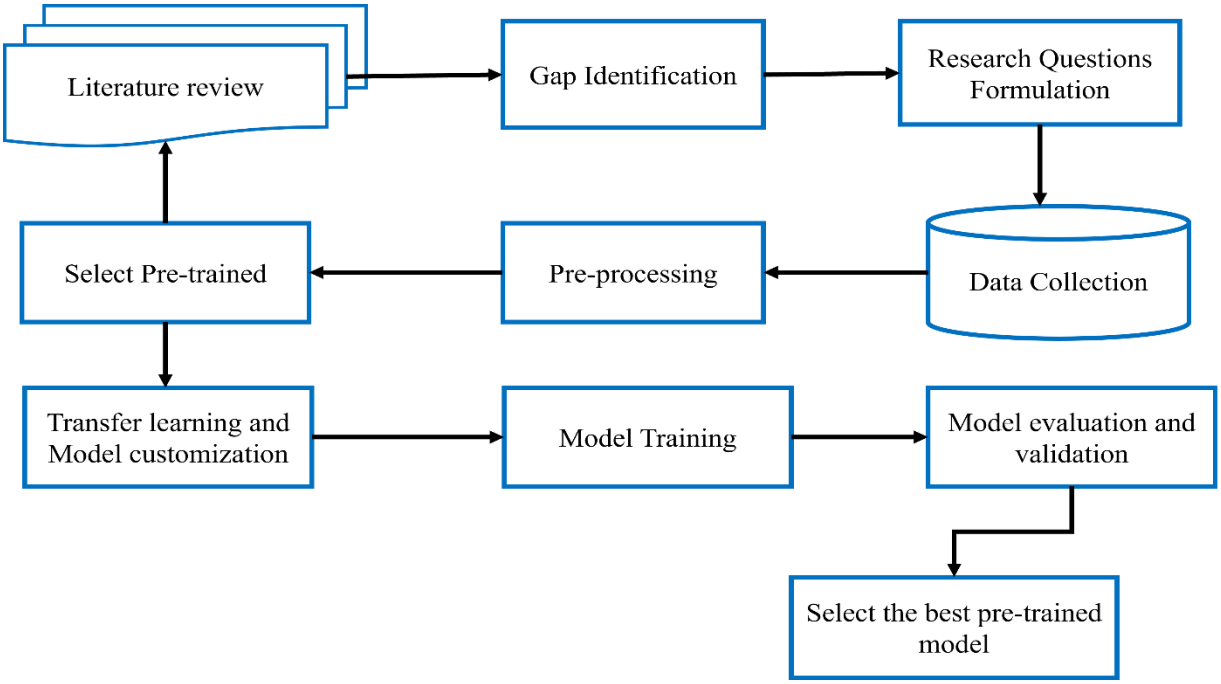


Figure 3. 1 Overall Process of the Study.

3.2. Datasets

The dataset for the study is collected from different benchmarked research papers available to the public. These are the fire and Smoke Detection (DFS) dataset (Wu et al., 2023), Dataset(Foggia et al., 2015) and Dataset(Chino et al., 2015) with two classes fire and non-fire.

Guide for Selecting Data for Fire Detection

This guide outlines the process for selecting data to train a fire detection system. We identified the specific environmental uncertainties expected at our deployment location, which include smoke, lighting, atmospheric conditions, and background clutter. Thick smoke can obscure flames and hinder sensors' ability to detect fire, with smoke density, color, and composition all

impacting detection. Poor lighting conditions, such as shadows, harsh sunlight, or flickering lights, can make it difficult to distinguish fire from reflections or normal lighting changes. Atmospheric conditions like fog, rain, dust, and snow can attenuate light and heat signals, impacting sensor performance. Additionally, objects in the scene, such as furniture, machinery, or normal variations in background textures, can be misinterpreted as fire by the system.

Data Selection from Benchmarked Research Papers Public Datasets Criteria:

Location: Consider the specific deployment location for our fire detection system. This helps the system learn the visual characteristics of fire in that specific context.

Size: we include images of varying-sized fires, from small, controlled burns to large wildfires. This improves the system's ability to detect fires at different stages.

Distance of Capture: We select images where the fire occupies different portions of the frame. include close-up images of flames as well as distant views where the fire might be a smaller element within the scene. This helps the system generalize fire detection across various distances.

Environmental Conditions: This is critical for uncertain environments. We Look for datasets containing fire and non-fire images captured under various conditions that might challenge fire detection.

We collected images containing elements that can be confused with fire, specifically addressing various uncertainties from benchmarked research papers, datasets, and online sources. These images include those depicting weather conditions such as smoke, snow, and fog; fire-like objects such as bonfires, fireworks, and welding sparks; red-colored objects like traffic lights and brake lights; and fire-like scenarios such as sunsets and reflections on water.By following these criteria, we create a training dataset that reflects the complexities of real-world environments.

Table 3. 1 Details of the dataset

Dataset	Fire	Non-fire	Total
Train	7425	7425	14850
Test	825	825	1650
Total	8250	8250	16500

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 3. 2 (a)-(d) Dataset having fire images, (e)-(h) Dataset containing non-fire images.

3.3 Pre-processing Techniques

Image preprocessing is a vital first step when preparing images for computer vision and machine learning. It uses various techniques to improve image quality and make them more suitable for analysis. This process helps enhance the images, making them clearer and more useful for further processing and modeling. By addressing common issues like noise, distortion, and inconsistencies, image preprocessing lays the groundwork for more effective computer vision and machine learning applications(Kumar et al., 2024).

3.3.1 Resize Image

Deep learning models are difficult and need significant computational resources. Most pre-trained deep learning models used for transfer learning have a specific input image size. Resizing our fire images to meet this desired size guarantees the model can process them correctly. By resizing all images in our fire dataset to a uniform scale, we create consistency for

the model. This allows the model to focus on learning the fire aspects themselves, rather than being distracted by variations in image size. Two points to be considered while scaling images for transfer learning: are target model input size and resizing methods. During training, pre-trained models are designed to work with specific input sizes, which are crucial for their internal operations and feature- extraction. Common sizes for image models include 150x150, 224x224, 299x299, and even larger dimensions like 448x448. However, certain models can adjust to different input sizes through dynamic resizing or alternative architectures.

3.3.2 Normalization

Normalization techniques are used to adjust the pixel intensities of an image to a specific range, commonly between 0 and 1 or -1 and 1. This helps to standardize the data and improve the training process for machine learning models. Standardizing the data ensures it is in a consistent format, which is essential for machine learning models to train effectively. There are two common techniques used to normalize data for training neural networks. Image normalization is calculated as follows:

$$Normalized\_Value = \frac{(\text{Original\_Value} - \text{Mean})}{\text{Standard\_Deviation}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots eq.1$$

Where, mean=0.5 and standard deviation = 0.5, the original value in their real range

So, the Normalized value was in the range of (-1,1)

Original value: The value of a particular pixel in the image (usually between 0 and 255 for RGB images).

Mean The average value for that feature across the entire dataset.

Standard Deviation: The standard deviation of that feature across the entire dataset.

Normalized Value: The value after applying standardization.

3.3.3 Data Augmentation

Data augmentation is a powerful technique in machine learning, particularly deep learning, that artificially boosts the size and variety of a training dataset. (Mikolajczyk & Grochowski, 2018). This is done by creating new data points from existing data through various transformations. Image data augmentation approaches are all about producing new versions of our current images

to deceive our machine-learning model into thinking it has a much bigger dataset(Xu et al., 2023). This helps the model perform better on unseen data by preventing it from just remembering the characteristics of the original images, and it will help prevent overfitting and improve the model's generalization capabilities(Mumuni & Mumuni, 2022). The most common image data augmentation strategies include random cropping, rotation, zooming, random flipping, brightness/contrast shifts, and color range shifting.

Random Cropping: Cropping exposes the model to a wider variety of fire sizes and locations within an image. This trains the model to effectively detect fires even when they appear in smaller areas.

Rotations: Fires can appear at a variety of angles, so rotating images slightly may expand the ability of a model to detect fires regardless of direction.

Zooming: Zooming in on fire regions enables the model to focus on specific elements such as flame texture, color changes, and flicker patterns. This is especially useful for models that need to distinguish between various kinds of fires or burning materials.

Random Flips (Horizontal/Vertical): This technique generates mirrored versions of our original fire images, like flipping them left to right or top to bottom. This enhances the model's ability to find fires in various directions.

Brightness/Contrast Shifts: The brightness and contrast of the images are randomly manipulated. This helps the model become more resistant to variations in lighting conditions, making it perform better in real-world circumstances where illumination might be unpredictable.

Color Range Shifting: Fire pixels often stay in specific color ranges (reds, oranges, yellows). Techniques like changing the color balance or saturation might emphasize certain fire-related colors, making them easier for the model to recognize.

3.4 Evaluation Metrics

Selecting the optimal model during training hinges on employing an appropriate evaluation metric. This metric guides us in identifying the model that generalizes best to unseen data. (M

& M.N, 2015). Evaluating deep learning models requires a toolbox of metrics, including accuracy, precision, recall, F1-score, and loss. These metrics provide a comprehensive picture of the model's performance. Four different parameters are employed to compute these measures. (Naidu et al., 2023)

True Positives (TP) are the correctly classified positive class records, indicating instances where the model accurately identifies the presence of a condition. True Negatives (TN) are the correctly classified negative class records, where the model accurately identifies the absence of a condition. False Positives (FP) are the incorrectly classified positive class records, representing instances where the model incorrectly identifies the presence of a condition. False Negatives (FN) are the incorrectly classified negative class records, where the model fails to identify the presence of a condition.

1) Accuracy: The proportion of accurately classified objects, representing a fundamental and extensively employed evaluation measure in classification tasks. The accuracy metric quantifies the relationship between accurate predictions and the overall occurrences analyzed.(Idrissi et al., 2021). Calculated as:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{eq.2}$$

2) Loss: This is a metric that demonstrations how far the method differs from the desired output. The method performs better when the loss is lower(Zhang & Sabuncu, 2018). A typical loss measure for classification issues is cross-entropy loss. It is calculated using the following formula:

$$\text{loss} = -\sum_i yi * \log(p\,i) \quad \text{eq.3}$$

Where y_i is the correct label's output and p_i is its predicted probability. Cross-entropy loss assesses the classifier's performance by punishing wrong predictions. Greater cross-entropy loss indicates more inaccurate predictions by the classifier.

3) Recall: The recall is defined as the proportion of accurately predicted positive instances to all positive instances(Cakir et al., 2019). Recall, also known as true positive rate or sensitivity, measures how well the model finds all actual fire instances. High recall indicates the model

rarely misses real fires. It's crucial to consider recall alongside metrics like precision and accuracy for a complete picture of the model's effectiveness.

$$\text{Recall} = \frac{TP}{TP+FN} \quad \text{.....eq.4}$$

N.B. Recall is used to calculate the percentage of positive patterns that are correctly classified.

3) Precision: Precision measures the accurately predicted positive patterns in a positive class among all predicted positive patterns.(Cakir et al., 2019). A high-precision model excels at identifying true positives, meaning it rarely mistakes negative instances for fire. Conversely, a low-precision model often misclassifies actual fires as something else. Here's how we calculate precision:

$$\text{Precision} = \frac{TP}{TP+FP} \quad \text{.....eq.5}$$

4. F1-score: In machine learning, the F1 score goes beyond just accuracy by considering both precision and recall assessing a model's effectiveness. It is the harmonic means of precision and recall, the F1 score assigns greater importance to lower values, rendering it particularly suitable for situations where both precision and recall carry significance. The F1-score calculation method is as follows:

$$\text{F1} - \text{Score} = \frac{2*(\text{Recall}*\text{Precision})}{(Recall+Precision)} \quad \text{..... eq.6}$$

3.5 Development Tools

This portion outlines the various tools and methodologies that be employed throughout the research and development phases of this study. These tools encompass Unified Modeling Language (UML) modeling software for system design, prototyping platforms for iterative development, and added resources to facilitate the research process. The upcoming portions will give a comprehensive overview of the specific tools selected for each stage of the fire detection system development process.

3.5.1. Design Tools

Tools for designing are instruments for the creation, interpretation, and presentation of design ideas. The suggested system will employ an Enterprise mastermind for design purposes. It is a

movable, yet broadly productive graphic design tool for constructing networks, flowchart diagrams, and other visual representations of data that appear professional.

3.5.2 Hardware Tools

The following is a list of the hardware components utilized during the implementation phase.

Table 3. 2 Hardware tools

Number	Component names	Description
1	Random Access Memory	To collaborate with the graphics card to speed up the training operation.
2	GPU	To enhance computation and speed up training time.
3	Hard disk	Used to store massive amounts of data.

3.5.3 Software Tools

This section explores various software and coding tools employed to implement research findings through coding. The following list details these tools along with their functionalities.

Table 3. 3 Software tools

S.no	applications	Tools description
1	Anaconda[2]	Anaconda is a free, open-source Python environment popular for data science and machine learning, including essential libraries and a Python interpreter.

[2] https://www.anaconda.com/

2	Jupyter Notebook[3]	A free, web-based platform that allows users to code and see visualizations in real time.
3	Keras[4]	An open-source library simplifying neural network creation in Python with TensorFlow.
4	Microsoft	Suitable for creating text documents, reports, and presentations.
5	Python language	A beginner-friendly yet powerful language ideal for building machine learning models.

3.6 Baseline Work

Our implemented model is evaluated against existing approaches that utilize transfer learning with pre-trained CNN architectures (InceptionV3, MobileNet, VGG19, VGG16) to enhance fire detection accuracy. This comparison aims to solidify the effectiveness of transfer learning in the fire detection domain.(Pednekar et al., 2023)

We chose (Pednekar et al., 2023) study as a baseline for this study because this work shows that transfer learning leads to higher accuracy compared to training models from scratch.

[3] https://jupyter.org/
[4] https://keras.io/

CHAPTER FOUR

4. PROPOSED ARCHITECTURE

4.1 Proposed Methods

In the proposed method, we implemented a transfer learning approach to develop a fire detection system in uncertain surveillance environments. First, a pre-trained CNN model was selected, and the fully connected layers of the model were removed. A new fully connected layer with two output nodes for 'fire' and 'no fire' classes was added and the weights in this layer were randomly initialized for each selected pre-trained model.

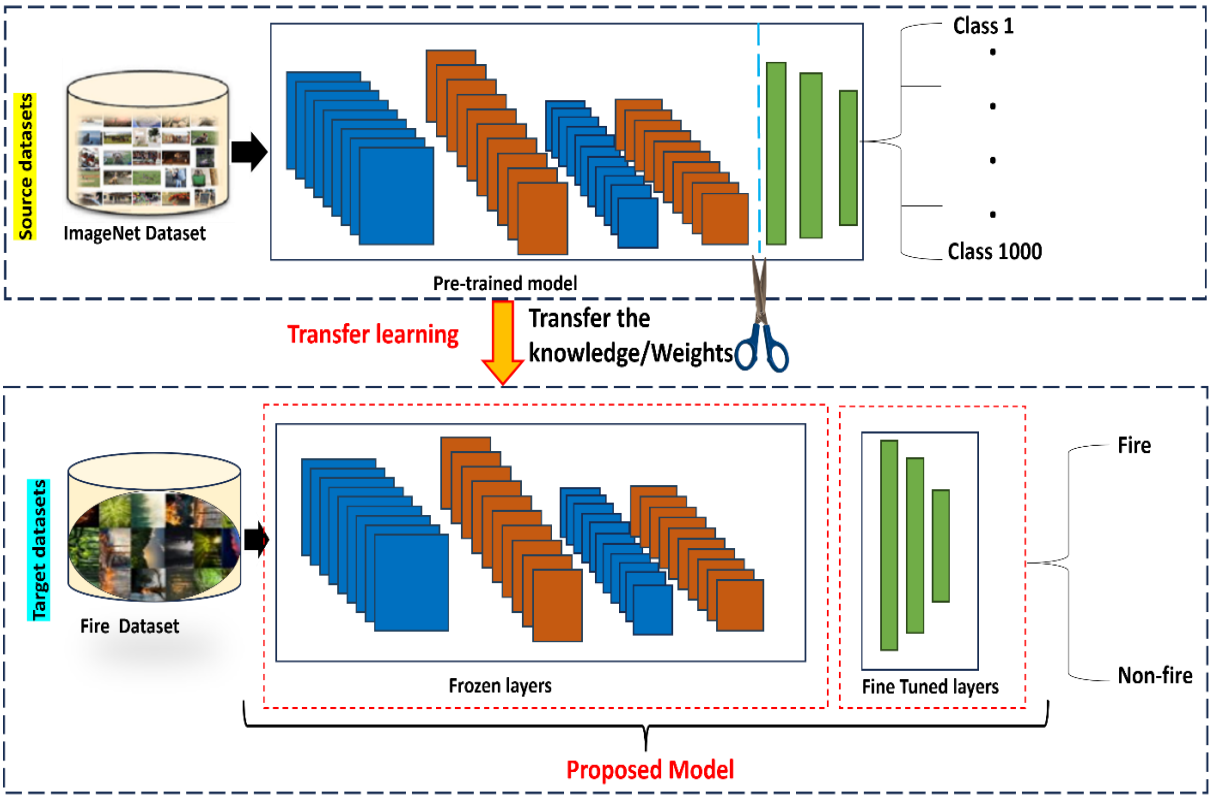


Figure 4. 1 Proposed Transfer Learning-based Framework for Fire Detection.

This research investigates the efficacy of various pre-trained CNN architectures for fire detection in uncertain surveillance environments. We conducted a comprehensive literature review of recent advancements in this field to select suitable deep-learning architectures for fire detection in uncertain environments. Additionally, we explored the pre-trained models available within the TensorFlow framework, and their performance on image recognition benchmarks. This comprehensive approach led us to select VGG19, ResNet50, and InceptionV3, known for

their high accuracy, and MobileNetV2, a lightweight model optimized for edge-device speed. These models are all pre-trained on massive image datasets (e.g., ImageNet) encompassing a wide variety of objects and scenes. During this training, they develop a robust ability to extract generic and powerful lower-level features from images. These features capture essential characteristics such as edges, shapes, and textures, which are crucial for various computer vision tasks, in our case fire detection(Shaha & Pawar, 2018). The ability to leverage these pre-trained features is particularly valuable in uncertain environments. Fire often manifests with specific visual patterns, such as flickering, irregular shapes, and distinct color variations. The pre-trained models' feature extraction capabilities can aid in identifying these patterns even accompanied by the variations present in uncertain surveillance situations (e.g., lighting changes, background clutter). A key advantage of pre-trained models is Transfer learning(Cody & Beling, 2023). This technique allows us to harness the knowledge learned by a model on a vast dataset and apply it to a new, specific task like fire detection in an uncertain environment. These pre-trained models can serve as a powerful foundation. Their learned features can be transferred and adapted to the specific fire detection problem(Waisberg et al., 2023). This approach offers several benefits:

Reduced Training Time: Training from scratch becomes much more difficult and resource-intensive with limited data, as deep learning models typically require vast amounts of data for optimal performance. Transfer learning from pre-trained models significantly reduces training time by utilizing the existing feature extraction capabilities. This allows for faster experimentation and development of the fire detection system.

Regularization: Transfer learning acts as a regularization, helping to prevent overfitting. Overfitting occurs when a model memorizes the training data too well and does not generalize to unseen examples. By leveraging the pre-trained model's knowledge as a starting point, the fire detection system can focus on learning the specific characteristics of fire in an uncertain environment, minimizing the risk of overfitting to the limited training data.

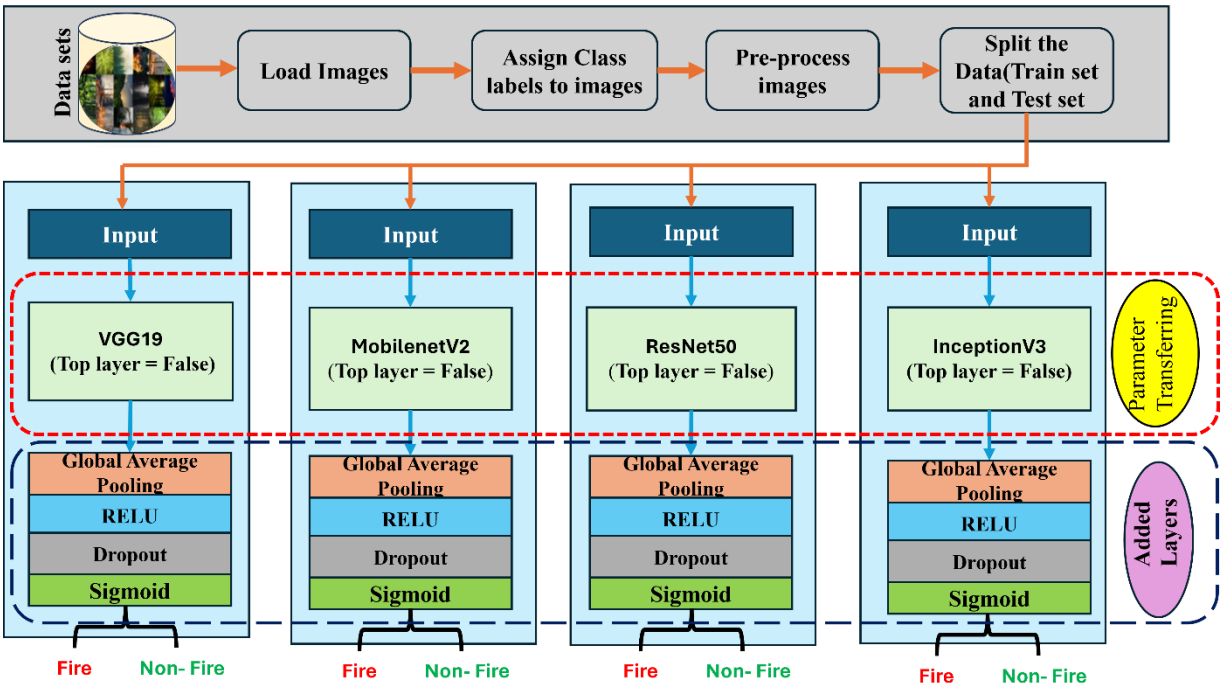


Figure 4. 2 Proposed Solution Architecture.

4.1.1 Transfer Learning With VGG-19 Model

By freezing the pre-trained layers of VGG19 and adding custom layers on top, we fine-tune the model to a specific problem, like fire image detection in our case, without starting the train from scratch. The first layer, GlobalAveragePooling2D, significantly reduces the complexity of the data by calculating the average value for each feature across the entire image. This dimensionality reduction improves efficiency for subsequent layers. Following this step, a fully connected layer with 512 neurons and a ReLU activation function further processes the data. Here, the features are transformed into a higher-level representation that's more suitable for distinguishing fire from non-fire images. The Rectified Linear Unit activation function is a critical component, introducing non-linearity. This allows the model to learn complex relationships between the extracted features. To prevent the model from overfitting during training, a dropout layer randomly drops out half of the neurons in the previous layer on each iteration. This forces the model to rely on broader features, ultimately improving its ability to generalize to unseen data. Finally, the model utilizes a single neuron with a sigmoid activation function in the output layer. Since this is a binary classification problem (fire vs. non-fire), the

output represents the probability of the image belonging to the fire class, with a value ranging from 0 (not fire) to 1 (highly likely fire).

4.1.2 Transfer Learning with Mobile Net V2 Model

We built custom layers on top of the pre-trained MobileNetV2 model to specialize it for fire image classification. We employ a GlobalAveragePooling2D layer. This layer significantly reduces the dimensionality of the data extracted by MobileNetV2, converting the feature maps into one vector with a lower number of dimensions. This compressed representation is more efficient for processing by the fully connected layers that follow, which require 1D inputs. Following the GlobalAveragePooling2D layer, a Dense layer with 512 neurons and a ReLU activation function takes over. This layer refines the features, focusing on learning higher-level patterns specific to fire classification, leveraging the foundation provided by the pre-trained features. In order to mitigate the risk of overfitting, a Dropout layer with a 0.5 rate is incorporated. During training, this layer randomly disables 50% of the neurons in this layer on each iteration. This forces the remaining neurons to learn more robust features and reduces reliance on any single data point. Finally, the model utilizes a single neuron with a sigmoid activation function in the output layer. Since this is a binary classification problem (fire vs. non-fire), the output represents the probability of the image fitting to the fire class, with a value ranging from 0 (not fire) to 1 (highly likely fire).

4.1.3 Transfer Learning with ResNet 50 Model

To take advantage of this pre-trained knowledge, we freeze the base layers (convolutional layers) of the model. This prevents them from being updated during training and ensures they keep the valuable generic features they've learned. Freezing these layers also helps reduce the overall training time(Hossain et al., 2022). We then build upon this base by adding a small set of custom layers on top layers. These custom layers handle learning features specific to identifying fire in our images. First utilizing a GlobalAveragePooling2D layer. This layer significantly reduces the dimensionality of the feature maps extracted by the pre-trained model by performing average pooling across the entire width and height. This compressed representation makes subsequent layers computationally more efficient. Following this, a densely connected layer with 512 neurons and ReLU (Rectified Linear Unit) activation is added. This layer transforms the pooled features into a higher-level representation which is more suitable for the binary classification task, potentially distinguishing fire from non-fire images.

The Rectified Linear Unit activation function is a critical component, introducing non-linearity. This allows the model to learn complex relationships between the extracted features. To prevent the model from overfitting during training, a dropout layer randomly drops out half of the neurons in the previous layer on each iteration. This forces the model to rely on a broader set of features, ultimately improving its ability to generalize to unseen data. Finally, the model utilizes a single neuron with a sigmoid activation function in the output layer. Since this is a binary classification problem (fire vs. non-fire), the output represents the probability of the image belonging to the fire class, with a value ranging from 0 (not fire) to 1 (highly likely fire).

4.1.4 Transfer Learning with Inception -V3 Model

On top of the pre-trained InceptionV3 model, we add several custom layers to improve its ability to distinguish fire from non-fire images: We start with implementing GlobalAveragePooling2D layers. This layer performs average pooling across the entire width and height of the feature maps extracted by the pre-trained model. This technique effectively reduces the dimensionality of the data, making it computationally more efficient for subsequent layers to process. Then A densely connected layer with 512 neurons and ReLU (Rectified Linear Unit) activation is added. This layer transforms the pooled features into a higher-level representation more suitable for the classification task. The Rectified Linear Unit activation function is a critical component, introducing non-linearity. This allows the model to learn complex relationships between the extracted features. To prevent the model from overfitting during training, a dropout layer randomly drops out half of the neurons in the previous layer on each iteration. This forces the model to rely on a broader set of features, ultimately improving its ability to generalize to unseen data. Finally, the model utilizes a single neuron with a sigmoid activation function in the output layer. Since this is a binary classification problem (fire vs. non-fire), the output represents the probability of the image belonging to the fire class, with a value ranging from 0 (not fire) to 1 (highly likely fire).

CHAPTER FIVE

5. IMPLEMENTATION OF THE PROPOSED MODEL

5.1 Chapter Overview

This section describes how we built the fire detection model. It details the chosen software, hardware (computing environment), and data augmentation techniques. We then leveraged transfer learning by loading a pre-trained model and fine-tuning it for fire detection.

5.2 Working Environment

This section outlines the hardware and software resources leveraged to develop our proposed model and achieve its desired functionality.

Desktop:

- Processor: Intel(R) Core (TM) i7-8700 CPU @ 3.20GHz 3.19 GHz.
- Installed RAM: 8.00 GB
- Operating system: Windows 11 Enterprise
- Graphics: Intel® UHD Graphics 630
- System type: 64-bit operating system, x64-based processor
- GPU: NVIDIA GeForce RTX 2070 Super with 8GB Dedicated Random access memory.

5.3 Environmental Setup

To set up the environment for our system, we utilized several tools. Anaconda, an open-source package manager, was employed to simplify the setup of Python and its scientific libraries. Additionally, Jupyter Notebook, a web-based application, was used to facilitate coding and real-time visualization. Python, a user-friendly and high-level programming language, served as the primary language for our system's implementation.

5.4 Data Augmentation Implementation

In transfer learning, data augmentation shines as a powerful tool, particularly for tasks with limited datasets. To compensate for the original dataset's size, we implemented various augmentation techniques. These techniques include randomly flipping images, adjusting brightness levels, translating images horizontally or vertically, zooming in or out on specific regions, and rotating images at various angles. By artificially expanding the training data with these variations, we enhance the model's ability to generalize and improve its performance on unseen fire detection scenarios.

5.5 Proposed Model Implementation

Several powerful libraries fuel the development of our fire detection model. TensorFlow is the core deep learning framework, offering tools for data processing, model definition, training, and deployment. Keras simplifies building and training neural networks on top of either framework. OpenCV offers real-time computer vision capabilities for image processing, feature extraction, and object detection – crucial aspects of fire detection. Scikit-image offers other functionalities for pre-processing image data. Visualization is handled by Matplotlib for exploring data and model results. Finally, NumPy and Pandas lay the foundation for numerical computations and data manipulation, potentially used for data exploration and pre-processing tasks.

Our fire detection model employs transfer learning by using the pre-trained model architecture. This pre-trained model, trained on the large ImageNet dataset, offers valuable knowledge about low-level visual features. To focus on feature extraction and avoid task-specific biases, we exclude the final classification layers of our pre-trained model (achieved through include top=False). The model additionally expects input images of size 224x224x3 (RGB format). This configuration allows our model to harness the pre-trained weights of the pre-trained model for efficient feature extraction, which will later be used for fire detection in later stages of the model.

To effectively leverage transfer learning, we employ a technique known as freezing the base model layers. This is achieved by iterating through each layer in the pre-trained model (base model) using a loop. Within the loop, setting the trainable attribute of each layer to False (achieved through the layer. trainable = False) freezes the weights (parameters) associated with that layer. Our fire detection model builds upon the pre-trained model for feature extraction. We add a classification head on top, starting with global average pooling to summarize spatial information into a feature vector. This vector is then fed into a densely connected layer with ReLU activation for learning complex relationships between features. Dropout regularization is applied to prevent overfitting. Finally, a densely connected layer with an activation function predicts the probability of fire or non-fire in the input image. This approach leverages the pre-trained knowledge of the base model while adapting it to the specific task of fire recognition.

This study employs Keras' functional API to assemble our fire detection model, defining the architecture by integrating a pre-trained model for feature extraction and using its input layer to

ensure compatible image formats. The output is assigned to the prediction variable, which represents the final layer of our custom classification head with Sigmoid activation, used to determine fire/non-fire probabilities. By Using the compile function, we prepare the fire detection model (model) for training. This function specifies the loss function, optimizer, and metrics used during training. In this case, 'binary_crossentropy' is likely chosen as the loss function because the fire detection task is assumed to be a binary classification problem (fire vs. no fire). The optimizer (represented by 'optimizer') guides the model's weight updates during training. Additionally, 'accuracy' is included as a metric to monitor the model's performance (percentage of correct predictions) on the training data. By compiling the model with these elements, we define how the model will learn and improve its fire detection capabilities on the provided fire image dataset.

To improve the training process, we incorporate two callback functions from Keras: Early stopping and model checkpoint. The early stopping callback monitors the model's performance on unseen validation data. If the model's performance on this data doesn't improve for training rounds (epochs) in a row, Early Stopping automatically halts training. This saves time and helps prevent the model from memorizing the training data too much, which can lead to poor performance on new data. To ensure efficient training and prevent overfitting, we employ the model checkpoint callback. This callback creates checkpoints of the model throughout training and only saves the model that achieves the highest validation accuracy. This strategy, incorporating early stopping and model checkpointing, allows us to select the model that generalizes best to unseen fire data, ultimately improving the fire detection system's performance.

We employed the fit function to train our fire detection model (model). This function takes the training data generator (train_generator) as input, along with the desired number of training epochs (epochs). The validation_data argument is set to the validation data generator (validation_generator), allowing the model to check its performance on unseen validation data during training. To enhance the training process, we include a list of callbacks (callbacks= [early_stopping, model_checkpoint_callback]). The early_stopping callback prevents overfitting, while the model_checkpoint_callback automatically saves the model with the highest validation accuracy. By incorporating these elements, we start the training process,

where the model learns to effectively distinguish between fire and non-fire images within the provided training data.

To understand how well our fire detection model is learning, we use a special tool called Matplotlib. This tool helps us create a visual representation of the model's accuracy as it trains over time (epochs). Imagine a graph with training accuracy on one line and validation accuracy on another. We use labels like "Train" and "Val" to distinguish them. By analyzing this plot, we can gain insights into the model's learning behavior, show potential overfitting issues, and evaluate its generalizability to unseen data.

We plot the model's training and validation losses over the training epochs. The title "Model loss" and labels on the axes ("Loss" for the y-axis and "Epoch" for the x-axis) provide context. The legend with labels "Train" and "Val" (positioned at the upper right corner with loc='upper right') differentiates the two lines. Additionally, plt. ylim (top=1.2, bottom=0) sets the y-axis limits to ensure all potential loss values are displayed (typically loss is non-negative). By analyzing this plot alongside the accuracy plot, we gain a more comprehensive understanding of the training process. Ideally, the loss curves should decrease over epochs, indicating the model is learning. A significant gap between training and validation loss could suggest overfitting. This visualization helps assess the model's ability to learn effectively and avoid overfitting during training.

We employ the evaluate function with a test data generator (test_generator) to evaluate our fire detection model's performance on unseen data. This generator was created using ImageDataGenerator for normalization (rescale=1. / 255) and prepares the test images by resizing them to the expected size (224x224 pixels) and setting the batch size. The class_mode='binary' argument signifies that we are treating the fire detection task as a binary classification problem (fire vs. no fire). The evaluate function returns test loss, accuracy (how well the model's predictions align with true labels), and the percentage of correct classifications on the unseen test data, respectively. These metrics offer crucial insights into the model's generalizability and effectiveness in real-world fire detection scenarios.

To predict class, we define a function predict_image that enables fire classification on new images using the trained model. The function takes an image path (img_path) as input. It first

loads and resizes the image to the dimensions expected by the model (likely 224x224 pixels in this case). Then, it normalizes pixel values and adds a batch dimension for compatibility with the model. The pre-trained model is loaded from the specified path (saved_model_path). The function uses the model to predict the image, returning the predicted class name.

5.6 Hyperparameter Configuration

Deep learning models, like neural networks, require careful hyperparameter tuning for optimal performance. (Zheng et al., 2019). These parameters control the learning process but are not directly learned by the model itself. Effectively tuning hyperparameters can substantially affect the training efficiency and simplification capability of the fire detection model.

Table5. 1 Hyperparameter Configuration for the proposed model

Hyperparameter	VGG19	MobilenetV2	ResNet50	InceptionV3
Number of Epochs	40	40	40	40
Dropout	0.5	0.5	0.5	0.5
Activation	ReLU	ReLU	ReLU	ReLU
Batch Size	32	32	32	32
Optimizer	Adam, SGD	Adam, SGD	Adam, SGD	Adam, SGD
Learning Rate	0.001, 0.0001	0.001, 0.0001	0.001, 0.0001	0.001, 0.0001

CHAPTER SIX
6. RESULTS AND DISCUSSION

6.1 Chapter Overview

This section presents the study's core findings. It explores the impact of data augmentation techniques, the performance of the modified pre-trained CNN model, and the outcomes of the various experiments conducted, including those that evaluated the proposed solution. The chapter utilizes figures, graphs, and tables to facilitate a clear comparison of these results.

6.2 Results of the Study

6.2.1 Data Augmentation Results

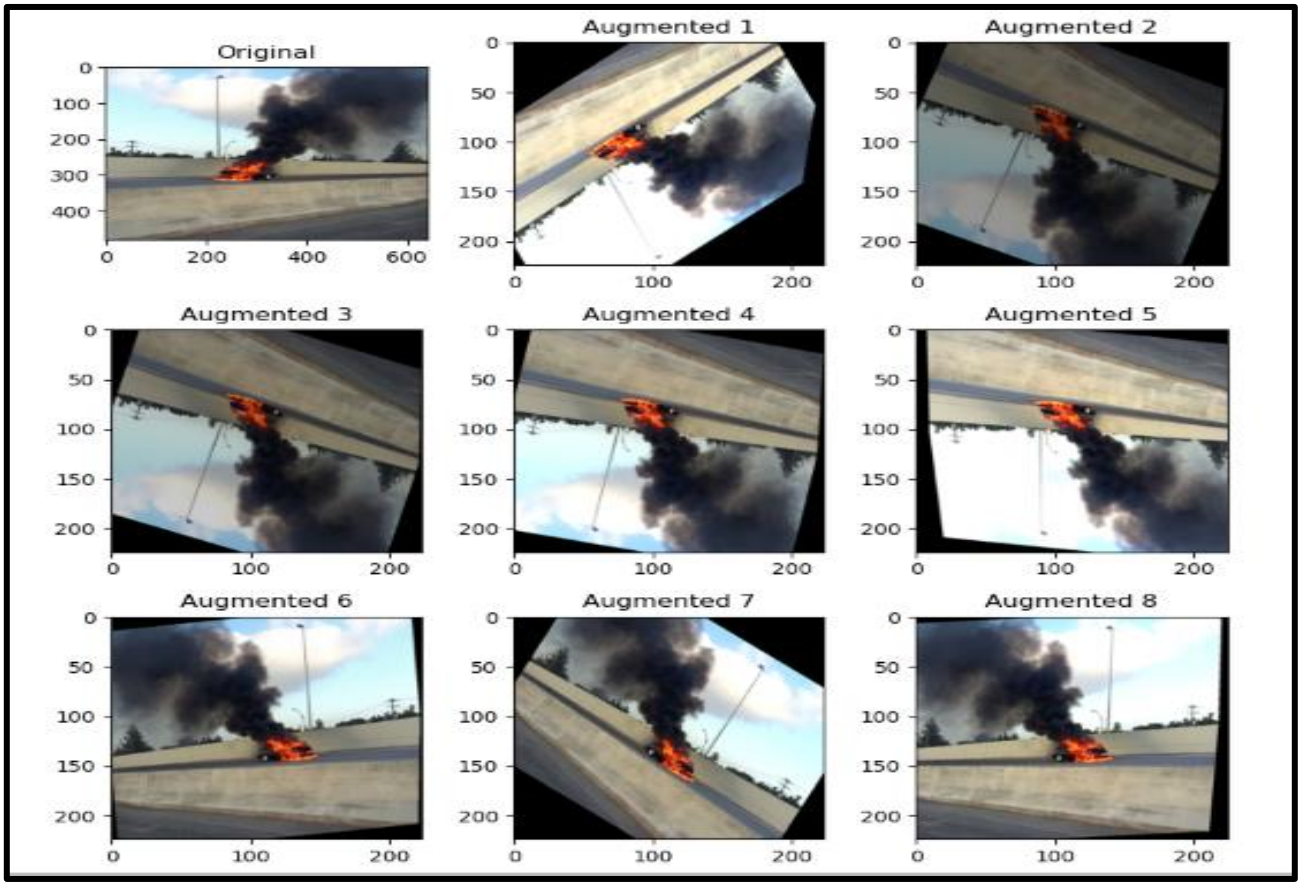


Figure 6. 1Data augmentation results.

6.2.2 Results of the Train Model from Scratch

In our study, we evaluated the performance of a fire detection model trained from scratch and compared it with models utilizing transfer learning. The model trained from scratch achieved an accuracy of 92.54%, a recall of 96.36%, a precision of 89.52%, and an F1-score of 92.81%. These results were obtained using a batch size of 32, over 40 epochs, with a 0.5 dropout rate, ReLU activation function, and Adam optimizer with a 0.001 learning rate. These metrics indicate that the model trained from scratch performed reasonably well, it required extensive computational resources and training time to reach this level of performance. Moreover, achieving these results involved careful tuning of hyperparameters to avoid overfitting and ensure the model's generalizability to unseen data.

The training process for the model trained from scratch is visualized in Figure 6.2 and Figure 6.3. It depicts the model's accuracy and loss curves over 40 training epochs. The model trained from scratch achieved 92.54% accuracy on the Fire test dataset. The confusion matrix, presented in Figure 6.4, provides a detailed view of this performance.

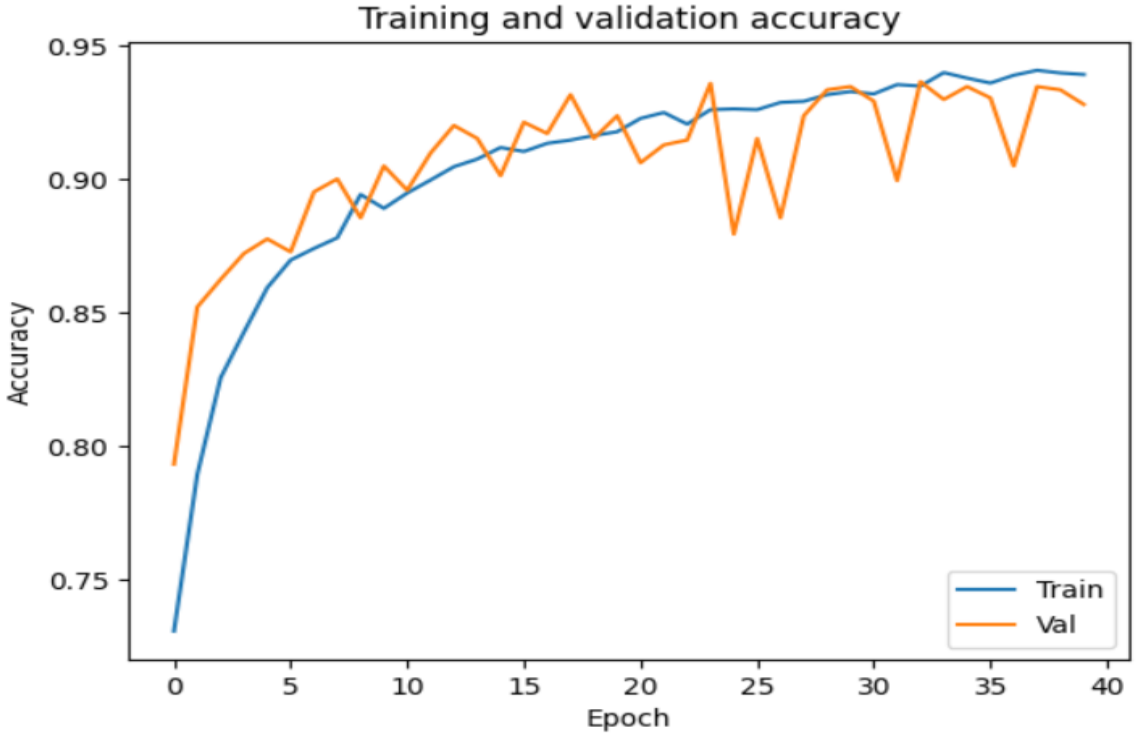


Figure 6. 2Training and validation accuracy for the trained model from scratch.

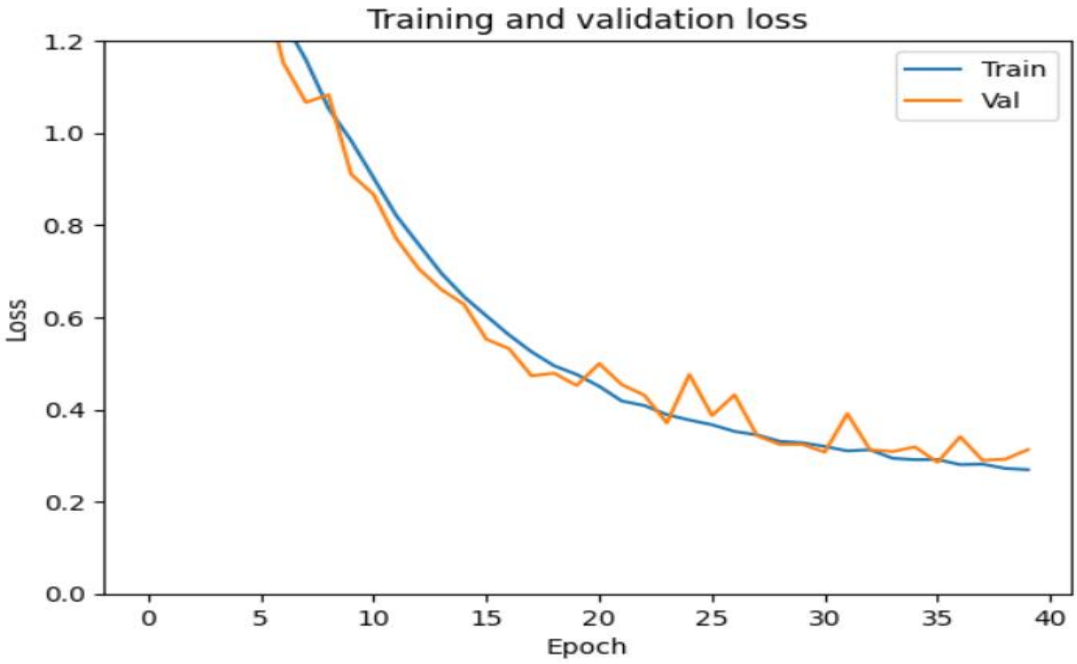


Figure 6. 3 Training and validation loss for the trained model from scratch.

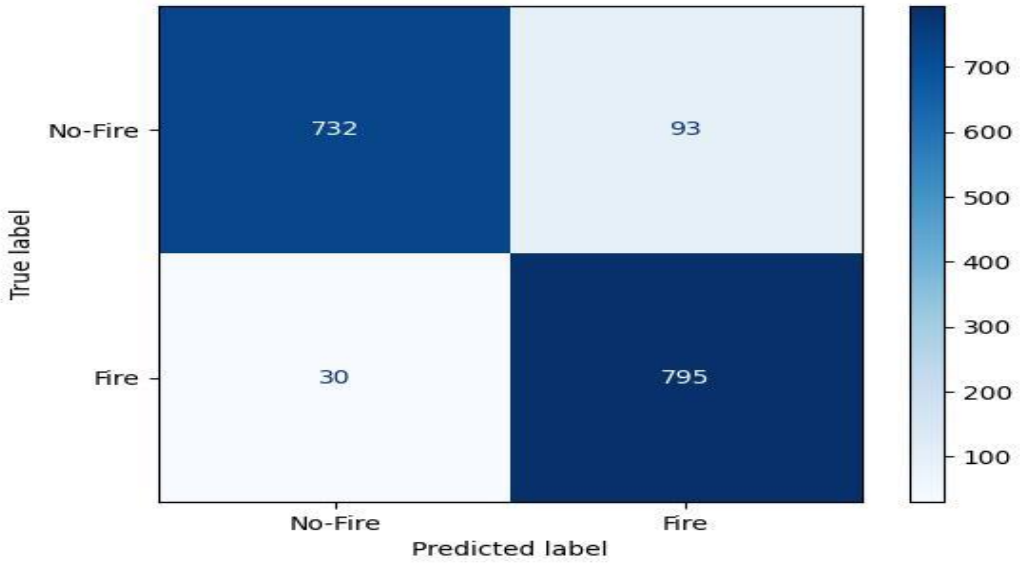


Figure 6. 4Confusion matrix of the trained model from scratch

6.2.3 Transfer Learning with VGG19 Model Results

To extract features from images, we leveraged a pre-trained VGG19 model. We reused the weights it learned from a massive image dataset (ImageNet) and kept these layers unchanged. We then added new layers on top, including ReLU for activation, average pooling for dimensionality reduction, Dropout to prevent overfitting, and finally, a Sigmoid layer for fire detection prediction. These new layers were specifically trained for our fire classification task.

Our experiments with transfer learning on our modified VGG19 model explored the impact of hyperparameters on fire detection performance. We evaluated various combinations of optimizers (Adam, SGD), dropout rates (0.5), batch sizes (32), and learning rates (0.001, 0.0001) within 40 epochs, all using the ReLU activation function. The most promising configuration achieved an accuracy of 98.36%, recall of 99.27%, F1-score of 98.37%, and precision of 97.50%. This configuration employed a batch size of 32, 40 epochs, 0.5 dropout rate, the SGD optimizer, and a learning rate of 0.001. The high recall (99.27%) indicates a excellent capability to recognize fire images, and the F1-score (98.37%) demonstrates a balanced performance between precision and recall.

To prevent overfitting and improve generalization, we employed an early stopping technique in addition to Dropout layers during the training process. This technique monitors the validation performance of the model and terminates training if the validation loss fails to improve for a predefined number of epochs.

The training process for our modified VGG19 architecture is visualized in Figures 6. 5 and 6.6. It depicts the model's accuracy and loss curves over 40 training epochs. As observed, the model demonstrates rapid learning during the initial phase, achieving a high validation accuracy of 95.5% within the first 5 epochs. This suggests the model can effectively extract relevant features from the training data.

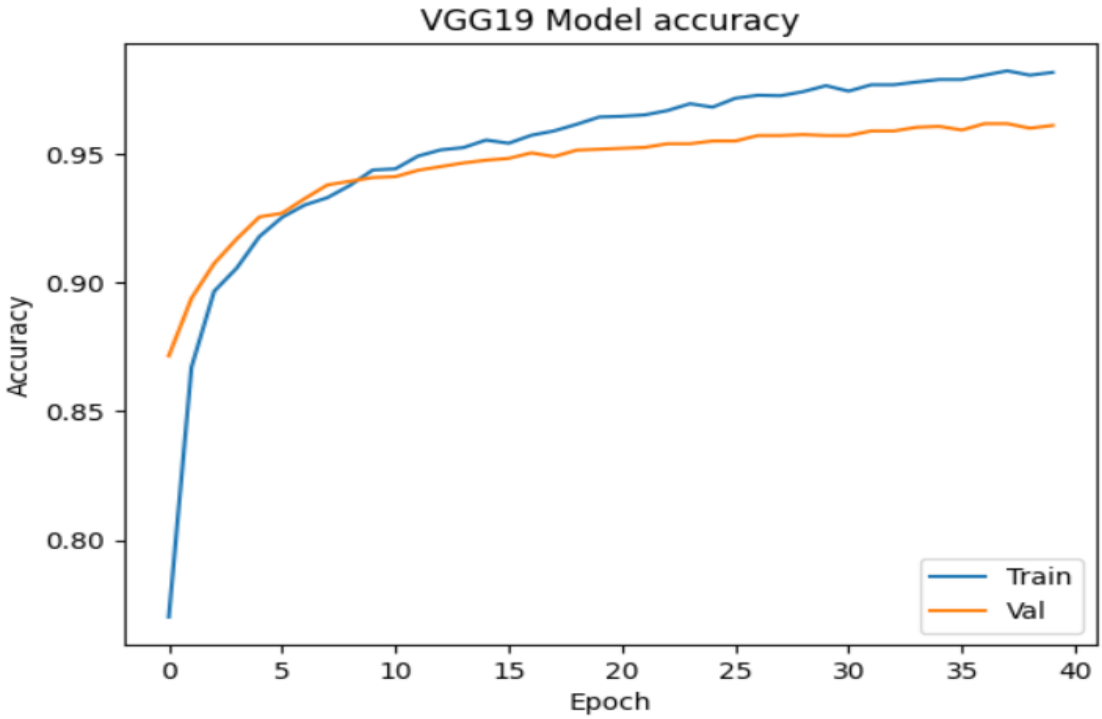


Figure 6. 5Training and validation accuracy for proposed VGG19.

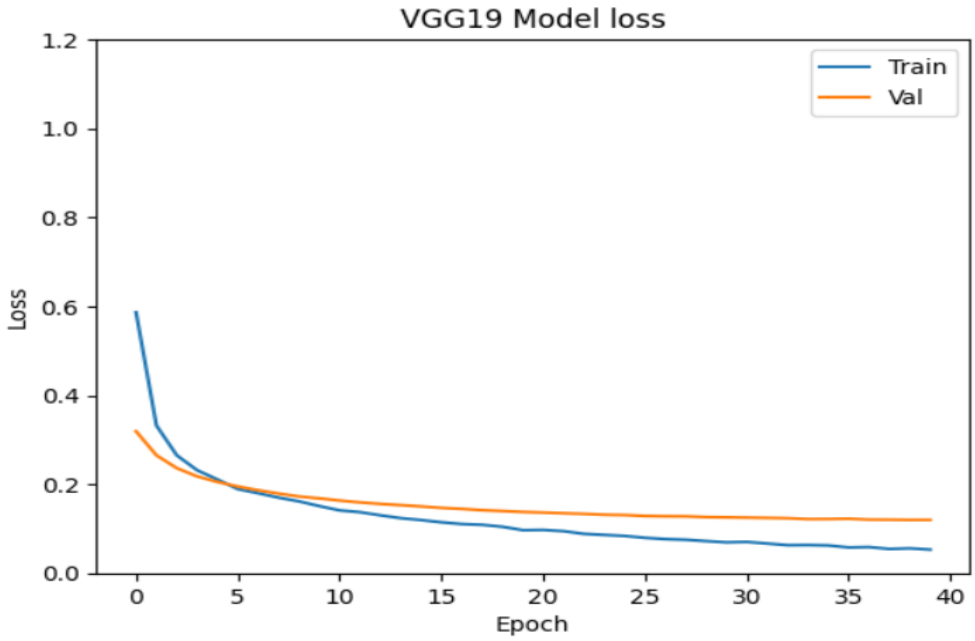


Figure 6. 6Training and validation losses for proposed VGG19.

The modified VGG19 architecture achieved an impressive 98.36% accuracy on the Fire test dataset. The confusion matrix, presented in Figure 6.7, provides a detailed view of this performance.

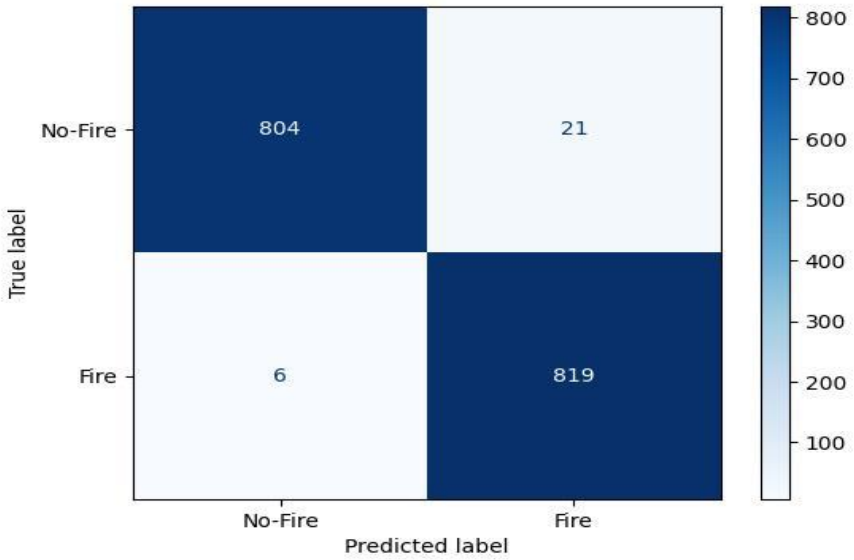


Figure 6. 7. Confusion matrix of Proposed Modified VGG19.

6.2.4 Transfer Learning with MobileNetV2 Model Results

Here we employed a pre-trained MobilenetV2 model as a feature extractor. The model's weights, learned from extensive training on the ImageNet dataset, were leveraged for feature extraction. These layers were then frozen to prevent them from adapting to the new task. To perform fire classification, we added several layers on top of the pre-trained feature extractor. These layers consisted of a ReLU activation for non-linearity, global average pooling for dimensionality reduction, a Dropout layer to prevent overfitting, and finally, a sigmoid activation for binary classification (fire/non-fire). These newly added layers were fine-tuned during training to specialize in the fire detection classification task.

We further investigated the effectiveness of transfer learning by applying it to the MobileNetV2 pre-trained model for fire detection. Similar to the VGG19 experiments, we evaluated various hyperparameter combinations including optimizers (Adam, SGD), dropout rates (0.5), batch sizes (32), and learning rates (0.001, 0.0001) within 40 epochs, all using the ReLU activation function. The best performing configuration achieved an accuracy of 95.75%, recall of 92.37%, F1-score of 95.61%, and precision of 99.09%. This configuration used a batch size of 32, 40 epochs, 0.5 dropout rate, the SGD, and a 0.001 learning rate.

While the scored accuracy (95.75%) is lower compared to VGG19, MobileNetV2 demonstrates a significant strength in precision (99.09%). This suggests the model excels at minimizing false

positives, which can be crucial depending on the application. However, the lower recall (92.37%) indicates it might miss some fire images compared to VGG19.

The training process for our modified MobilenetV2 architecture is visualized in Figure 6.8 and Figure 6.9. It depicts the model's accuracy and loss curves over 40 training epochs. As observed, the model demonstrates rapid learning during the initial phase, achieving a high validation accuracy of 92.5% within the first 5 epochs. This suggests the model can effectively extract relevant features from the training data.

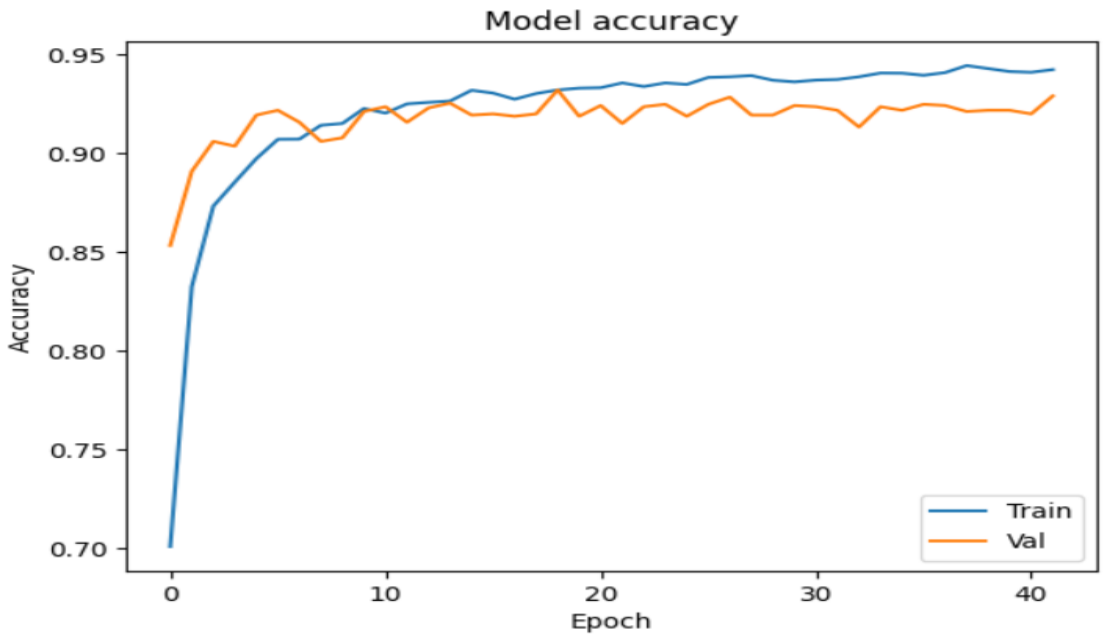


Figure 6. 8Training and validation accuracy for proposed mobilenetV2.

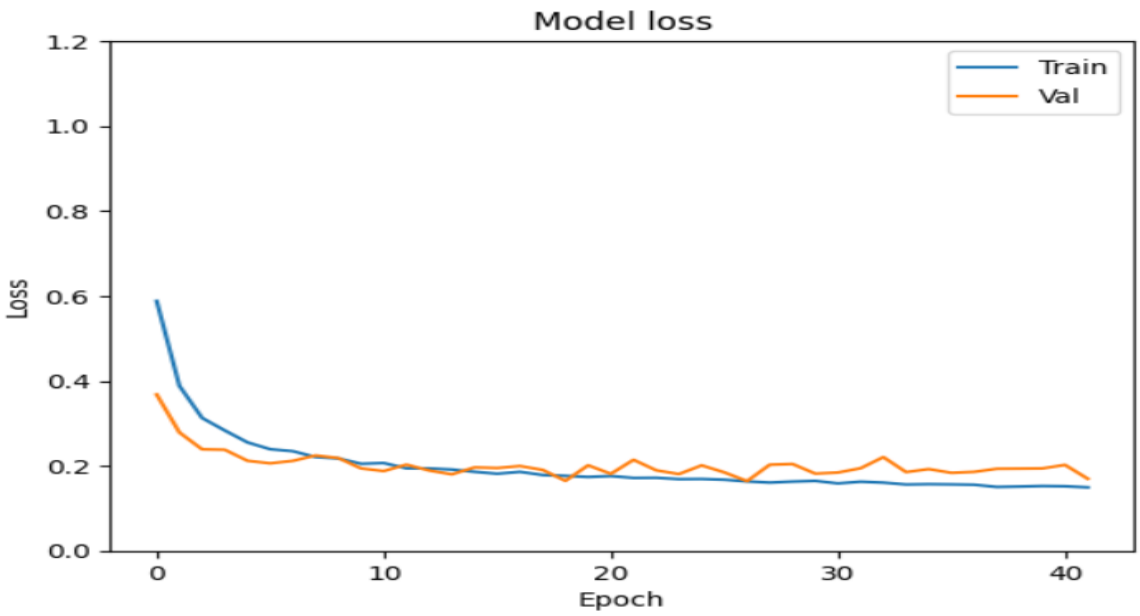


Figure 6. 9Training and validation loss for proposed mobilenetV2.

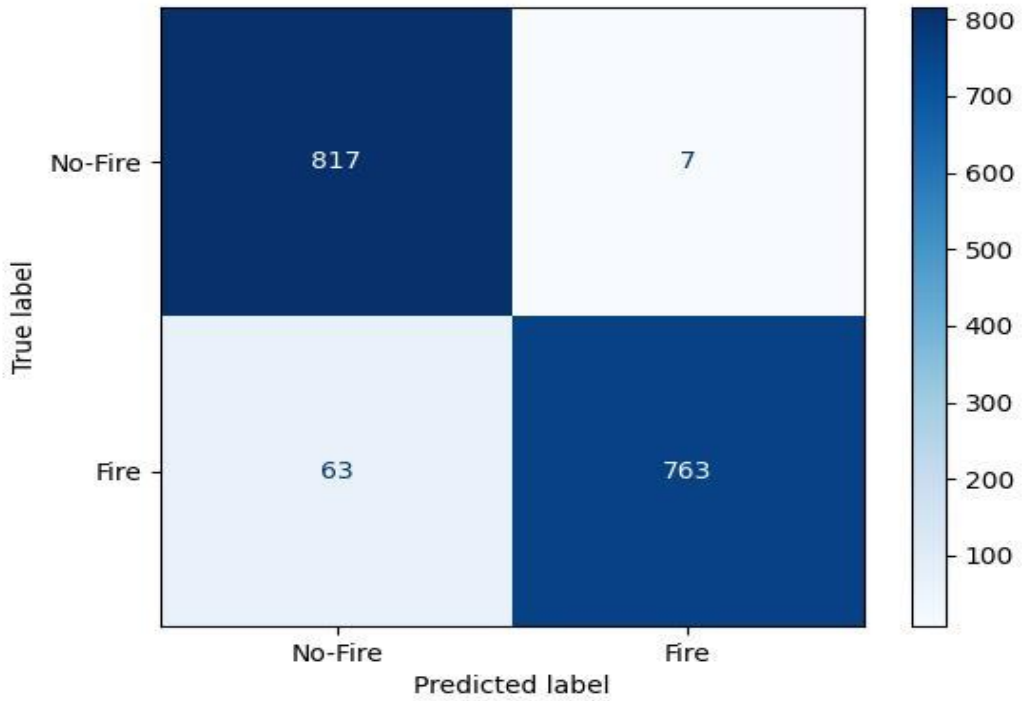


Figure 6. 10 Confusion matrix of Proposed Modified MobileV2.

6.2.5 Transfer Learning with Resnet50 Model Results

We employed transfer learning with ResNet50. The model leveraged pre-trained weights from ImageNet for feature extraction. We then added new classification layers on top, including a ReLU activation layer, average pooling for dimension reduction, dropout to avoid overfitting, and finally, a sigmoid output layer for classification. The model was trained for 40 epochs, processing images resized to 224 x 224 pixels. We also explored the potential of ResNet50, another popular pre-trained model, for fire detection classification. Following the same approach as with VGG19 and MobileNetV2, we evaluated various hyperparameters (optimizers: Adam, SGD; dropout rates: 0.5; batch sizes:32; learning rates: 0.001, 0.0001) within 40 epochs, all using the ReLU activation function. The best configuration achieved an impressive accuracy of 98.60%, a well-balanced F1-score of 98.59%, high recall (98.06%) and precision (99.14%). This configuration used a batch size of 32, 40 epochs, 0.5 dropout rate, the SGD, and a 0.001 learning rate.

These results suggest that ResNet50 performs exceptionally well on this fire detection task. The high accuracy and F1-score indicate its ability to correctly classify most fire images while maintaining a balance between precision and recall. The strong recall value (98.06%) is particularly noteworthy, suggesting it effectively minimizes missed fire detections.

The provided visualizations in Figures 6.11 and 6.12 depict the training accuracy and loss curves throughout 40 epochs. Our proposed modified ResNet50 model, leveraging transfer learning, achieves a high validation accuracy of 96% relatively early in the training process. Furthermore, the loss function exhibits a continuous downward trend, suggesting the model is still learning effectively even beyond 40 epochs.

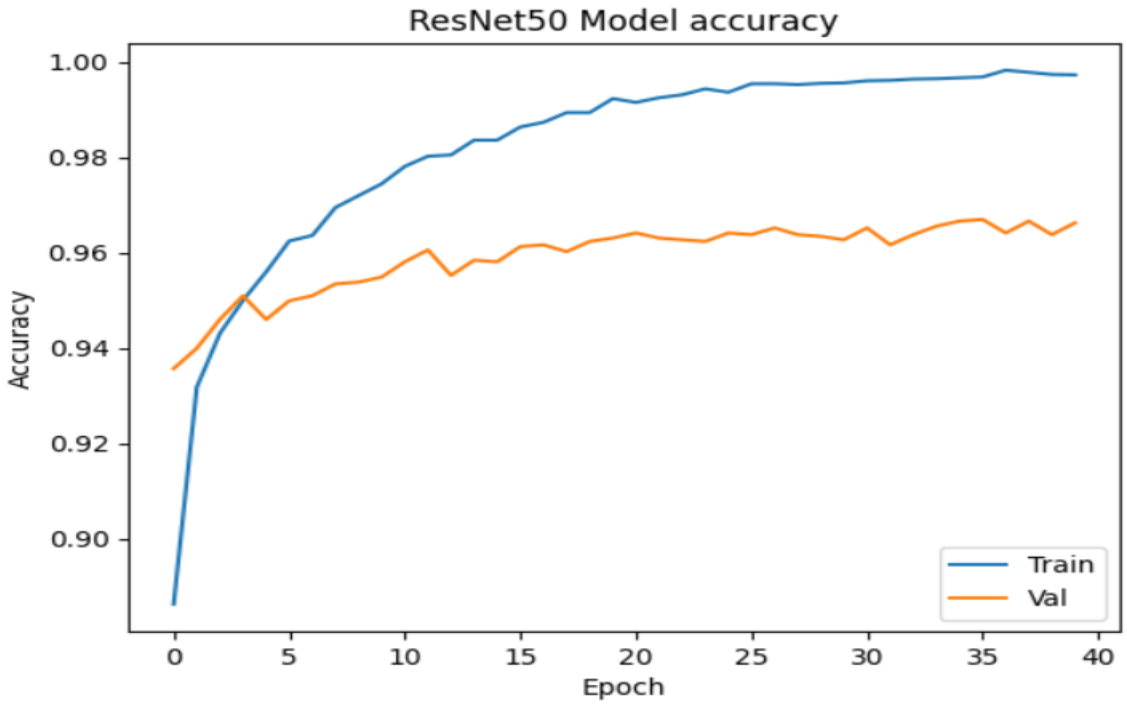


Figure 6. 11Training and validation accuracy for proposed ResNet50.

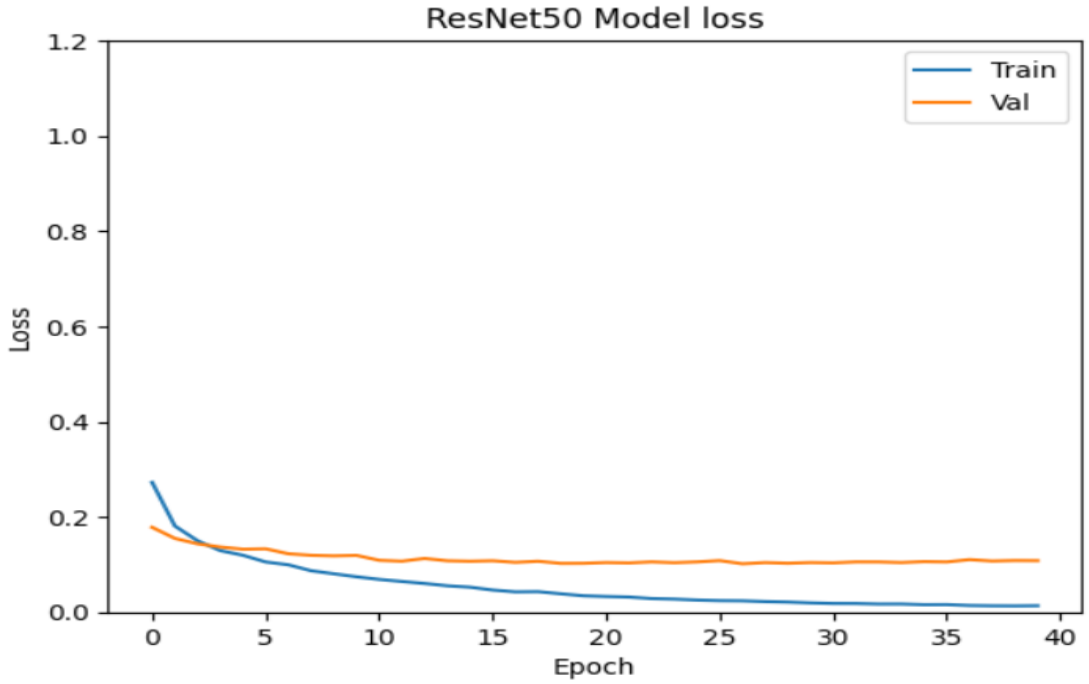


Figure 6. 12Training and validation loss for the proposed ResNet50.

The Proposed Modified InceptionV3 architecture achieved a remarkable 98.60% accuracy on the Fire test dataset. The confusion matrix, presented in Figure 6.13, provides a detailed view of this performance.

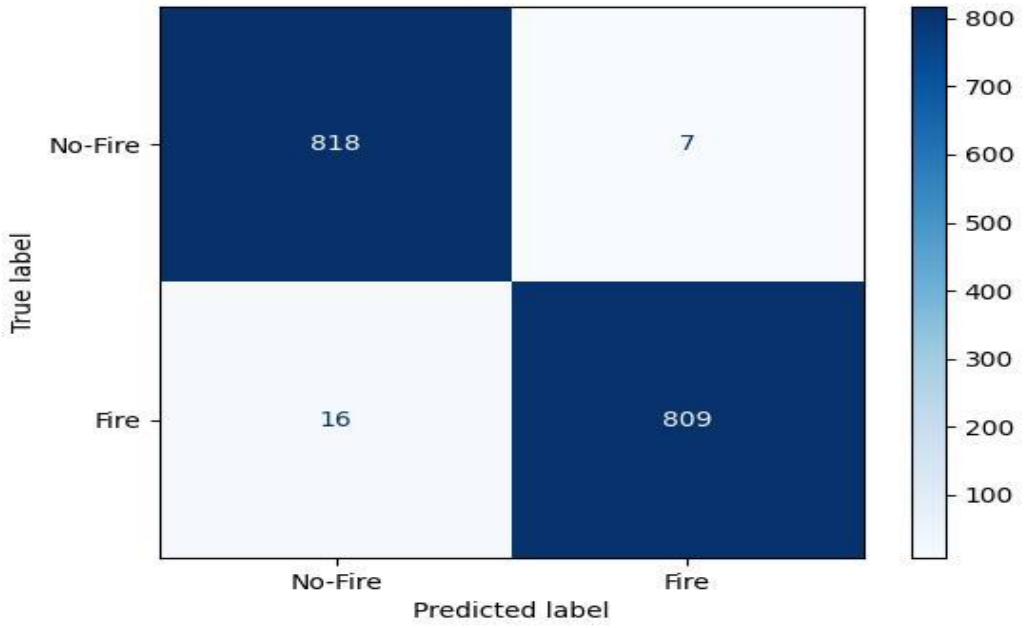


Figure 6. 13 Confusion matrix of Proposed Modified ResNet50.

6.2.6 Transfer Learning with Inception -V3 Model Results

In this study, we employed transfer learning to leverage the feature extraction capabilities of a pre-trained InceptionV3 model. Originally trained on the vast ImageNet dataset, InceptionV3 provided a robust foundation for image recognition tasks. To adapt the model for fire detection in our specific dataset, we added custom layers on top of the InceptionV3 pre-trained architecture. These layers included a ReLU activation function to introduce non-linearity, average pooling for dimensionality reduction, Dropout to mitigate overfitting during training, and a final layer with sigmoid activation for binary classification (fire vs. non-fire). We evaluated the performance of the InceptionV3 pre-trained model for fire detection classification using transfer learning. Similar to the other models, we experimented with various hyperparameter combinations (optimizers: Adam, SGD; dropout rates: 0.5; batch sizes:32; learning rates: 0.001, 0.0001) within 40 epochs, all using the ReLU activation function. The best configuration achieved an accuracy of 98.36%, a high recall of 99.15%, an F1-score of 98.37%, and a precision of 97.61%. This configuration employed a batch size of 32, 40 epochs, 0.50

dropout rate, the SGD, and a 0.001 of learning rate. InceptionV3 demonstrates a powerful performance, particularly in recall (99.15%), indicating its effectiveness in identifying fire images. However, its precision (97.61%) is slightly lower compared to some other models (e.g., ResNet50). This suggests a possibility of a few false positives, where non-fire images might be misclassified as fire.

As depicted in Figure 6.14, the proposed approach achieved rapid convergence, reaching a validation accuracy of 97%. Notably, in Figure 6.15 the loss curve continued to decrease even after 40 epochs, suggesting the potential for further performance gains with additional training.

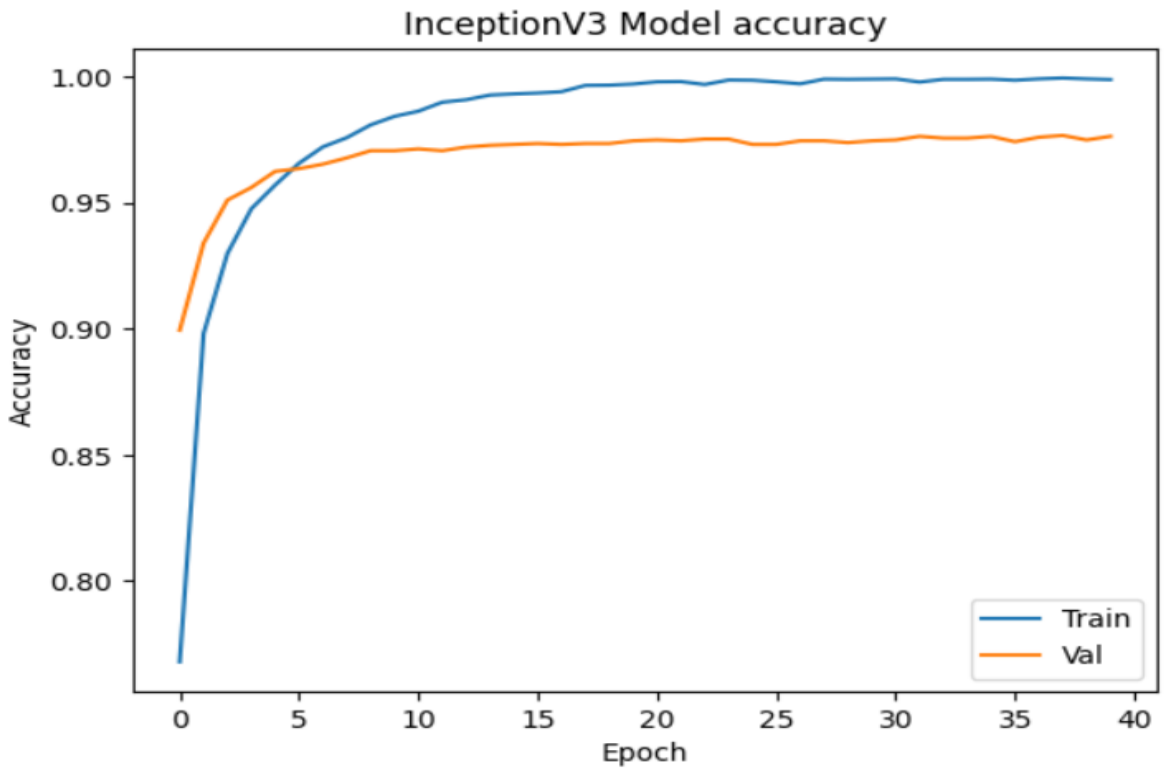


Figure 6. 14Training and validation accuracy for the proposed InceptionV3.

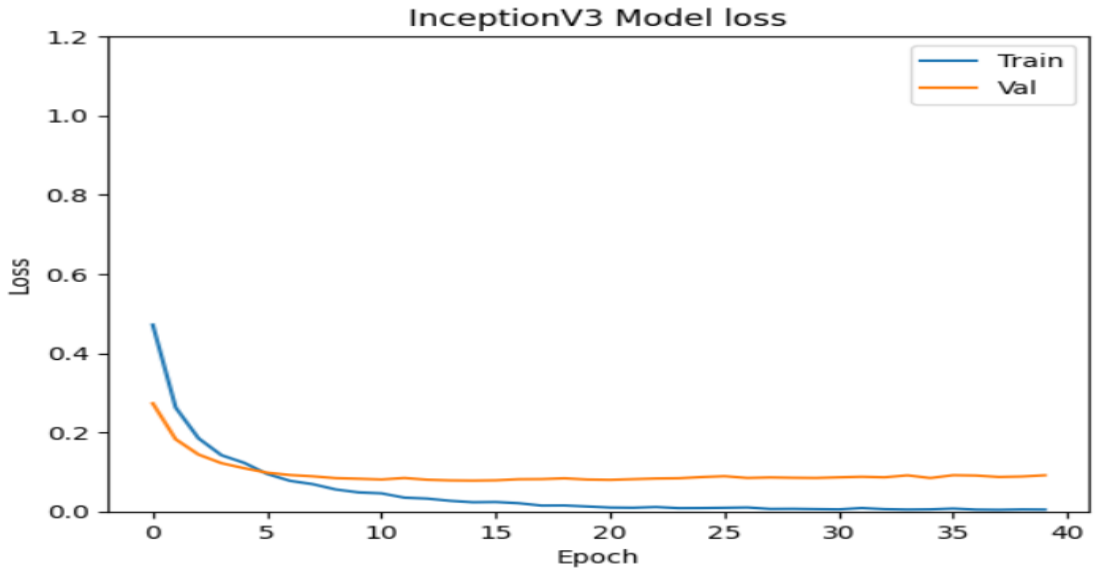


Figure 6. 15Training and validation loss for the proposed InceptionV3.

The Proposed Modified InceptionV3 architecture achieved a remarkable 98.36% accuracy on the Fire test dataset. The confusion matrix, presented in Figure 6.16, provides a detailed view of this performance.

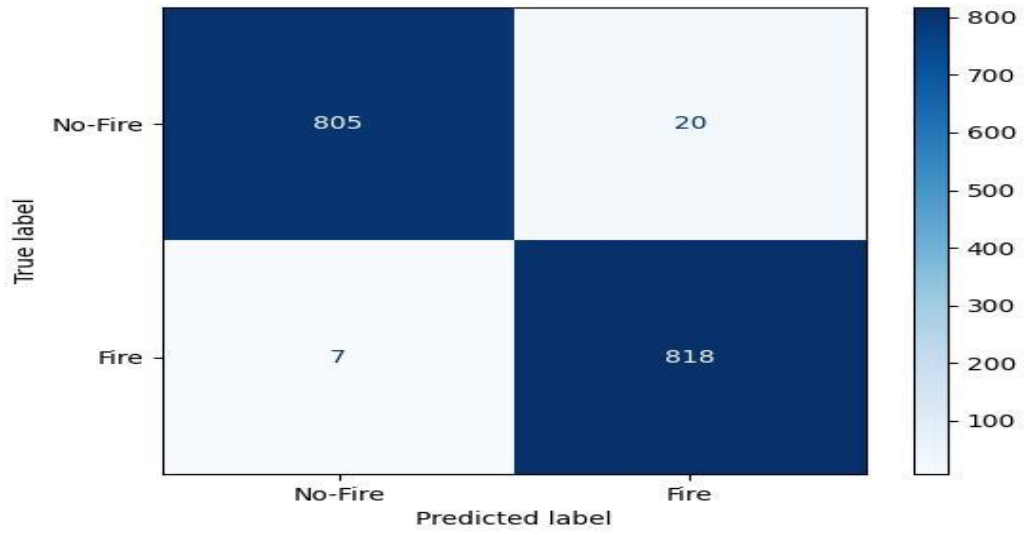


Figure 6. 16 Confusion matrix of Proposed Modified InceptionV3.

6.2.7 Comparison of Proposed Architecture Results

We employed standard validation split techniques to ensure our modified pre-trained model generalizes well on unseen data. We experimented with different split ratios for training and validation sets, including 70/30, 90/10, and 80/20. We achieved a good result from these techniques across all metrics (accuracy, recall, F1-score, precision), with the 80/20 split as we see in Table 6.1 and Table 6.2. This suggests that using a larger training set while maintaining a reasonable validation size can be beneficial for our model. However, we don't record the results of 70/30 standard validation split techniques on our documents because of got a poor result in all metrics. The outcomes result of our modified pe-trained model (VGG19, MobileNetV2, ResNet50, InceptionV3) based on an 80/200 standard validation split technique are presented in Table 6. 1

Table 6. 1 Summary of results based on 80/20 standard validation split technique.

Model	Batch size	Epochs	Dropout	Activation function	Optimizers	Learning rate	Accuracy	Recall	F1 score	Precision
VGG19	32	40	0.50	RELU	Adam	0.001	97.63	97.33	97.62	97.92
						.0001	98.16	98.87	98.13	97.41
					SGD	**0.001**	**98.36**	**99.27**	**98.37**	**97.50**
						.0001	96.57	96.87	96.17	95.49
MobileNetV2	32	40	0.50	RELU	Adam	0.001	87.69	89.93	87.96	86.06
						.0001	95.23	93.56	95.75	97.85
					SGD	**0.001**	**95.75**	**92.37**	**95.61**	**99.09**
						.0001	92.36	90.86	93.05	95.36
ResNet50	32	40	0.5	RELU	Adam	0.001	93.54	92.27	93.91	95.63
						.0001	95.81	95.81	95.82	95.76
					SGD	**0.001**	**98.60**	**98.06**	**98.59**	**99.14**
						.0001	94.96	95.01	95.01	94.17
Inception V3	32	40	0.5	RELU	Adam	0.001	92.56	92.37	92.07	91.78
						.0001	96.78	94.96	94.21	93.48
					SGD	**0.001**	**98.36**	**99.15**	**98.37**	**97.61**
						.0001	93.45	92.56	93.16	93.78

The outcomes result of our modified pe-trained model (VGG19, MobileNetV2, ResNet50, InceptionV3) based on a 90/10 standard validation split technique are presented in Table 6. 2

Table 6. 2 Summary of results based on 90/10 standard validation split technique.

Model	Batch size	Epochs	Dropout	Activation function	Optimizer	Learning rate	Accuracy	Recall	F1 score	Precision
VGG19	32	40	0.50	RELU	Adam	0.001	95.57	97.09	95.64	94.23
						.0001	**95.93**	**97.09**	**95.98**	**94.90**
					SGD	0.001	95.69	96.36	95.72	95.09
						.0001	94.90	95.39	94.93	94.47
MobileNetV2	32	40	0.50	RELU	Adam	0.001	90.25	91.17	91.01	90.86
						.0001	92.85	91.65	89.37	91.16
					SGD	**0.001**	**93.68**	**91.96**	**93.86**	**96.28**
						.0001	85.63	87.27	85.86	84.50
ResNet50	32	40	0.50	RELU	Adam	**0.001**	**96.96**	**96.84**	**96.96**	**97.08**
						.0001	96.06	96.72	96.08	95.45
					SGD	.0001	94.14	96.24	94.29	92.43
						.001	95.69	96.24	95.72	95.20
Inception V3	32	40	0.50	RELU	Adam	0.001	92.54	94.86	93.96	94.05
						.0001	94.46	94.56	94.26	93.98
					SGD	**0.001**	**94.63**	**96.85**	**95.67**	**94.87**
						.0001	92.86	93.64	92.79	91.96

Table 6. 3 Results of model training from scratch

Model	Batch size	Epochs	Dropout	Activation function	Optimizers	Learning rate	Accuracy	Recall	F1 score	Precision
CNN from Scratch	32	40	0.50	RELU	Adam	**0.001**	**92.54**	**96.36**	**92.81**	**89.52**
						.0001	90.24	88.60	90.08	91.60
					SGD	0.001	92.42	93.09	92.47	91.18
						.0001	88.42	92.84	88.91	85.30

6.3 Research Question and Answer Discussion

RQ1: *How do pre-trained CNN architectures perform when applied to fire detection in surveillance environments using transfer learning?*

Our experiments evaluated four pre-trained models (VGG19, MobileNetV2, ResNet50, InceptionV3) using transfer learning for fire detection classification. All models achieved high accuracy exceeding 95%. However, the optimal choice hinges on the specific application's priorities, particularly in uncertain surveillance environments.

ResNet50: Achieved the highest overall accuracy (98.60%) and a well-balanced F1-score (98.59%), indicating strong performance in both identifying fires and minimizing false positives. This balanced performance is crucial in uncertain environments where differentiating fire from other visual elements might be challenging.

InceptionV3: Boasted the highest recall (99.15%), suggesting it excels at catching most fire images with a minimal chance of missing them. However, its slightly lower precision (97.61%) implies a possibility of some false positives. While minimizing missed fires is important, false positives can also be problematic in uncertain environments.

MobileNetV2: Stood out for its exceptional precision (99.09%), meaning it rarely classified non-fire images as fire. However, its accuracy (95.75%) and recall (92.37%) were lower than other models, indicating it might miss more fire images.

VGG19: Delivered good overall performance with high accuracy (98.36%) and recall (99.27%). However, its F1-score (98.37%) suggests room for improvement in balancing precision and recall compared to other models.

Based on our analysis, **ResNet50** emerges as the most suitable model for fire detection in uncertain surveillance environments. Its high accuracy and well-balanced F1 score indicate its ability to handle uncertainties while effectively classifying fire images and minimizing false positives.

In contrast, models leveraging transfer learning, such as VGG19, MobileNetV2, ResNet50, and InceptionV3, showed superior performance with less training time and computational cost. The pre-trained models, which had already learned rich feature representations from large-scale datasets, adapted more effectively to our specific fire detection task with minor adjustments. These models demonstrated higher overall accuracy, recall, precision, and F1 scores compared to the model **trained from scratch**.

Specifically, the transfer learning models not only converged faster but also required fewer epochs to achieve better performance metrics. This efficiency makes transfer learning a more practical and scalable solution for real-world applications, where rapid deployment and adaptability to diverse environmental conditions are crucial.

RQ2: *How can hyperparameter optimization be used to improve the performance of pre-trained models for fire detection in surveillance environments?*

Our experiment showed that hyperparameter optimization with learning rates (0.001 vs 0.0001) and optimizers (Adam vs SGD) resulted in performance improvements for our models on an unseen dataset.

VGG19 achieved a significant accuracy boost, increasing from 97.63% to 98.16% with the Adam optimizer and a learning rate of 0.0001. Using SGD with a learning rate of 0.001, VGG19's accuracy further improved from 96.57% to 98.36%. MobileNetV2 exhibited the most substantial improvement with the Adam optimizer and a 0.0001 learning rate, raising accuracy from 87.69% to 95.23%. Additionally, MobileNetV2's accuracy increased from 92.36% to 95.75% using SGD with a 0.001 learning rate. ResNet50's accuracy improved by 2.27%, from

93.54% to 95.81%, when using the Adam optimizer with a learning rate of 0.0001. The greatest improvement for ResNet50 was seen with SGD and a 0.001 learning rate, enhancing accuracy from 94.96% to 98.60%. InceptionV3 saw a 4.22 percentage point increase in accuracy, from 92.56% to 96.78%, using the Adam optimizer with a 0.0001 learning rate. The greatest accuracy improvement for InceptionV3 was achieved with SGD and a 0.001 learning rate, raising accuracy from 93.45% to 98.36%.Both Adam and SGD optimizers achieved significant accuracy improvements across all models, suggesting that proper learning rate selection is crucial for performance.

A learning rate of 0.0001 seems to be generally beneficial for achieving significant accuracy improvements across all models (VGG19, MobileNetV2, ResNet50, InceptionV3) with the Adam optimizer.

A learning rate of 0.001 seems to be generally beneficial for achieving significant accuracy improvements across all models (VGG19, MobileNetV2, ResNet50, InceptionV3) with the SGD optimizer.

CONCLUSIONS AND RECOMMENDATIONS

CONCLUSIONS

The continuous advancements in computational power and deep learning techniques have opened doors for exploring and developing more robust and reliable fire detection systems specifically designed for uncertain surveillance environments. Often plagued by poor lighting, smoke, and varying camera angles, these environments present significant challenges. A key hurdle in developing such systems lies in the scarcity of high-quality training data that accurately reflects the complexities of real-world scenarios. We opted to leverage pre-trained deep learning models through transfer learning techniques to address this challenge. Our research confirms that training a deep learning model from scratch with limited data is less effective than utilizing transfer learning in this context. We compiled a dataset of images specifically tailored for fire detection in uncertain surveillance environments. While efforts were made to remove redundant frames extracted from the videos, further data cleaning and augmentation techniques might be necessary to optimize the training process.

Using transfer learning, our experiments evaluated four pre-trained models (ResNet50, InceptionV3, VGG19, MobileNetV2). All models achieved impressive accuracy exceeding 95%, demonstrating their strong ability to classify fire images even under potentially unclear conditions. Notably, ResNet50 emerged as the leader with the highest overall accuracy and a well-balanced F1-score, indicating good performance in identifying fires and minimizing false positives. This achievement, alongside the robust performance of other models, suggests that transfer learning can be a highly effective approach for fire detection in uncertain surveillance environments.

RECOMMENDATIONS

Our continuing efforts aim further to improve the model's performance through several avenues. First, we will refine the training dataset by potentially employing data cleaning and augmentation techniques. This will optimize the training process and potentially lead to improved model performance. Additionally, we will explore the integration of image processing techniques specifically designed for handling uncertainties in surveillance environments, such

as poor lighting, smoke, and varying camera angles. These techniques could further enhance the model's ability to identify fire in challenging conditions accurately.

Furthermore, future work will explore techniques for detecting fire and accurately locating it within the image frame. This enhanced functionality, known as fire localization, would provide valuable information for emergency response purposes, allowing for faster and more targeted interventions. Techniques like bounding box regression or heatmap generation could be explored to achieve this objective.

Looking towards the future, we are particularly interested in exploring the potential of fusion vision within the framework of transfer learning. Fusion vision combines information from multiple sensors, such as visible light and thermal cameras. In the context of fire detection, thermal cameras can be highly effective in detecting heat signatures even in low-light or smoky conditions. By incorporating information from both types of cameras through fusion vision techniques, we believe a transfer learning-based model could achieve even greater accuracy and robustness in uncertain surveillance environments. This approach can significantly improve fire detection capabilities in real-world applications.

* * *

Special Acknowledgment

This research work was funded by Adama Science and Technology University under a grant number:

ASTU/SM-R/1066/24

Adama, Ethiopia

REFERENCES

Ahmad, K., Khan, M. S., Ahmed, F., Driss, M., Boulila, W., Alazeb, A., Alsulami, M., Alshehri, M. S., Ghadi, Y. Y., & Ahmad, J. (2023). FireXnet: an explainable AI-based tailored deep learning model for wildfire detection on resource-constrained devices. *Fire Ecology*, *19*(1). https://doi.org/10.1186/s42408-023-00216-0

Bania, R. K. (2023). Ensemble of deep transfer learning models for real-time automatic detection of face mask. *Multimedia Tools and Applications*, *82*(16), 25131–25153. https://doi.org/10.1007/s11042-023-14408-y

Bansal, M., Kumar, M., Sachdeva, M., & Mittal, A. (2023). Transfer learning for image classification using VGG19: Caltech-101 image data set. *Journal of Ambient Intelligence and Humanized Computing*, *14*(4), 3609–3620. https://doi.org/10.1007/s12652-021-03488-z

Bari, A., Saini, T., & Kumar, A. (2021). Fire detection using deep transfer learning on surveillance videos. *Proceedings of the 3rd International Conference on Intelligent Communication Technologies and Virtual Mobile Networks, ICICV 2021*, 1061–1067. https://doi.org/10.1109/ICICV50876.2021.9388485

Cakir, F., He, K., Xia, X., Kulis, B., & Sclaroff, S. (2019). Deep Metric Learning to Rank. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1861–1870. https://doi.org/10.1109/CVPR.2019.00196

Chen, T. (2023). Effect and Significance of Surveillance Cameras for Individuals and Society. *Lecture Notes in Education Psychology and Public Media*, *5*(1), 80–86. https://doi.org/10.54254/2753-7048/5/20220418

Chino, D. Y. T., Avalhais, L. P. S., Rodrigues, J. F., & Traina, A. J. M. (2015). *BoWFire: Detection of Fire in Still Images by Integrating Pixel Color and Texture Analysis*. https://doi.org/10.1109/SIBGRAPI.2015.19

Cody, T., & Beling, P. A. (2023). A Systems Theory of Transfer Learning. *IEEE Systems Journal*, *17*(1), 26–37. https://doi.org/10.1109/JSYST.2022.3224650

Courbat, J., Pascu, M., Gutmacher, D., Briand, D., Wöllenstein, J., Hoefer, U., Severin, K., & de Rooij, N. F. (2011). A colorimetric CO sensor for fire detection. *Procedia Engineering*, *25*, 1329–1332. https://doi.org/10.1016/j.proeng.2011.12.328

Das, S., G, L., Prakash, S. J., Dey, N. S., Panuganti, J., & Poojitha, R. (2023). Lung Cancer Detection and Classification using Transfer Learning with Pre-trained VGG19 Convolutional Neural Networks. *2023 3rd International Conference on Emerging*

Frontiers in Electrical and Electronic Technologies (ICEFEET), 1–6. https://doi.org/10.1109/ICEFEET59656.2023.10452195

Dogan, S., Datta Barua, P., Kutlu, H., Baygin, M., Fujita, H., Tuncer, T., & Acharya, U. R. (2022). Automated accurate fire detection system using ensemble pretrained residual network. *Expert Systems with Applications*, *203*, 117407. https://doi.org/10.1016/j.eswa.2022.117407

Dong, K., Zhou, C., Ruan, Y., & Li, Y. (2020). MobileNetV2 Model for Image Classification. *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*, 476–480. https://doi.org/10.1109/ITCA52113.2020.00106

Dua, M., Kumar, M., Singh Charan, G., & Sagar Ravi, P. (2020). An Improved Approach for Fire Detection using Deep Learning Models. *2020 International Conference on Industry 4.0 Technology, I4Tech 2020*, 171–175. https://doi.org/10.1109/I4Tech48345.2020.9102697

Foggia, P., Saggese, A., & Vento, M. (2015). Real-Time Fire Detection for Video-Surveillance Applications Using a Combination of Experts Based on Color, Shape, and Motion. *IEEE Transactions on Circuits and Systems for Video Technology*, *25*(9), 1545–1556. https://doi.org/10.1109/TCSVT.2015.2392531

Fonollosa, J., Solórzano, A., Jiménez-Soto, J. M., Oller-Moreno, S., & Marco, S. (2016). Gas Sensor Array for Reliable Fire Detection. *Procedia Engineering*, *168*, 444–447. https://doi.org/10.1016/j.proeng.2016.11.540

Frizzi, S., Kaabi, R., Bouchouicha, M., Ginoux, J.-M., Moreau, E., & Fnaiech, F. (2016). Convolutional neural network for video fire and smoke detection. *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 877–882. https://doi.org/10.1109/IECON.2016.7793196

Hall, S., & Evarts, B. (2022). *Fire Loss in the United States During 2021 (NFPA ®) Key Findings*.

Hassan, A., & Audu, A. I. (2022). *TRADITIONAL SENSOR-BASED AND COMPUTER VISION-BASED FIRE DETECTION SYSTEMS: A REVIEW*. *18*(3), 469–492. www.azojete.com.ng

Hossain, Md. B., Iqbal, S. M. H. S., Islam, Md. M., Akhtar, Md. N., & Sarker, I. H. (2022). Transfer learning with fine-tuned deep CNN ResNet50 model for classifying COVID-19 from chest X-ray images. *Informatics in Medicine Unlocked*, *30*, 100916. https://doi.org/10.1016/j.imu.2022.100916

Idrissi, I., Azizi, M., & Moussaoui, O. (2021). Accelerating the update of a DL-based IDS for IoT using deep transfer learning. *Indonesian Journal of Electrical Engineering and Computer Science*, *23*(2), 1059–1067. https://doi.org/10.11591/ijeecs.v23.i2.pp1059-1067

Ivanova, B., Shoilekova, K., & Rusev, R. (2024). *Trends and Challenges in Surveillance - A Systematic Review of Camera Systems Implementing Artificial Intelligence* (pp. 103–112). https://doi.org/10.1007/978-3-031-53549-9_11

Jin, C., Wang, T., Alhusaini, N., Zhao, S., Liu, H., Xu, K., & Zhang, J. (2023). Video Fire Detection Methods Based on Deep Learning: Datasets, Methods, and Future Directions. *Fire*, *6*(8), 315. https://doi.org/10.3390/fire6080315

Khan, F., Xu, Z., Sun, J., Khan, F. M., Ahmed, A., & Zhao, Y. (2022a). Recent Advances in Sensors for Fire Detection. In *Sensors* (Vol. 22, Issue 9). MDPI. https://doi.org/10.3390/s22093310

Khan, F., Xu, Z., Sun, J., Khan, F. M., Ahmed, A., & Zhao, Y. (2022b). Recent Advances in Sensors for Fire Detection. *Sensors*, *22*(9), 3310. https://doi.org/10.3390/s22093310

Khan, M. U., Saeed, Z., Raza, A., Abbasi, Z., Ali, S. Z.-Z., & Khan, H. (2022). Deep Learning-based Decision Support System for classification of COVID-19 and Pneumonia patients. *JAREE (Journal on Advanced Research in Electrical Engineering)*, *6*(1). https://doi.org/10.12962/jaree.v6i1.229

Kim, H. E., Cosa-Linan, A., Santhanam, N., Jannesari, M., Maros, M. E., & Ganslandt, T. (2022). Transfer learning for medical image classification: a literature review. *BMC Medical Imaging*, *22*(1), 69. https://doi.org/10.1186/s12880-022-00793-7

Kumar, D., Pandey, R. C., & Mishra, A. K. (2024). A review of image features extraction techniques and their applications in image forensic. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-023-17950-x

Li, P., & Zhao, W. (2020). Image fire detection algorithms based on convolutional neural networks. *Case Studies in Thermal Engineering*, *19*. https://doi.org/10.1016/j.csite.2020.100625

M, H., & M.N, S. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, *5*(2), 01–11. https://doi.org/10.5121/ijdkp.2015.5201

Mahalle, V. S., Kandoi, N. M., & Patil, S. B. (2024). *Transfer Learning by Fine-Tuning Pre-trained Convolutional Neural Network Architectures for Image Recognition* (pp. 273–287). https://doi.org/10.1007/978-981-99-9179-2_21

Mahmoud, H. A. H., Alharbi, A. H., & Alghamdi, N. S. (2022). Time-efficient fire detection convolutional neural network coupled with transfer learning. *Intelligent Automation and Soft Computing*, *31*(3), 1393–1403. https://doi.org/10.32604/IASC.2022.020629

Majid, S., Alenezi, F., Masood, S., Ahmad, M., Gündüz, E. S., & Polat, K. (2022). Attention based CNN model for fire detection and localization in real-world images. *Expert Systems with Applications*, *189*, 116114. https://doi.org/10.1016/j.eswa.2021.116114

Mendonça, M. O. K., Netto, S. L., Diniz, P. S. R., & Theodoridis, S. (2024). Machine learning: Review and trends. *Signal Processing and Machine Learning Theory*, 869–959.

Mikolajczyk, A., & Grochowski, M. (2018). Data augmentation for improving deep learning in image classification problem. *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, 117–122. https://doi.org/10.1109/IIPHDW.2018.8388338

Muhammad, K., Ahmad, J., Lv, Z., Bellavista, P., Yang, P., & Baik, S. W. (2019). Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *49*(7), 1419–1434. https://doi.org/10.1109/TSMC.2018.2830099

Muhammad, K., Khan, S., Elhoseny, M., Hassan Ahmed, S., & Wook Baik, S. (2019). Efficient Fire Detection for Uncertain Surveillance Environment. *IEEE Transactions on Industrial Informatics*, *15*(5), 3113–3122. https://doi.org/10.1109/TII.2019.2897594

Mukti, I. Z., & Biswas, D. (2019). Transfer Learning Based Plant Diseases Detection Using ResNet50. *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, 1–6. https://doi.org/10.1109/EICT48899.2019.9068805

Mumuni, A., & Mumuni, F. (2022). Data augmentation: A comprehensive survey of modern approaches. *Array*, *16*, 100258. https://doi.org/10.1016/j.array.2022.100258

Naidu, G., Zuva, T., & Sibanda, E. M. (2023). *A Review of Evaluation Metrics in Machine Learning Algorithms* (pp. 15–25). https://doi.org/10.1007/978-3-031-35314-7_2

O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., Riordan, D., & Walsh, J. (2020). *Deep Learning vs. Traditional Computer Vision* (pp. 128–144). https://doi.org/10.1007/978-3-030-17795-9_10

Pednekar, P., Srivastava, A., & Jadhav, A. (2023). Fire Detection using Transfer Learning and Pre-Trained Model. *2023 3rd International Conference on Intelligent Technologies, CONIT 2023*. https://doi.org/10.1109/CONIT59222.2023.10205560

Perilla, F. S., Villanueva, G. R., Cacanindin, N. M., & Palaoag, T. D. (2018). Fire Safety and Alert System Using Arduino Sensors with IoT Integration. *Proceedings of the 2018 7th*

International Conference on Software and Computer Applications, 199–203. https://doi.org/10.1145/3185089.3185121

Rafiq, R. Bin, & Albert, M. V. (2022). *Transfer Learning: Leveraging Trained Models on Novel Tasks* (pp. 65–74). https://doi.org/10.1007/978-3-030-84729-6_4

Reddy, S. N., & Raju, B. V. (2023). *Brain Tumour Detection Using Deep Models*. https://doi.org/10.32628/IJSRST

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

Sarker, I. H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science*, *2*(6), 420. https://doi.org/10.1007/s42979-021-00815-1

Shaha, M., & Pawar, M. (2018). Transfer Learning for Image Classification. *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 656–660. https://doi.org/10.1109/ICECA.2018.8474802

Sharma, J., Granmo, O.-C., Goodwin, M., & Fidje, J. T. (2017). *Deep Convolutional Neural Networks for Fire Detection in Images* (pp. 183–193). https://doi.org/10.1007/978-3-319-65172-9_16

Singh, A., Bhatt, S., Nayak, V., & Shah, M. (2023). Automation of surveillance systems using deep learning and facial recognition. *International Journal of System Assurance Engineering and Management*, *14*, 236–245. https://doi.org/10.1007/s13198-022-01844-6

Suyal, P., Bhatt, A. K., Pant, J., & Mohan, L. (2021). Comparative analysis between Traditional learning and Deep learning. *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, 376–379. https://doi.org/10.1109/ICAC3N53548.2021.9725370

Talaei Khoei, T., Ould Slimane, H., & Kaabouch, N. (2023). Deep learning: systematic review, models, challenges, and research directions. *Neural Computing and Applications*, *35*(31), 23103–23124. https://doi.org/10.1007/s00521-023-08957-4

Texier, G., Pellegrin, L., Vignal, C., Meynard, J.-B., Deparis, X., & Chaudet, H. (2017). Dealing with uncertainty when using a surveillance system. *International Journal of Medical Informatics*, *104*, 65–73. https://doi.org/10.1016/j.ijmedinf.2017.05.006

Ullah, N., Mohmand, M. I., Ullah, K., Gismalla, M. S. M., Ali, L., Khan, S. U., & Ullah, N. (2022). Diabetic Retinopathy Detection Using Genetic Algorithm-Based CNN Features and Error Correction Output Code SVM Framework Classification Model. *Wireless*

Communications and Mobile Computing, *2022*, 1–13. https://doi.org/10.1155/2022/7095528

Waisberg, E., Ong, J., Kamran, S. A., Paladugu, P., Zaman, N., Lee, A. G., & Tavakkoli, A. (2023). Transfer learning as an AI-based solution to address limited datasets in space medicine. *Life Sciences in Space Research*, *36*, 36–38. https://doi.org/10.1016/j.lssr.2022.12.002

Welsh, B. C., Piza, E. L., Thomas, A. L., & Farrington, D. P. (2020). Private Security and Closed-Circuit Television (CCTV) Surveillance: A Systematic Review of Function and Performance. *Journal of Contemporary Criminal Justice*, *36*(1), 56–69. https://doi.org/10.1177/1043986219890192

Wiley, V., & Lucas, T. (2018). Computer Vision and Image Processing: A Paper Review. *International Journal of Artificial Intelligence Research*, *2*(1), 22. https://doi.org/10.29099/ijair.v2i1.42

Wu, S., Zhang, X., Liu, R., & Li, B. (2023). A dataset for fire and smoke object detection. *Multimedia Tools and Applications*, *82*(5), 6707–6726. https://doi.org/10.1007/s11042-022-13580-x

Xie, J., Zheng, Y., Du, R., Xiong, W., Cao, Y., Ma, Z., Cao, D., & Guo, J. (2021). Deep Learning-Based Computer Vision for Surveillance in ITS: Evaluation of State-of-the-Art Methods. *IEEE Transactions on Vehicular Technology*, *70*(4), 3027–3042. https://doi.org/10.1109/TVT.2021.3065250

Xu, M., Yoon, S., Fuentes, A., & Park, D. S. (2023). A Comprehensive Survey of Image Augmentation Techniques for Deep Learning. *Pattern Recognition*, *137*, 109347. https://doi.org/10.1016/j.patcog.2023.109347

Yang, X., Song, Z., King, I., & Xu, Z. (2023). A Survey on Deep Semi-Supervised Learning. *IEEE Transactions on Knowledge and Data Engineering*, *35*(9), 8934–8954. https://doi.org/10.1109/TKDE.2022.3220219

Yogesh, P. S., Prakash, P. K., Santosh, R. A., Shrishail, S. A., & Shital, M. (2022). Image fire detection algorithms based on convolution alneural networks. *International Journal of Advances in Engineering and Management (IJAEM)*, *4*, 2395–5252. https://doi.org/10.35629/5252-04050814

Zhang, Z., & Sabuncu, M. R. (2018). *Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels*. http://arxiv.org/abs/1805.07836

Zheng, M., Tang, W., & Zhao, X. (2019). Hyperparameter optimization of neural network-driven spatial models accelerated using cyber-enabled high-performance computing.

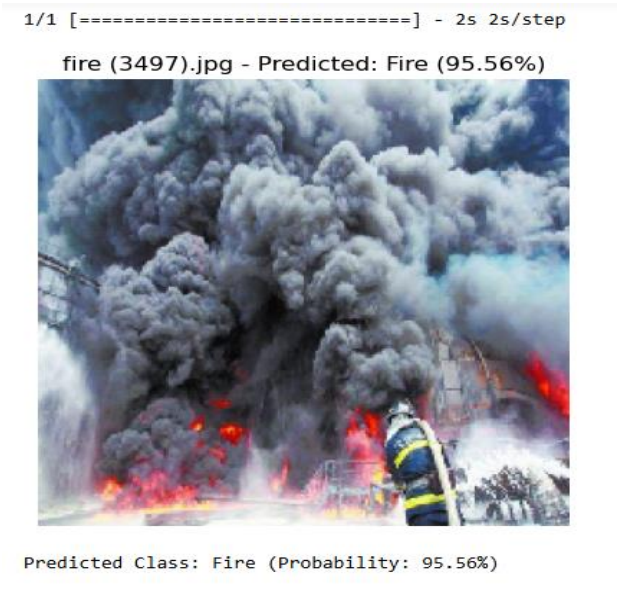
International Journal of Geographical Information Science, *33*(2), 314–345. https://doi.org/10.1080/13658816.2018.1530355

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A Comprehensive Survey on Transfer Learning. In *Proceedings of the IEEE* (Vol. 109, Issue 1, pp. 43–76). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/JPROC.2020.3004555

APPENDICES

Appendix A: Result predicted of modified proposed models on Test dataset.

1. MobilenetV2 Model Predicted Results.



2. VGG19 Model Predicted Results

3. ResNet50 Model Predicted Results

4. InceptionV3 Model Predicted Results