FireGuard: A CRNN-Based Forest Fire Detection System

Submitted in partial fulfilment of requirements to CSE (Data Science)

Project - 1 Mini Project (CD-363)

III/IV B. Tech CSE(DS) (VI Semester)

Submitted by

Batch No. 06

Y21CD065 – Y Meghana

Y21CD040 – P Bhanu Saketh

Under the Guidance of

Smt. K. Aravinda

Assistant Professor

May-2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

R.V.R. & J.C. COLLEGE OF ENGINEERING (AUTONOMOUS)

(NAAC A+ Grade) (Approved by A.I.C.T.E.)

(Affiliated to Acharya Nagarjuna University)

Chandramoulipuram :: Chowdavaram

Guntur - 522 019

R.V.R. & J.C. COLLEGE OF ENGINEERING

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

CERTIFICATE

This is to Certify that this Mini Project work entitled **“FireGuard: A CRNN-Based Forest Fire Detection System”** is the bonafide work of **Y Meghana(Y21CD065), P Bhanu Saketh(Y21CD040)** of **III/IV B.Tech** who carried the work under my supervision, and submitted in the partial fulfilment of the requirements to **Project - 1 Mini Project (CD-363)** during the year 2023-2024.

Smt. K. Aravinda	**Dr. P. Srinivasa Rao**	**Dr. M.V.P. Chandra Sekhara Rao**
Assistant Professor	Assistant professor	Prof. & HOD, CSE(DS)
(Project Guide)	(Project In-charge)	

ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without proper suggestion, guidance, and environment. Combination of these three factors acts like a backbone to our project work **"FireGuard: A CRNN-Based Forest Fire Detection System"**.

We are profoundly pleased to express our deep sense of gratitude and respect towards the management of the **R. V. R. & J. C. College of Engineering**, for providing the resources to complete the project.

We are very much thankful to **Dr. Kolla Srinivas**, Principal of **R. V. R. & J. C. College of Engineering** for allowing us to deliver the project successfully.

We are greatly indebted to **Dr. M.V.P. Chandra Sekhara Rao, Professor and Head, Department of Computer Science and Engineering (Data Science)** for providing the laboratory facilities fully as and when required and for giving us the opportunity to carry the project work in the college.

We are also thankful to our Project Coordinator **Dr. P. Srinivasa Rao** who helped us in each step of our Project.

We extend our deep sense of gratitude to our Guide, **Smt.K.Aravinda** and other Faculty Members & Support staff for their valuable suggestions, guidance, and constructive ideas in every step, which was indeed of great help towards the successful completion of our project.

Y21CD065 – Y Meghana

Y21CD040 – P Bhanu Saketh

ABSTRACT

Rising global temperatures have intensified the frequency and severity of wildfires, posing significant threats to forests and biodiversity. These wildfires, driven by both natural occurrences and human activities, have become a pressing global concern. In response to this challenge, a novel lightweight Convolutional Recurrent Neural Network (CRNN) model is proposed, specifically tailored for wildfire detection using RGB images. The CRNN architecture is optimized for compatibility with aerial images and video surveillance systems, enabling seamless integration with on-site image processing. This approach enhances the efficiency and responsiveness of wildfire detection efforts, particularly in remote or resource-constrained environments. Notably, the system autonomously issues wildfire alerts without transmitting images to cloud servers, reducing latency and reliance on external infrastructure. Practical deployment scenarios include monitoring forested areas, national parks, and other vulnerable regions susceptible to wildfires. Integration into existing wildfire management systems provides decision-makers with timely information to facilitate effective prevention and response strategies. Evaluation results demonstrate the model's promising performance, achieving high accuracy of 97.93% and F1-score of 97%, thus offering a robust solution to address the escalating wildfire crisis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

1. INTRODUCTION

1.1 Introduction

The ongoing threat of wildfires presents a significant challenge globally, with increasing frequency and severity attributed to rising temperatures and human activities. Traditional monitoring methods, reliant on satellite imaging, face limitations due to delayed monitoring and natural factors like cloud cover or small fire size. To address these challenges, this proposed approach based is on deep learning techniques for real-time wildfire detection using edge devices such as CCTV cameras and UAV aerial photographs.

Unlike conventional methods, which may struggle with image preprocessing or feature extraction, the proposed method harnesses the power of deep learning, specifically Convolutional Recurrent Neural Networks (CRNNs), to enhance detection accuracy. By automatically adjusting filters during training, CRNNs offer a more effective solution for wildfire detection tasks, even without the use of thermal cameras, thus reducing costs and increasing accessibility. Previous research has explored various techniques for fire and smoke detection, including image processing and artificial intelligence. However, the proposed method stands out for its utilization of conventional cameras and synthetic datasets to generate realistic smoke images. While limited to daytime operations, this approach offers advantages in cost-effectiveness and dataset construction, facilitating comprehensive training and validation.

By comparing and validating the proposed FireGuard method against existing CNN-based approaches, the research aims to advance wildfire detection and mitigation strategies, particularly in forest environments. In summary, the proposed method offers a promising solution to the pressing challenge of wildfire detection, providing real-time monitoring capabilities and overcoming limitations associated with traditional satellite-based methods. By leveraging deep learning techniques and edge devices, the approach demonstrates significant potential for improving wildfire prevention and response efforts, ultimately enhancing environmental conservation and public safety on a global scale.

1.2 Problem Statement

The development of a real-time wildfire detection system is essential for swiftly identifying and alerting authorities to potential fire outbreaks, mitigating damage to forests, biodiversity, and human communities. By utilizing advanced image processing techniques, the system can analyze RGB images in real-time, accurately detecting wildfires as they occur. Through seamless integration with existing monitoring infrastructure, it enhances response efforts by ensuring the prompt deployment of resources to contain and extinguish fires before they escalate. This proactive approach not only reduces the environmental impact of wildfires but also safeguards lives and property. Leveraging machine learning algorithms and real-time data analysis, the system distinguishes between natural phenomena and potential wildfire events, minimizing false alarms and enabling targeted interventions. Moreover, its continuous monitoring capabilities enhance situational awareness, empowering authorities to manage fire risks effectively and allocate resources efficiently. By leveraging automation and predictive analytics, decision-makers can make informed choices and implement preventive measures, reducing the likelihood of catastrophic wildfire incidents. Overall, the real-time wildfire detection system plays a crucial role in enhancing wildfire management and response capabilities, safeguarding ecosystems, and protecting communities.

1.3 Objectives of the Study

The project aims to deploy a real-time wildfire detection system utilizing CRNN technology, crucial for preserving forests, protecting human health, and ensuring animal safety. By analyzing images in real-time, the system enables swift response actions, minimizing environmental damage and enhancing overall safety. Its applications span forest conservation efforts, proactive health measures, and the preservation of biodiversity, underscoring its significance in addressing the multifaceted challenges posed by wildfires.

Chapter 2
Literature Survey

2. LITERATURE SURVEY

[21] Y. Chen et at., proposed a CNN model specifically designed for image-based fire detection. Images of forested areas captured by UAVs were used as input data for CNN model. The CNN model was trained to distinguish between images containing fire and those without fire. The training process involved iteratively adjusting the CNN's parameters using optimization algorithms to minimize the prediction error. The CNN-based approach demonstrated promising results in accurately detecting forest fires from UAV-captured images.

[25] A. Viseras, M. Meissner, and J. Marchal, utilized Deep Q-learning, a form of reinforcement learning to train UAV agent to make decisions about their movement in real time. They developed a simulation environment that emulates wildfire scenarios, allowing the UAV agents learn and optimize their behavior through interaction with the simulated environment.The UAV agents selected actions based on the learned Q-values, aiming to maximize cumulative rewards. The Deep Q-learning-based approach enabled the UAVs to autonomously navigate and monitor wildfire fronts more efficiently compared to traditional methods.

[29] P. Ma, F. Yu, C. Zhou and M. Jiang, proposed CNN model for smoke detection in images, the images were preprocessed to enhance features relevant to smoke detection, such as contrast and edge information. Images containing scenes with potential smoke were acquired using cameras or other imaging devices. The integrated approach combining CNNs and image processing techniques demonstrated promising results in accurately detecting smoke in images.

[32] T. Gupta, H. Liu, and B. Bhanu, proposed Support Vector Machine (SVM) to train video frames containing scenes potentially affected by wildfire smoke were extracted from the video footage. The algorithm may utilize image processing techniques, machine learning models. The smoke detection algorithm was implemented to process video frames in real-time, enabling early detection of wildfire smoke and allowing authorities to initiate timely response measures.

[7] W. Li, S. Xiaobbo, C.Junn, and L. Ying, proposed an algorithm that extracts color features from images of forested areas using statistical methods. These features may include color histograms, color moments, or other statistical measures of color distribution within the image. The extracted color features are analyzed to identify patterns characteristic of forest fire occurrences. By analyzing color distributions, the algorithm can accurately detect the presence of fires in images, enabling early detection and response.

[10] W. Thomson, N. Bhowmik, and T. P. Breckon, proposed a compact and efficient CNN architecture tailored specifically for non-temporal fire detection. This architecture is optimized to minimize computational complexity and memory requirements while maintaining high detection accuracy. Special attention is given to optimizing the CNN architecture for real-time processing, ensuring that fire detection can be performed efficiently and rapidly without significant delays. The optimized CNN architecture enables real-time fire detection with

minimal processing delays, facilitating timely response and mitigation efforts in emergency situations.

[8] H. Tao and X. Lu, proposed a specialized 3D CNN architecture designed to analyze spatiotemporal features in video data. Unlike traditional 2D CNNs, which process individual frames independently, 3D CNNs capture both spatial and temporal information by considering sequences of frames. The CNN architecture employs parallel processing techniques to efficiently analyze video streams in real-time. he parallel processing approach enables the model to scale efficiently to handle large volumes of video data, making it suitable for deployment in systems requiring continuous monitoring of video streams.

[12] Y. Zhang and Y.Hu, proposed a CNN-based architecture tailored for analyzing video frames to detect the presence of smoke. The CNN model operates on individual frames extracted from video streams, treating each frame as an independent input. By analyzing spatial patterns within each frame, the model learns to identify visual cues associated with smoke, such as color, texture, and shape. The trained CNN model can be deployed for real-time inference on streaming video data, enabling the detection of smoke in video streams as they are captured.

Chapter 3

System Analysis & Feasibility Study

3. SYSTEM ANALYSIS & FEASIBILITY STUDY

3.1 Existing System

The escalating frequency and severity of forest fires pose a critical challenge to ecosystems and communities worldwide. With the 2020 forest fire occurrences ranking among the highest in scale and carbon emissions over the past two decades, the urgency to develop effective detection and mitigation strategies has never been greater. According to the World Health Organization, wildfires and volcanic activities have affected millions of people, leading to serious health implications such as respiratory problems and even fatalities. Moreover, wildfires disrupt essential services like transportation, communication, and power supply, while causing significant environmental and economic damage.

Traditional wildfire monitoring methods, primarily reliant on satellite imagery, face limitations in providing real-time detection due to operational constraints. Satellites in polar orbits offer daily image sets, while geostationary satellites provide hourly images, but neither ensures instantaneous monitoring. Various environmental factors further impede satellite-based detection, including small fire sizes, dense forest floors, cloud cover, and rapid fire occurrences between image captures. Consequently, there is an urgent need for innovative solutions to overcome these challenges and enhance wildfire detection capabilities.

In response to this, the novel approach on deep learning techniques for immediate wildfire and smoke detection. Leveraging convolutional neural networks (CNNs), the proposed EdgeFireSmoke model is tailored for seamless integration with edge devices such as closed-circuit television (CCTV) surveillance cameras and unmanned aerial vehicles (UAVs). By harnessing the power of CNNs, which automatically adjust filters during training to capture features crucial for wildfire detection, EdgeFireSmoke offers superior accuracy and efficiency compared to traditional methods.

EdgeFireSmoke offers a lightweight yet powerful solution for real-time wildfire detection. Moreover, its versatility allows for easy integration with existing surveillance systems, enabling prompt detection in both forested and urban areas.

3.1.1 Limitations of Existing system

- Its reliance on natural or artificial lighting conditions for effective operation, restricting its applicability to environments where sufficient illumination is available.
- The method's dependency on CNN architecture may pose challenges in scenarios where computational resources are limited or where edge devices lack the necessary processing power.

3.2 Proposed System

Detecting forest fires and smoke in real-time is critical for timely intervention and mitigation efforts. Traditional methods often struggle to capture the dynamic nature of wildfires, which evolve rapidly over time. To address this challenge, an innovative approach integrating Convolutional Recurrent Neural Network (CRNN) architecture has emerged as a promising solution.

CRNN architecture combines the strengths of convolutional and recurrent layers, enabling simultaneous analysis of spatial and temporal features in video data. Convolutional layers extract spatial information from frames, capturing essential visual cues such as color, shape, and texture. Meanwhile, recurrent layers, typically Long Short-Term Memory (LSTM) cells, capture temporal dependencies, recognizing patterns and changes over time.

The proposed model excel in capturing temporal information, crucial for identifying changes in fire intensity, size, and movement as they evolve. By analyzing sequences of frames, CRNNs offer superior sensitivity to the evolving dynamics of fires, enabling more accurate and timely detection.

Dropout regularization is employed to enhance model generalization and prevent overfitting, ensuring robust performance across diverse scenarios. This approach leverages advancements in deep learning and computer vision to provide an efficient and effective solution for real-time forest fire and smoke detection. By harnessing the power of deep learning and recurrent networks, this approach offers a promising tool for early detection and rapid response to forest fires, ultimately contributing to the preservation of ecosystems, property and lives.

3.2.1 Advantages of proposed system

- **Superior Performance:** CRNN models consistently achieve high accuracy rates, often surpassing conventional methods, with accuracy rates of up to 97.93%. This superior performance ensures reliable detection of forest fires, enhancing early intervention efforts.
- **Temporal Pattern Recognition:** The integration of LSTM layers enables CRNN models to understand temporal patterns in fire behavior. This capability allows for more accurate and timely detection of wildfires by capturing subtle changes and evolving dynamics over time.
- **Handling Long-range Dependencies:** CRNN architectures excel in handling long-range dependencies in video data, crucial for detecting forest fires. By effectively capturing spatial and temporal features, CRNN models can detect even subtle signs of fire activity, facilitating proactive intervention measures.
- **Scalability and Adaptability:** CRNN-based approaches offer scalability and adaptability, making the suitable for deployment in diverse environments and conditions. This flexibility allows for seamless integration with existing surveillance systems, including CCTV cameras and UAVs, enabling comprehensive monitoring of forested areas in real-time.

3.2.2 Dataset

UAV Images of Wildfires: A dataset to complete the conceptual model of wildfire and smoke detection from UAV images was built. In this dataset, public videos for fire detection from the UAVs were selected and processed to build the novel dataset. The videos recorded fire, smoke, burned areas, and green areas, mainly in forested areas, as well as in some urban areas. In total, 93 videos in the .mp4 format in an HD resolution of 1280 × 720 pixels were recorded. The saved videos were converted to .jpeg images, and some frames were stored and then divided into the following four classes.

1) Burned-area (IAQ): The IAQ class has 9348 images after a fire. In general, the floor is black and the trees are without leaves.

2) Fire-smoke (ICFF): The ICFF class has 15,579 images captured when a fire started, usually showing fire, smoke, and the location, such as forested areas and urban areas, with dry vegetation or pasture. The smoke can be white or dark.

3) Fog-area (ICN): The ICN class has 9762 images of a foggy environment.

4) Green-area (IAV): Finally, there are 14,763 images in the IAV class. This last class has images of the forest, dry vegetation, and pasture areas without any fire, smoke, or fog.

Table 3.1: Description of dataset with the classes: IAQ, ICFF, ICN and IAV

Groups	Images	Percent (%)	IAQ	ICFF	ICN	IAV
Training	9889	20	1870	3115	1952	2952
Validation	14830	30	2803	4672	2928	4427
Test	24711	50	4668	7784	4879	7380
Total	**49430**	**100**	**9341**	**15571**	**9759**	**14759**

burned-area(IAQ)

fire-smoke(ICFF)

fog-area(ICN)

Green-area(IAV)

Figure 3.1: Images from four classes

3.2.3 Data Pre-Processing

The data preprocessing pipeline for CRNN-based forest fire detection involves standardizing RGB images captured by surveillance cameras or UAVs through normalization and resizing to a uniform resolution, typically 224 × 224 pixels. For video data, frames are organized into temporal sequences to capture the temporal dynamics of fire behavior. Convolutional layers extract spatial features from the pre-processed images, while recurrent layers like Long Short-Term Memory (LSTM) layers capture temporal dependencies within the sequences. Data augmentation techniques may be applied to increase dataset diversity. Finally, additional normalization and scaling are applied to the extracted features to facilitate model training. Overall, these preprocessing steps ensure that input data are appropriately prepared for effective forest fire detection using CRNN models.

3.2.3.1 Image Normalization

In the preprocessing stage, normalization of input RGB images is crucial for maintaining consistency in pixel values across different images. This process involves scaling pixel values to a standardized range, commonly [0, 1] or [-1, 1]. By doing so, variations in pixel intensity are minimized, ensuring uniformity in data representation. Normalization optimizes the efficiency of subsequent processing by the neural network, as it helps prevent biases caused by differences in pixel scales. Ultimately, this standardized input enhances the model's ability to accurately capture and analyze spatial features during training and inference stages.

3.2.3.2 Resizing

Resizing is crucial to ensure uniformity in image dimensions, aligning them with the input specifications of the CRNN architecture. Typically, this involves adjusting images to a predefined size, such as 224 × 224 pixels, optimizing compatibility with convolutional layers. By standardizing image dimensions, the model can effectively process input data, extracting spatial features consistently across all images. This uniformity enhances the model's ability to capture relevant information during training and inference, contributing to more accurate forest fire detection. Ultimately, resizing facilitates seamless integration of input images into the CRNN framework, enabling efficient analysis of spatial patterns within the data.

3.2.3.3 Data Augmentation

Data augmentation is a critical preprocessing step aimed at improving the CRNN model's robustness and generalization ability. By applying transformations like rotations, flips, and shifts to the input data, the diversity of the training dataset is increased. This expanded dataset exposes the model to a broader range of variations and scenarios, enabling it to learn more effectively from the available data. As a result, the model becomes more adept at recognizing patterns and features associated with forest fires, enhancing its performance when deployed in real-world scenarios with unseen data or environmental conditions**.**

3.2.3.4 Normalization and Scaling

Normalization and scaling of extracted features further standardize their values, aiding efficient model training. By bringing features to a consistent range, the neural network can more effectively learn from the data, preventing issues like vanishing or exploding gradients during training. This preprocessing step ensures that the model can accurately interpret and weigh different features, leading to improved performance and generalization when deployed for forest fire detection.

Table 3.2 Data Pre-Processing Values

Parameter	**Value**
Rescale	1./255
Rotation Range	40
Width Shift Range	0.2
Height Shift Range	0.2
Shear Range	0.2
Zoom Range	0.2
Horizontal Flip	True
Fill Mode	nearest

3.3 Methodology

In forest fire detection, the CRNN (Convolutional Recurrent Neural Network) architecture stands out as an innovative approach that integrates convolutional and recurrent layers to tackle the challenges of spatial and temporal analysis in video data. At the heart of CRNN lies the Long Short-Term Memory (LSTM) architecture, a type of recurrent neural network specifically designed to capture temporal dependencies in sequential data.

LSTM networks consist of memory cells equipped with gating mechanisms that regulate the flow of information. These mechanisms include the input gate, forget gate, and output gate, each responsible for controlling the flow of input data, previous cell state, and output information, respectively. This sophisticated architecture allows LSTM cells to retain important information over long sequences, making them particularly well-suited for modeling temporal patterns in videos.

The integration of LSTM layers within the CRNN framework enables the model to understand the dynamic nature of fire behavior over time. Convolutional layers preceding the LSTM layers extract spatial features from individual frames of video data, such as color, texture, and shape, providing crucial visual cues for fire detection. These spatial features are then fed into the LSTM layers, where they are processed over sequential frames to capture temporal patterns in fire intensity, size, and movement.

Using LSTM layers in the CRNN architecture is their ability to effectively handle long-range dependencies in video data. By maintaining a memory of past observations, LSTM cells can capture subtle changes and trends in fire behavior that may span multiple frames. This capability is essential for accurately detecting evolving fire dynamics in real-time scenarios.

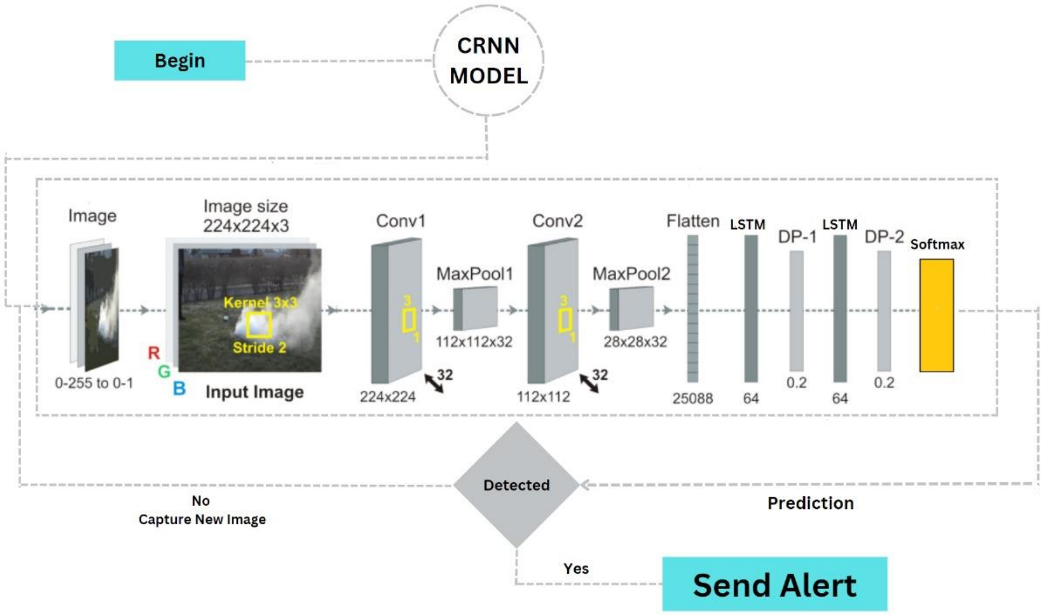

Figure 3.2 Working of CRNN Architecture

1. Embedding Layer: This layer transformed the input text data into dense vector representations, mapping each word to a high-dimensional embedding space. The embedding layer limited the number of distinct words or tokens it could process to 100,000.
2. LSTM Layer: It is a sequel of Recurrent Neural Networks (RNNs), and was responsible for handling the sequential nature of the input data and capturing long-range dependencies within the text, and a dropout rate of 0.5 was applied to mitigate overfitting.
3. Dense Output Layer: The last layer was a dense layer with a SoftMax activation function, which produced the output probabilities for the three sentiment classes (positive, negative, and neutral).

Table 3.3 Experimental Setup of LSTM Model

Embedding Layer	Size=100, max_features=100000
LSTM Layer	Layer=2, units=64, dropout=0.2
Dense Layer	Units=3, activation=SoftMax

Need for Activation Function

If an activation function is not used in a neural network, then the output signal would simply be a simple linear function which is just a polynomial of degree one [37]. Although a linear equation is simple and easy to solve but their complexity is limited, and they do not have the ability to learn and recognize complex mappings from data. Neural Network without an activation function acts as a Linear Regression Model with limited performance and power most of the time. It is desirable that a neural network not only learn and compute a linear function but perform tasks more complicated than that like modelling complicated types of data such as images, videos, audio, speech, text, etc. This is the reason that activation functions and artificial neural network techniques like Deep Learning are used, as they make sense of complicated, high dimensional and nonlinear datasets where the model has multiple hidden layers.

3.4 Model Training and Testing

The model, trained on a custom dataset extracted from high-definition videos, demonstrates remarkable proficiency in analyzing real-time video footage. When presented with footage of a forest fire, it efficiently identifies the dynamic patterns of flames and smoke, offering insights into the severity and spread of the blaze. Additionally, when tasked with analyzing videos depicting serene forest scenes adorned with lush greenery and shrouded in fog, the model adeptly discerns the intricate details of the landscape, from the dense foliage to the ethereal mist. Its ability to discern between the contrasting scenarios of devastation and tranquility showcases its versatility and reliability in interpreting complex visual data, making it an invaluable tool for monitoring and managing natural environments.

3.5 Evaluation Metrics

Four frequently employed evaluation metrics were used to compare the performances of the proposed classification model and the compared models: “Accuracy, Precision, Recall, and F1- score.”

1. Accuracy: Accuracy measures the overall correctness of the predictions made by the model. It is calculated as the ratio of the number of correct predictions to the total number of predictions.

$$Accuracy = \frac{TP + TN}{FP + FN + TP + TN}$$

2. Precision: Precision measures the proportion of correctly predicted positive instances (True Positives) out of all instances predicted as positive (True Positives + False Positives). It indicates how many of the predicted positive instances are positive.

$$Precision = \frac{TP}{TP + FP}$$

3. Recall: Recall, also known as sensitivity or true positive rate, measures the proportion of correctly predicted positive instances (True Positives) out of all actual positive instances (True

Positives + False Negatives). It indicates how many of the actual positive instances are correctly predicted by the model.

$$Recall = \frac{TP}{TP + FN}$$

4. F1-score: F1-score is the harmonic mean of Precision and Recall. It provides a single metric that balances both Precision and Recall.

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

In addition to these metrics, the text mentions using accuracy-loss curves for training and test validation to verify the performance of the proposed model. These curves can help visualize how the accuracy of the model changes as the loss function decreases during training, indicating how well the model is learning from the data.

3.6 Feasibility Study

3.6.1 Economic Feasibility

From an economic standpoint, the development of the forest fire detection model is feasible, considering the potential benefits it can provide to businesses in the tourism and hospitality industry. By accurately classifying the fire and smoke in the images, the model can help businesses make informed decisions to improve customer satisfaction and loyalty, ultimately leading to increased revenue and profitability. The cost of developing and implementing the model is expected to be outweighed by the benefits it brings in terms of improved business performance and customer experience.

3.6.2 Operational Feasibility

The operational feasibility of the project is also high, as the forest fire detection model can be seamlessly integrated into existing systems or used as a standalone tool. The model can be deployed on cloud platforms, making it easily accessible and scalable. Once deployed, the model can continuously analyze the fires, smokes in the forest, providing real-time insights to authorities. This operational efficiency can help businesses quickly respond to customer feedback and improve overall customer satisfaction.

3.6.3 Technical Feasibility

The technical feasibility of the project is high, given the availability of advanced machine learning libraries and frameworks such as TensorFlow and Keras. These tools provide the necessary functionality to develop and train complex models like LSTM for forest fire detection.

Chapter 4
System Requirements

4. SYSTEM REQUIREMENTS

The project involved analyzing the design of few applications so as to make the application more user friendly. To do so, it was really important to keep the navigations from one screen to the other well-ordered and at the same time reducing the amount of typing the user needs to do. To enhance accessibility, the application was adapted for compatibility with Python IDE, ensuring wider accessibility and ease of use for developers.

4.1 Functional Requirements

Graphical User Interface: A graphics-based operating system interface that uses icons, menus and a mouse (to click on the icon or pull down the menus) to manage interaction with the system.

4.2 Technologies and Languages used to develop

1. Python: Python is a versatile, high-level programming language known for its simplicity and readability. It offers extensive libraries and frameworks for various applications, including web development, data analysis, artificial intelligence, and more.

2. Deep Learning: Deep Learning is a subset of machine learning focused on learning representations of data using artificial neural networks with multiple layers. It enables computers to learn from large amounts of data and make complex decisions, leading to advancements in tasks like image recognition, natural language processing, and autonomous driving.

3. HTML & CSS: Web technologies used for creating user interfaces. They facilitate the construction of web pages that users interact with. In the backend, Python code executes to provide predictions.

4.2.1 Debugger and Emulator

1.PyCharm: PyCharm is an integrated development environment (IDE) utilized for building deep learning models and Flask applications. It offers robust debugging features essential for identifying and resolving coding errors.

2.Google Chrome: Google Chrome is a popular web browser renowned for its compatibility and efficiency in rendering web-based applications. It provides a seamless experience for accessing the internet and running web pages.

4.2.2 Hardware Requirements

- Computer with 1.6 GHz or faster processor
- Minimum 4 GB of RAM or more
- 2.5 GB of available hard-disk

4.2.3 Software Requirements

- Operating Systems: Windows 11
- Workspace Editor: PyCharm
- Backend- Python 3.10

Chapter 5
Design

5. DESIGN

5.1 System Design

The proposed system uses a Long Short-Term Memory (LSTM) model to analyze the UAV Images of Wildfire dataset as shown in Figure 5.1. It starts by preprocessing the text, extracting features, and training the model. Cross-validation techniques evaluate the model's performance. The trained model predicts fire, smoke, fog and green area on new or unseen data. External libraries like Flask, OpenCV, NumPy, TensorFlow and scikit-learn.

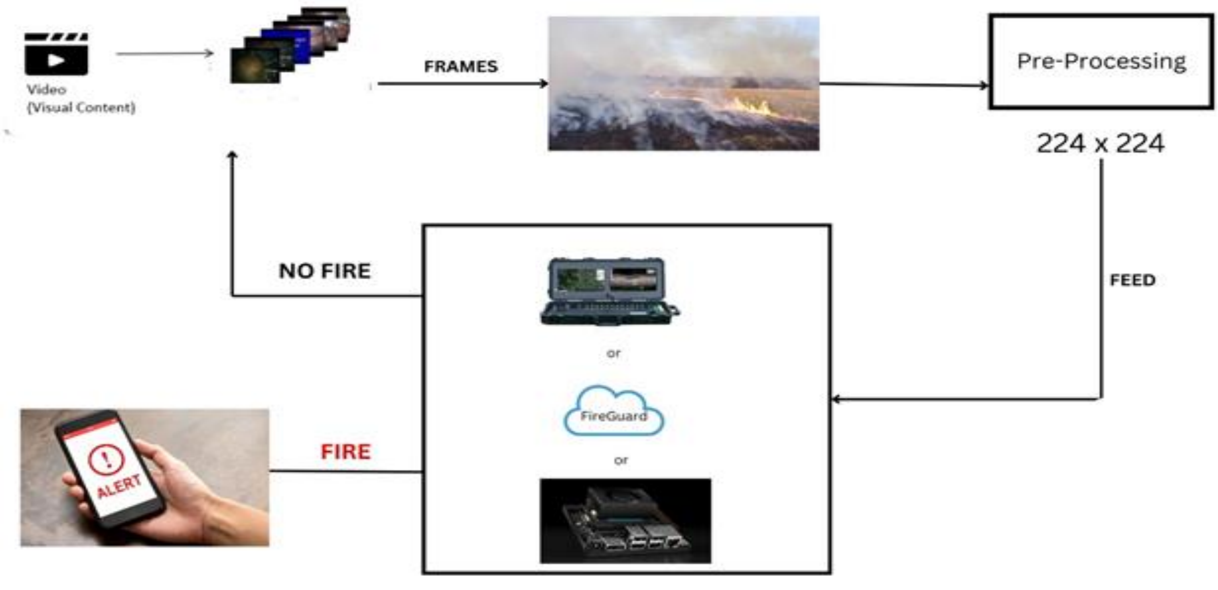

Figure 5.1: Work flow of CRNN Model

This image presents a flow diagram illustrating the process of building a forest fire detection model using deep learning techniques. The process starts with loading a dataset named UAV Images of Wildfire. These images or videos are subsequently converted into frames, setting the groundwork for analysis. Following this, the frames undergo pre-processing to enhance quality and extract relevant features. The model is then trained using these processed frames, facilitating the identification of distinct elements within the region of interest: fire, smoke, fog, or green areas. The culmination of this process results in the model outputting among fog, smoke, fire, or green area, thereby providing critical insights for wildfire monitoring and management. The overall process showcases the typical steps involved in building a Fireguard model for forest fire detection.

5.2 UML Diagrams

UML, or Unified Modeling Language, is a standardized modeling language used in software engineering to visually represent software systems. Its importance lies in providing a common language and notation for software developers, designers, and stakeholders to communicate and understand the structure, behavior, and interactions of complex systems. UML diagrams such as class diagrams, sequence diagrams, and use case diagrams help in conceptualizing, designing, documenting, and communicating software systems, leading to better understanding, collaboration, and more efficient development processes.

Unified Modeling Language (UML) diagrams are a standardized way of visually representing software systems. They provide a way for software developers to communicate system designs, architectures, and processes in a clear and consistent manner. UML diagrams use various graphical elements such as boxes, lines, and arrows to represent different aspects of a system, making it easier for stakeholders to understand complex systems.

One of the key benefits of UML diagrams is that they help in the visualization of the system's architecture and design. By using different types of diagrams such as class diagrams, sequence diagrams, and use case diagrams, developers can create a comprehensive picture of the system, which can be used as a blueprint for implementation.

Another important aspect of UML diagrams is that they help in the communication between different stakeholders involved in the software development process. For example, developers can use UML diagrams to explain their designs to non-technical stakeholders such as project managers or clients, helping them to understand the system requirements and functionalities.

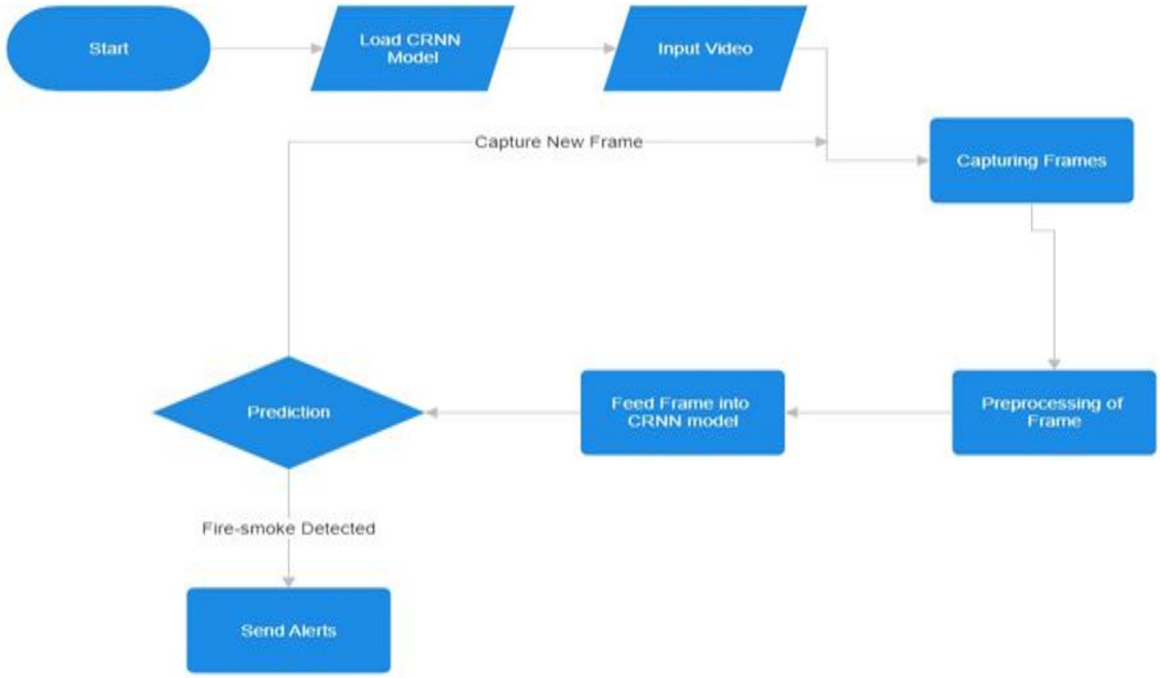

Figure 5.2: Flow Diagram

Chapter 6
Implementation

6. IMPLEMENTATION

6.1 CRNN Model Algorithm:

ALGORITHM build_compile_fit_CRNN_model(train_generator, validation_generator, epochs):

CREATE model

ADD Conv2D layer with 32 filters, kernel size (3, 3), stride (1, 1), padding 'same', activation 'relu', input shape (224, 224, 3) to model

ADD MaxPooling2D layer with pool size (2, 2), stride (2, 2), padding 'valid' to model

ADD Conv2D layer with 32 filters, kernel size (3, 3), stride (1, 1), padding 'same', activation 'relu' to model

ADD MaxPooling2D layer with pool size (2, 2), stride (2, 2), padding 'valid' to model

ADD TimeDistributed layer with Flatten layer to model

ADD LSTM layer with 64 units, return sequences True to model

ADD Dropout layer with rate 0.2 to model

ADD LSTM layer with 64 units to model

ADD Dropout layer with rate 0.2 to model

ADD Dense layer with 4 units, activation 'softmax' to model

COMPILE model with:

- Optimizer: Adam with learning rate
- Loss: Categorical Crossentropy
- Metrics: Accuracy

FIT model using:

- Training generator: train_generator
- Validation data: validation_generator
- Number of epochs: epochs

RETURN trained model

6.2 Home Page:

<!DOCTYPE html>

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>FireGuard - Video Streaming</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-image:url("static/back1.jpg");
          background-repeat:no-repeat;
          background-position:down;
          background-size:cover;
        }
        h1 {
            text-align: left;
            margin-bottom: 20px;
            font-size: 36px;
            color:crimson;
            text-shadow:5px 1px black;
            margin-left:20px;
        }
        form {
            text-align: center;
            margin-bottom: 20px;
        }

        select {
            padding: 10px;
```

```
    font-size: 16px;
    border: 1px solid #ccc;
    border-radius: 4px;
}
 input[type="submit"],
input[type="button"],
.testing-btn {
    padding: 10px 20px;
    font-size: 16px;
    background-color: #28a745; /* Green color */
    color: #fff;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    margin-right: 10px;
    margin-bottom: 10px; /* Added margin bottom for spacing */
}
input[type="submit"]:hover,
input[type="button"]:hover,
.testing-btn:hover {
    background-color: #218838; /* Darker green on hover */
}
 img {
    display: block;
    margin: 20px auto;
    max-width: 100%;
    height: auto;  }

.btn-container {
    text-align: right; /* Center align buttons */
```

```
    margin-top: 20px;
 }
.btn-container2 {
    text-align: center; /* Center align buttons */
    margin-top: 20px;
 }
.btn-container button {
    padding: 10px 20px;
    font-size: 16px;
    background-color: #007bff;
    color: #fff;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    margin-right: 10px;
}
btn-container button:hover {
    background-color: #0056b3;
}
 /* Alert box for fire detection */
.alert {
    position: fixed;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    background-color: red;
    color: white;
    padding: 20px;
    border-radius: 8px;
    font-size: 24px;
```

```
            display: none;
            z-index: 9999; /* Ensure the alert is on top of other elements */
        }
        .testing-btn{
        button-align:center;
       }
    </style>
</head>
<body>

        <h1>FireGuard - Video Streaming</h1>
        <div class="btn-container">
            <button>Home</button>
            <button>About</button>
            <button>Services</button>
        </div>
        <form id="videoForm" action="{{ url_for('process') }}" method="post">
            <div class="btn-container2">
                <select name="video_choice">
                    <option value="1">Fire</option>
                    <option value="2">Green</option>
                    <option value="3">Fog</option>
                    <option value="4">Webcam</option>
                </select>
                <button class="testing-btn" type="submit">Start</button>
                <button class="testing-btn" onclick="stopVideo()">Stop</button>
            </div>
        </form>
        <br>
        <img id="videoFeed" src="" alt="Video Feed">
```

```
        <!-- Alert box for fire detection -->
        <div id="fireAlert" class="alert">Fire Detected!</div>
 <script>
        function startVideo(videoChoice) {
            var img = document.getElementById("videoFeed");
            img.src = "/video_feed?video_choice=" + videoChoice;
            // Start checking for fire-smoke detection
            checkFireSmoke();
        }
     function checkFireSmoke() {
    // Fetch the video feed
    fetch("/video_feed?video_choice=" + document.getElementById("video_choice").value)
        .then(response => response.text())
        .then(data => {
            if (data.trim() === "fire-smoke") {
                fireDetected();
            }
            // Continue checking for fire-smoke detection recursively
            checkFireSmoke();
        })
        .catch(error => {
            console.error('Error:', error);
        });
}
// Function to show alert and play sound for fire detection
function fireDetected() {
    console.log("Fire detected!");
    var alertBox = document.getElementById("fireAlert");
    alertBox.style.display = "block";
    // Play a loud sound
```

```
    var audio = new Audio('static/sound.mp3');
    audio.play();
}
     // Function to show alert and play sound for fire detection
       function fireDetected() {
           console.log("Fire detected!");
           var alertBox = document.getElementById("fireAlert");
           alertBox.style.display = "block";
           // Play a loud sound
           var audio = new Audio('static/sound.mp3');
           audio.play();
       }
    </script>
</body>
</html>
```

Chapter 7

Results

7. RESULTS

The model is compared with the existing CNN model and observed that the proposed model achieves more accuracy, F1-Score and R1 values as you can see in table 7.1. The proposed method achieves an accuracy of 97.93%, the F1-score of 97.93%, the precision of 97.94%, and the recall of 97.93%.

Table 7.1 Results CNN vs CRNN

Model	Accuracy	Precision	Recall	F1 Score
CNN	95.44%	95.54%	95.44%	95.45%
CRNN	97.93%	97.94%	97.93%	97.93%

In the confusion matrix of Fig 7.1, the proposed method shows a better ability to classify all classes and in Fig 7.2 you can find the existing method confusion matrix.

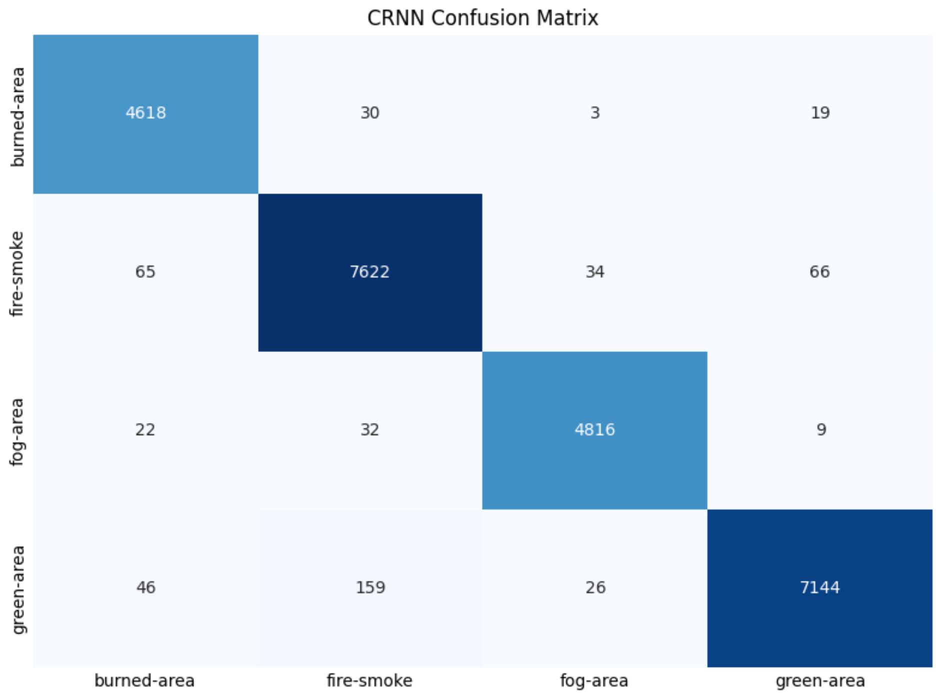

Figure 7.1 Confusion matrix of Proposed method

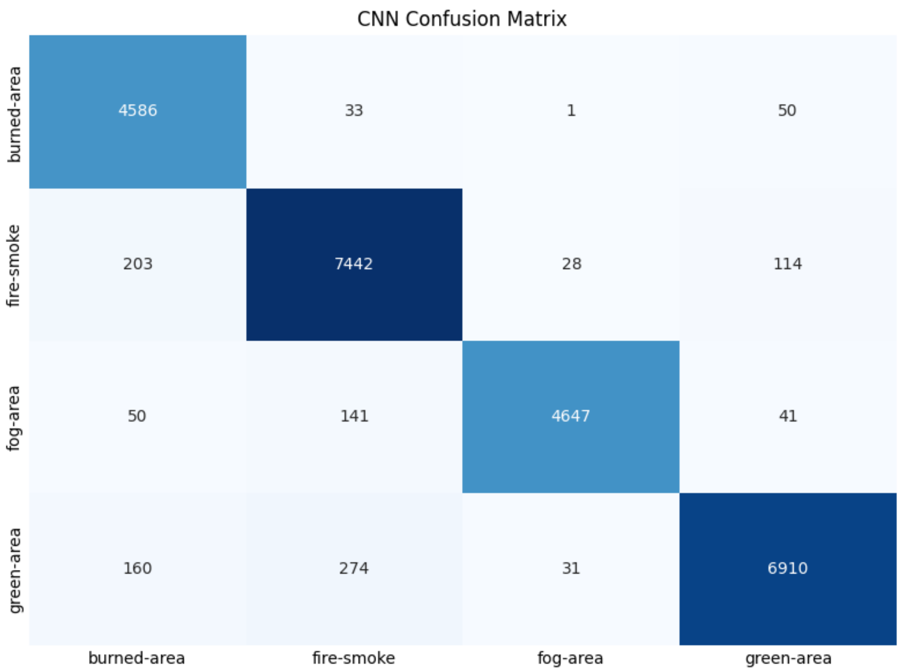

Fig 7.2 Confusion matrix of Existing method

Fig 7.3 shows that the training of the CRNN method has happened more accurately compared to the CNN method. The CRNN training accuracy curve is higher than CNN in all the epochs.

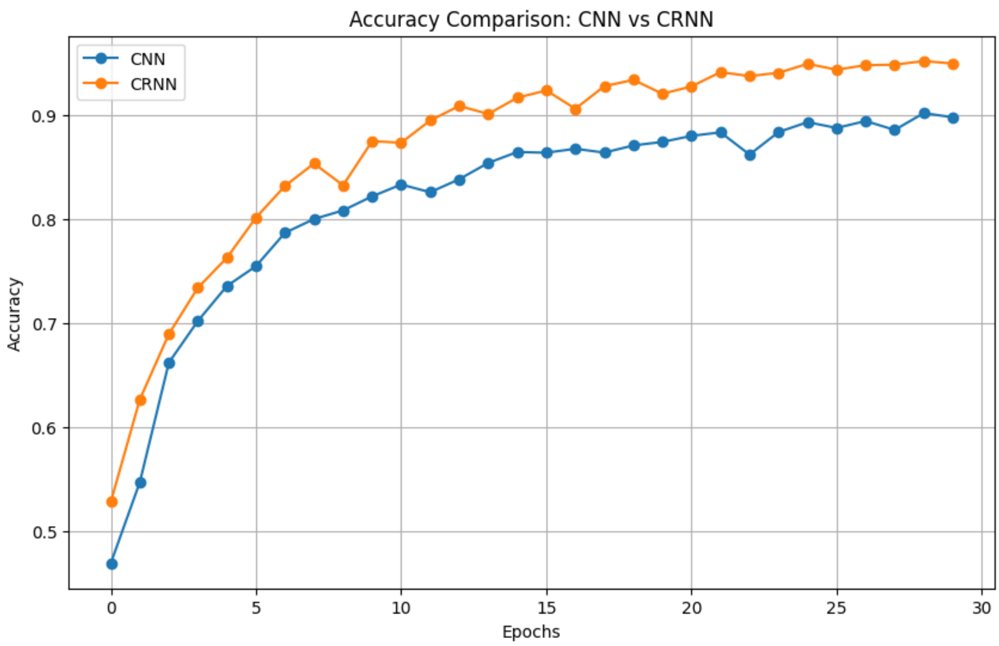

Fig 7.3 Training Accuracy Comparison between CNN and CRNN

Chapter 8

Social Impact

8. SOCIAL IMPACT

In the realm of modern commercial landscapes, leveraging advanced technology like the CRNN model for fire detection from UAV Images of wildfire holds immense importance. By analyzing these images, organizations can identify potential fire outbreaks swiftly, aiding in timely interventions and minimizing damage. This proactive approach not only enhances safety measures but also mitigates risks associated with wildfires, safeguarding both lives and properties.

The implementation of a cross-validated CRNN model for fire detection offers multifaceted benefits. Firstly, it enables organizations to detect fires in their early stages, allowing for prompt response and containment efforts. Secondly, this technology aids in optimizing resource allocation, ensuring that firefighting resources are deployed efficiently and effectively. Lastly, it contributes to enhancing public safety by providing reliable tools for wildfire monitoring and management.

Through the utilization of CRNN models for fire detection, the commercial sector can significantly impact societal well-being. By swiftly identifying and responding to wildfire threats, organizations contribute to the preservation of natural landscapes and the protection of communities residing in high-risk areas. Moreover, by investing in advanced technologies for fire detection, businesses demonstrate a commitment to corporate social responsibility and sustainable practices, fostering trust and goodwill among stakeholders.

- **Enhanced Emergency Response:** Utilizing CRNN models for wildfire detection enables proactive measures akin to improving customer experiences in the tourism industry. By analyzing UAV Images, authorities gain critical insights, facilitating swift responses to potential fire outbreaks and minimizing risks to lives and properties.

- **Transparency and Accountability in Disaster Management:** The adoption of CRNN models promotes transparency and accountability in wildfire management, akin to the tourism sector's emphasis on customer satisfaction. Through comprehensive analysis of imagery datasets, authorities can identify fire-prone areas and allocate resources efficiently, fostering community trust and safety.

- **Economic Resilience and Sustainability:** The economic impact of CRNN-based wildfire detection parallels the tourism industry's focus on revenue generation and sustainability. Early detection and containment of wildfires mitigate financial losses, preserve natural resources, and promote long-term economic prosperity for businesses and communities alike.

- **Proactive Risk Mitigation:** CRNN models empower early detection of wildfire outbreaks, akin to addressing customer feedback to enhance experiences. Analyzing UAV Images allows for swift responses, minimizing risks to lives and properties.

- **Efficient Resource Allocation:** Similar to optimizing services based on customer preferences, CRNN analysis aids in allocating firefighting resources effectively. Authorities pinpoint fire-prone areas, ensuring resources are deployed where needed most, enhancing overall effectiveness.

In summary, utilizing CRNN models for wildfire detection from UAV Images offers significant social impacts. Comparable to customer review analysis, it enhances emergency responses, fosters community engagement, and informs decision-making. This proactive approach empowers authorities to safeguard lives, properties, and ecosystems, ensuring effective wildfire management and community resilience.

Chapter 9

Conclusion & Future Work

9. CONCLUSION & FUTURE WORK

In conclusion, the FireGuard system, built upon a Convolutional Recurrent Neural Network (CRNN) architecture, represents a significant advancement in forest fire detection technology. Through the integration of deep learning techniques and real-time video analysis, FireGuard offers a proactive solution to the escalating threat of wildfires, particularly in forested and remote areas.

The system's ability to analyze live video streams from aerial vehicles or webcams enables timely detection of fire and burned areas, facilitating rapid response and mitigation efforts. By leveraging CRNNs, FireGuard captures both spatial and temporal features within video data, enhancing detection accuracy and reliability.

The experimental results validate the effectiveness of the CRNN model, demonstrating high accuracy rates and performance metrics compared to traditional CNN-based approaches. Moreover, the system's lightweight architecture and compatibility with edge devices ensure scalability and adaptability for diverse deployment scenarios.

FireGuard presents a robust and efficient solution to the pressing challenge of forest fire detection, offering significant benefits in terms of environmental conservation, public safety, and resource management. With further advancements and integration into existing wildfire management systems, FireGuard has the potential to revolutionize how wildfires are detected, monitored, and mitigated, contributing to a safer and more sustainable future.

- **Enhancing Model Architecture:** Future work could focus on refining the CRNN model architecture by adding additional LSTM layers and Conv2D layers. This could potentially improve the model's accuracy in detecting wildfires from UAV Images by capturing more intricate patterns and features in the data.
- **Exploring Advanced Techniques:** Further research could investigate the integration of advanced techniques such as attention mechanisms or ensemble learning to enhance the model's performance. These techniques could improve the model's ability to discern relevant features in the imagery datasets, leading to more accurate wildfire detection.
- **Mobile Application Development:** Developing a mobile application could extend the project's utility by providing real-time wildfire alerts to users. By integrating the trained CRNN model into the app, individuals in wildfire-prone areas could receive timely notifications, enabling them to take necessary precautions and evacuate if needed.
- **Geospatial Integration:** Future work could explore integrating geospatial data into the wildfire detection system. By incorporating information such as weather conditions, terrain elevation, and vegetation density, the model could generate more precise wildfire risk assessments, enhancing the effectiveness of early warning systems.
- **Crowdsourced Data Collection:** Leveraging crowdsourced data collection methods could enrich the project's dataset and improve model generalization. Encouraging users to contribute UAV Images of wildfire incidents from various locations could enable the model to learn from a broader range of scenarios, enhancing its robustness and accuracy in detecting wildfires.

BIBLIOGRAPHY

[1] Y. Chen et al., “UAV image-based forest fire detection approach using convolutional neural network,” in Proc. 14th IEEE Conf. Ind. Electron. Appl., 2019, pp. 2118–2123.

[2] A. Viseras, M. Meissner, and J. Marchal, “Wildfire front monitoring with multiple UAVs using deep Q-learning,” IEEE Access, to be published, doi: 10.1109/ACCESS.2021.3055651.

[3] P. Ma, F. Yu, C. Zhou, and M. Jiang, “An integrated smoke detection method based on convolutional neural network and image processing,” in Proc. IEEE 8th Int. Conf. Comput. Sci. Netw. Technol., 2020, pp. 32–36.

[4] T. Gupta, H. Liu, and B. Bhanu, “Early wildfire smoke detection in videos,” in Proc. 25th Int. Conf. Pattern Recognit., 2021, pp. 8523–8530.

[5] W. Li, S. Xiaobo, C. Jun, and L. Ying, “Research on forest fire image recognition algorithm based on color feature statistics,” in Proc. 6th Int. Conf. Intell. Comput. Signal Process., 2021, pp. 346–349.

[6] W. Thomson, N. Bhowmik, and T. P. Breckon, “Efficient and compact convolutional neural network architectures for non-temporal real-time fire detection,” in Proc. 19th IEEE Int. Conf. Mach. Learn. Appl., 2020, pp. 136–141.

[7] X. Li, Z. Chen, Q. J. Wu, and C. Liu, “3D parallel fully convolutional networks for real-time video wildfire smoke detection,” IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 1, pp. 89–103, Jan. 2020.

[8] Y. Zhang and Y. Hu, “Video smoke detection based on convolution neural network,” in Proc. Int. Comput., Signals Syst. Conf., 2018, pp. 123–127.

[9] Brazilian Institute of Environment and Renewable Natural Resources, “Burn monitoring in satellite images,” 2017. [Online]. Available: http://www.ibama.gov.br/consultas/incendios-florestais/consultasmonitoramento-de-queimadas/monitoramento-de-focos-de-queimadasem-imagens-de-satelites.

[10] J. Shi, W. Wang, Y. Gao, and N. Yu, “Optimal placement and intelligent smoke detection algorithm for wildfire-monitoring cameras,” IEEE Access, vol. 8, pp. 72326–72339, 2020.

[11] X. Shi, N. Lu, and Z. Cui, “Smoke detection based on dark channel and convolutional neural networks,” in Proc. IEEE 5th Int. Conf. Big Data Inf. Analytics, 2019, pp. 23–28.

[12] H. Ullah, K. Muhammad, M. Irfan, B. Z. Muhammad, and S. Anwar, “Light-DehazeNet: A novel lightweight CNN architecture for single image dehazing,” IEEE Trans. Image Process., vol. 30, pp. 8968–8982, Oct. 2021.

[13] O. M. Bushnaq, A. Chaaban, and T. Y. Al-Naffouri, “The role of UAV-IoT networks in future wildfire detection,” IEEE Internet Things J., vol. 8, no. 23, pp. 16984–16999, Dec. 2021.

[14] S. Dutta and S. Ghosh, "Forest fire detection using combined architecture of separable convolution and image processing," in Proc. 1st Int. Conf. Artif. Intell. Data Analytics, 2021, pp. 36–41.

[15] K. Shrestha, R. Dubey, A. Singandhupe, S. Louis, and H. La, "Multi objective UAV network deployment for dynamic fire coverage," in Proc. IEEE Congr. Evol. Comput., 2021, pp. 1280–1287.

[16] J. Jeyavel, A. A. Prasad, K. M. Shelke, P. D. Sargade, and U. V. Thoke, "Survey on fire fighting techniques using unmanned aerial vehicles," in Proc. Int. Conf. Adv. Comput. Innov. Technol. Eng., 2021, pp. 239–241.

[17] S. Khan, K. Muhammad, S. Mumtaz, S. W. Baik, and V. H. C. de Albuquerque, "Energy-efficient deep CNN for smoke detection in foggy IoT environment," IEEE Internet Things J., vol. 6, no. 6, pp. 9237–9245, Dec. 2019.