Real-Time Wildfire Progression Analysis and Prediction using Hybrid Model

MSc Research Project
MSc Data Analytics

Shraddha Eknath Gargote
Student ID: X23106263

School of Computing
National College of Ireland

1 Supervisor: Dr Catherine Mulwa

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Shraddha Eknath Gargote
Student ID:	X23106263
Programme:	MSc Data Analytics — **Year:** 2023-2024
Module:	MSc Research Project
Supervisor:	Dr Catherine Mulwa
Submission Due Date:	2nd September 2024
Project Title:	Real-Time Wildfire Progression Analysis and Prediction using Hybrid Model
Word Count:	***7649*** — **Page Count: 28**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Shraddha Eknath Gargote

Date: 31/08/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	Yes
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	Yes
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	Yes

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Real-Time Wildfire Progression Analysis and Prediction using Hybrid Model

Shraddha Ekath Gargote

X23106263

Abstract

The wildfire identification and propagation prediction is a necessary inclusion that allows for the realistic projection of the fire. Wildfires may cause drought, loss of vegetation, loss of property, rise in greenhouse gases, and loss of a wide range of wildlife and human lives too. To reduce this risk, there should be a mechanism where we can detect the fire in the initial stage of raging and predict its progression towards a wildfire to control its movement and stop it. Most of the conventional methods use machine learning models to predict the presence of a fire. This doesn't help much; we need to track the propagation path of the fire to save more lives and stocks. Hence, to achieve this, the proposed model utilizes a support vector machine for the classification of the fire in real-time images grabbed from NASA's repository for the given latitude and longitude. Thereafter, the YOLO model is used to detect the wildfire in the obtained satellite image. Finally, the deployment of the LSTM model enables us to provide the progression track of the wildfire.

1 Introduction

Rapid urbanization has made deforestation one of humanity's greatest concerns, resulting in the loss of one football field of forest every second, 24/7. Uncontrollable wildfires, in addition to other factors, significantly contribute to deforestation. These wildfires may have been caused by burning debris, downed power lines, equipment malfunction, intentional and wrongful acts, discarded cigarettes, unattended campfires, friction among trees, and a variety of other factors. Most of the time, this wildfire is so intense that it easily swallows millions of acres of forest fire within a few days.

1.1 Motivation and Research Background

The United States of America is the most wildfire-prone country, with all the possible causes mentioned above. According to the data from the National Interagency Fire Center (NIFC), around 70,000 wildfire incidents occurred every year between 1983 and 2021. In these years, these wildfire incidents almost gutted over five million acres of forest land. Therefore, protecting our planet Earth is of utmost importance, as there is no other planet B for human habitation. The table below displays the data obtained from NIFC, while table 1 and Figure 1 below display the subsequent graph.

Table 1: Statistical data of NIFC for wildfire incidents

Year	No of Fires	Acres Burnt	Year	No of Fires	Acres Burnt
2021	58985	7125643	2002	73457	71
2020	58950	10122336	2001	84079	35
2019	50477	4664364	2000	92250	73
2018	58083	8767492	1999	92487	56
2017	71499	10026086	1998	81043	13
2016	67743	5509995	1997	66196	28
2015	68151	10125149	1996	96363	60
2014	63312	3595613	1995	82234	18
2013	47579	4319546	1994	79107	40
2012	67774	9326238	1993	58810	17
2011	74126	8711367	1992	87394	20
2010	71971	3422724	1991	75754	29
2009	78792	5921786	1990	66481	46
2008	78979	5292468	1989	48949	18
2007	85705	9328045	1988	72750	50
2006	96385	9873745	1987	71300	24
2005	66753	8689389	1986	85907	27
2004	65461	8097880	1985	82591	28
2003	63629	3960842	1984	20493	11
			1983	18229	13

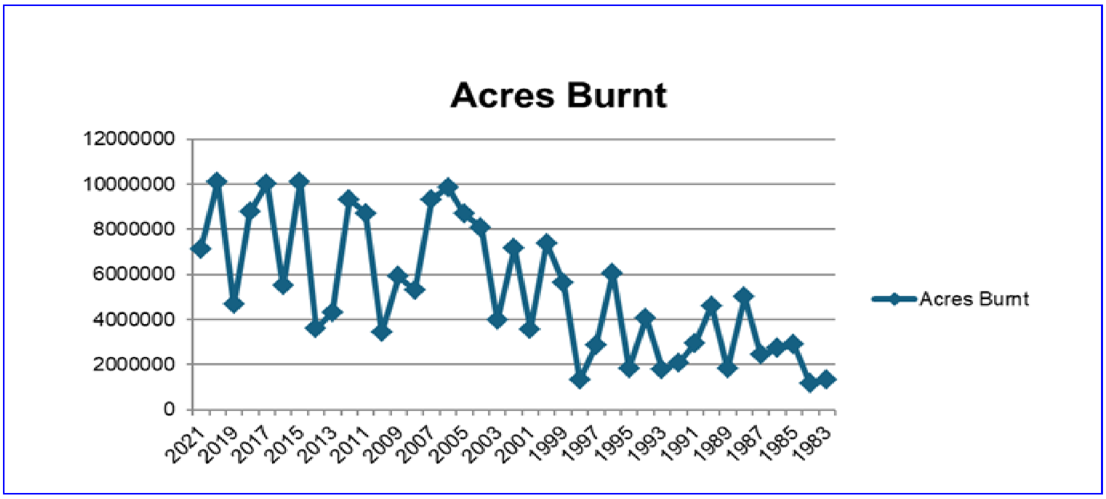

Figure 1: Number of acres burned v/s the year

Table 1 shows the number of fire incidents in each year and the area burned due to fires in terms of acres. This data was used to create a trend of the number of acres burned due to wildfires and the number of fire incidents over the years. The data demonstrates that, while the frequency of fires appears to be stable, the amount of land consumed by wildfires has been rising over time. This makes it quite evident that the failure to limit wildfires and the delay in responding to them may be to blame for the increasing number of acres destroyed. This is being exacerbated by the fact that the weather is changing, and the temperature is going up due to global warming and other ecological imbalances in the environment.

Therefore, in order to prevent wildfires from destroying the forest, various strategies, such as surface clearing, and aerial fire extinguishing, are employed. The most important of all of these is to identify the wildfire as soon as possible from the satellite images and predict its path of propagation. This allows for the efficient detection of the early spread and the implementation of necessary control measures. Many techniques are deployed to achieve this, like the satellite imagery process, information retrieval, and many more. But to achieve higher accuracy in the process, a deep learning mechanism is required with a hybrid approach.

The hybrid approach entails the deployment of one or more deep learning models to improve wildfire detection and propagation. Therefore, we design a system that retrieves a satellite image from NASA's repository, using an interactive desktop application that utilizes the tkinter library of the Python programming language. After retrieving the image from NASA's repository, a support vector machine classifies it for the presence of fire. After classifying the image as fire-related, we subject it to a YOLO neural network model to pinpoint the fire's location. Once we identify the fire's location, we use the vegetation dataset at that location to measure the wind speed, altitude, and other parameters for the LSTM neural network model. We utilize the LSTM prediction values to determine the wildfire's propagation path and display the outcome in a gif format image.

1.2 Research Question, Objective, and Contribution:

Increasing wildfire incidents pose the greatest threat to human existence, consuming millions of hectares of forest annually. Monitoring wildfires and predicting their spread is the urgent need to curb this menace. Finger counting methodologies, however, offer all-in-one solutions for wildfire classification, detection, and tracking propagation using hybrid deep learning models.

RQ: *"How well a hybrid deep learning model (SVM, YoLO, and LSTM) trained on a dataset of wildfire images and vegetation may aid fire and forest departments in detecting wildfires and determining their propagation paths using images stored in NASA's repository."*

Sub RQ: "Will fire classification, detection, and propagation detection be improved with the use of live coordinates in addition to the date used to retrieve the image from NASA's repository? "

Objectives -The study objectives listed in Table 2 were developed and put into action in order to address these research questions.

Table 2: Research Objectives

Objectives	Description
1	A critical review of literature on wildfire detection and classification (2021- 2024)
2	Dataset accessing, creation and preprocessing
2.1	Programatically obtain the roboflow dataset to train in Google colab
2.2	Creating vegetation dataset by combining California shashta county datasets
2.3	Selecting the important attribute required for the training of the LSTM model
3	Implementation, Evaluation and Results for classification of wildfire through SVM
4	Implementation, Evaluation and Results for detection of wildfire through YOLO Model
5	Implementation, Evaluation and Results for Tracking of wildfire through LSTM Model
6	Integration of SVM, YOLO, LSTM model using interactive Standalone application to Classify, Detect and track the wildfire
7	Comparison of Developed Model
8	Comparison of Developed Model Verses Existing Model

Contributions: The project's major contribution is to classify the satellite image for the presence of wildfires using a pre-trained SVM model. We download a wildfire image dataset programmatically from the Roboflow library and train the YOLO model in a T4 GPU environment using a Google Colab notebook. The proposed model generates a synthetic dataset of Shasta County vegetation to train the LSTM model, which tracks the wildfire propagation path. This study aims to assist firefighters and the forest department in controlling wildfires. The developed model is to classify, detect, and track the fire propagation path, which will help the wildfire fighters to take the early precautionary steps. This project's minor contribution involves retrieving the real-time stored satellite image from NASA's repository, using the entered coordinates of latitude and longitude for a specific date. This will improve the satellite imagery's real-time wildfire propagation. In Section 2 of the technical study, titled " Literature Review on Real-Time Wildfire Progression Analysis and Prediction Using Hybrid Model (2021-24)," the other sections are organized as follows. Section 3 lays out the research approach that was utilized to classify and detect wildfire using hybrid deep learning techniques; Section 4 details the hybrid models' execution, evaluation, and outcomes. The study and its recommendations for further research are wrapped up in Section 5.

2 Literature Review on Real-Time Wildfire Progression Analysis and Prediction Using Hybrid Model

2.1 Introduction

In this section, the earlier work that has been done in the process of identification of wildfire is evaluated by dividing the section further into some sub-section like (2.2) IOT Based Fire Detection, (2.3) Wild Fire Detection Through Remote Sensing Methodology, and (2.4) Deep Learning Based Fire Prediction System And Identified Gap.

2.2 IOT based fire detection

Some research has been done for the effective event classification and intensity estimation for forest fires using IOT nodes and fuzzy based methodologies, V.K. Singh et al. (2022). Many researchers are concentrated on dealing with wireless sensor networks along with some machine learning approaches to detect forest fires efficiently, Burak Kizilkaya et al.
(2022). Udaya Dampage et al. (2022) Dr. A Chrispin Jiji et al. (2022); Avazov, K et al. (2023). Anshul Sharma et al. (2023). The fire and smoke monitor using the satellite sensing is achieved efficiently through remote sensing sensors Zheng.Y et al. (2023). Edge computing can also be used on UAV images to detect surface flame detection and inspection through object detection modules, which boost the unmanned models working pattern successfully W. Tao et al. (2023).

2.3 Study of methods for wildfire detection through remote sensing methodology

Drone collection RGB/IR image dataset is utilized to detect the forest fire using the flame neural network,which is also a good choice and produces an accuracy around 94% X. Chen et al. (2022). Sentinel-1 and Sentinel-2 satellite image can also be used effectively to map the severity of fire R. Lasaponara et al. (2023). The haze in the aerial image can be removed efficiently using a de-hazing neural network. This can enhance the process of fire detection L. Rui et al. (2023). The grass fire mapping and the rate of fire spread measurement using an NIR image from a small fixed wing UAS are done around tall grass fires in Kansas to enhance the grass fire detection process S. Gowravaram et al. (2023). The magnitude of forest structure and AGBD are examine in Canadian forest from multi-temporal land set structure to characterized versatility of wild fire effect T. Feng et al. (2024). The real monitoring of fire spots using a novel SBT-firenet based on a Himawari-8 satellite image proved remote sensing provides an extra ordinary result in wildfire spot estimation Z. Hong et al. (2024).

2.4 Investigation of Deep learning based wildfire prediction system and identified gaps

Around 4,00,000 Wildfire images are used to train and test the object detection model using the backbone like ResNet, VoVNet, and FBNetV3. In addition to this, Bayesian neural networks are employed to estimate the damaged area D. Q. Tran et al. (2022). VGGNet and optimized CNN models can produce an accuracy of 91.2% and 97.35% on forest fire detection L. Wang et al. (2023). The fire and smoke can also be detected with high accuracy using pre-trained models such as VGG16, InceptionV3, and Xception Veerappampalayam Easwaramoorthy Sathishkumar et al. (2023). Wild fire detection is decoded with transparency and explainable AI insights through FireDetXplainer with an accuracy of 99.91% S. F. Rubab et al. (2024). A Deep Learning Framework: Predicting Fire Radiative Power from the Combination of Polar-Orbiting and Geostationary Satellite Data During Wildfire Spread is proposed on the California wildfire spread dataset using FRP (Fire Radiative Power) and deep learning model Z. Dong et al. (2024). AF-Net (Active Fire): an active fire detection model using improved object-contextual representations on unbalanced UAV datasets is done by using HRNet-W48 backbone X. Hu et al. (2024). The wildfire and smoke is detected in different vegetation using the YOLOV5 model and FFSRP to achieve good prediction on versatile dataset W. Li et al. (2023). X. Sun et al. (2024). By remote sensing nighttime light to detect forest fire, the random forest classification model is also a good choice Y. Yu et al (2024). An in-depth survey on forest fire detection along with the detection of forest fire using CCTV cameras through CNN is also a quick solution for the real time estimation Ahmad Alkhatib et al. (2024). Swaraj Singh et al. (2024).

2.5 Conclusion and Identified Gaps

When it comes to recognizing, classifying, and tracking the spread of wildfires, the assessed relevant publications fall short in offering a comprehensive solution. Because each of the offered solutions focuses on a different aspect of wildfire prevention, but not all.

In contrast with the reviewed literature, the necessity of developing a model to accurately categorize, detect, and trace the spread of wildfires using satellite imagery obtained at the given coordinates from NASA's archive becomes abundantly clear. In doing so, it addresses research question 1.2 and also shows where the current literature is lacking, providing strong evidence that this is necessary. Also, study aim 1.3 is met, and sub-research question 1.2 about real-time image pulling from the repository may be answered with the help of the reviewed literature. By doing all this, it satisfies the objective 1 mentioned table 2 of Section 1.2.

3 Methodology Approach used and Project Design

3.1 Introduction

The proposed system is designed to classify the satellite images for the presence of wildfire, detection of wildfire, and tracking of wildfire directions based on the above depicted figure 2. The proposed model is developed using the python programming language, and the model used three IDEs like Spyder, Jupyter, and Google Colab, and we detail the steps involved in the process below

.

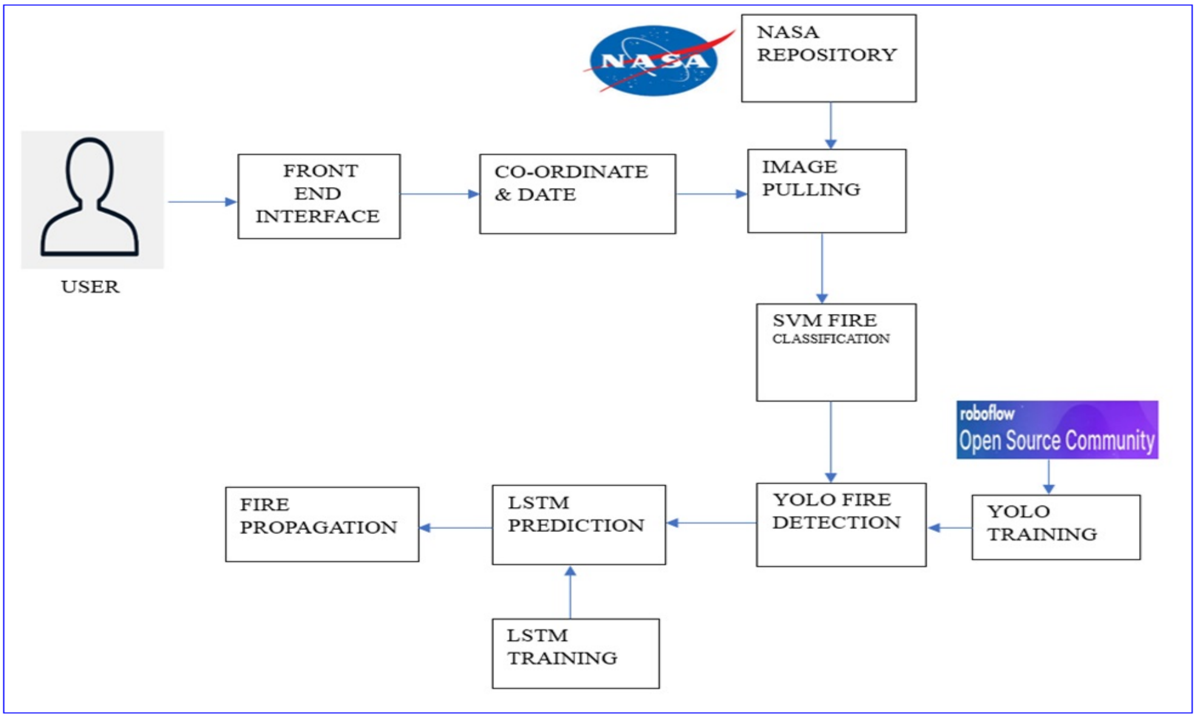

Figure 2: Proposed methodology System overview diagram

3.2 Image pulling from NASA's repository

This is the first step of the proposed model. The system receives two pairs of latitude and longitude coordinates, along with the date for fire detection, as input from an interactive desktop application that utilizes the TKinter library of the Python programming language. The system uses the first pair of coordinates to represent the top left corner of the image, and the second pair to represent the bottom right coordinate of the satellite image that NASA's repository contains. After entering the coordinates and date into the system via the user interface, the system creates a URL referencing the NASA repository to initiate a request to the website.

The system fires the request and downloads the corresponding satellite aerial image for the specified 800 x 800 dimension in the same root directory. Further systems can be depicted in detail using System Requirement Diagram in Figure 3 and through system architecture diagram in Figure 4.

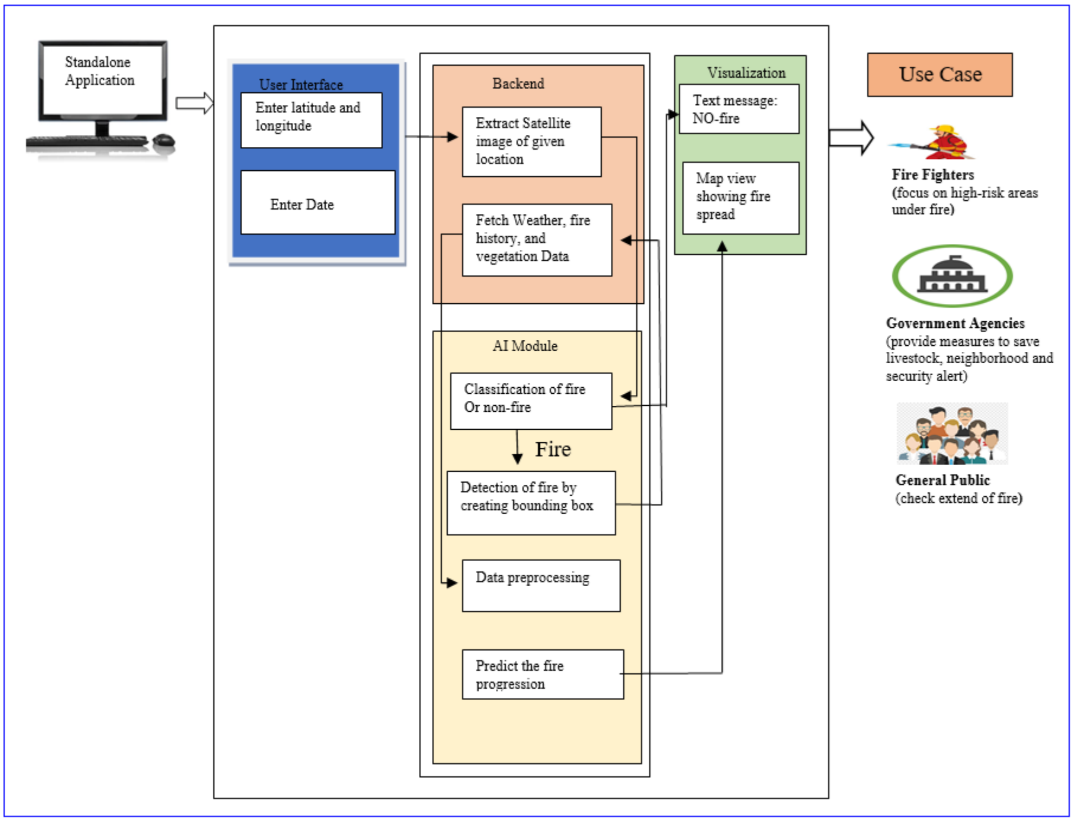

Figure 3: System Requirement Diagram

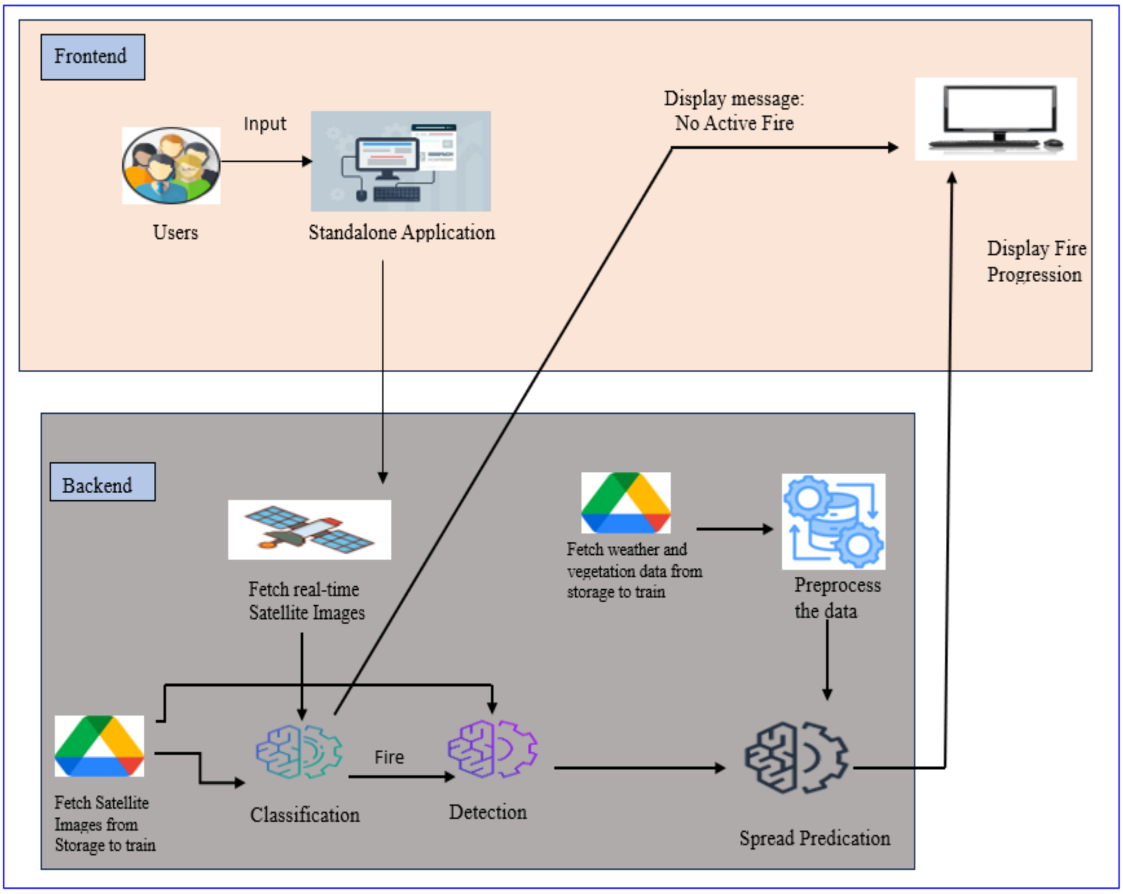

Figure 4: System Architecture Diagram

3.3 Fire Classification through SVM

For the purpose of detecting fires, we would employ the Support Vector Machine machine learning model. It is a supervised learning model in which the learning machine receives numerous inputs in addition to its output or values that correspond with it. SVMs use a hyperplane to differentiate between two classes. In doing so, the SVM classifier seeks to create the largest possible separation between the classes, as seen in Figure 5.

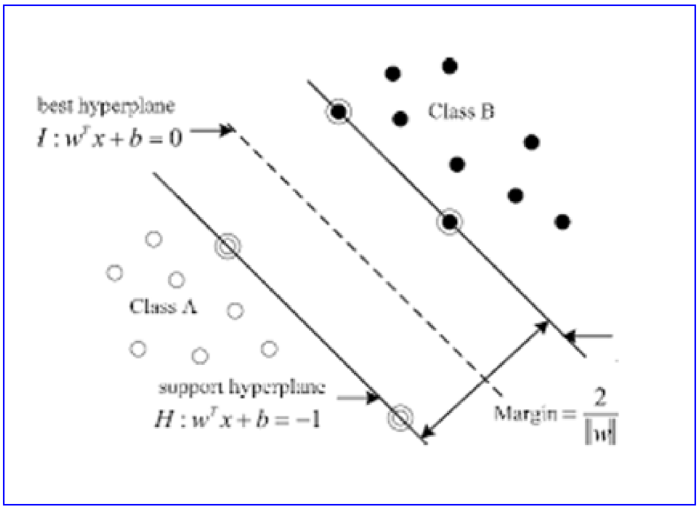

Figure 5: SVM Separating Hyperplanes

The SVM classifiers are separated maximally by a hyperplane in the form as in equation

(1): $wTx + b = 0$ (1),

and the other hyperplanes for bounding are given by the form in equation (2) and equation (3)

$wTx + b = -1$ (2) , $wTx + b = 1$ (3)

The +1 label class's data points satisfy the prerequisite. The label class data point for wTx+1 = <= -1 is satisfied when wTx + b >= 1. As a result, in our case, some of the data points' expected values will be different. When these data points diverge from the matching boundary planes, an error is produced. The primary objective is to minimize the number of data points that contribute to the inaccuracy while optimizing the bounding plane and margin. Equation (4) can be used to minimize the inverse of the distance between the two bounding hyperplanes from the origin in order to get the maximum margin.

½ wTw (4)

Consequently, the minimum error that is obtained from reducing the quantity is given by equation (5):

Min ½ $wTw + C \sum \boldsymbol{\xi} i$ (5)

Where, $\boldsymbol{\xi}$ = a positive weighted quantity. The weightage for the maximum margin required and the sum of errors is controlled by 'C'. The value of 'C' should be modified for the classifier's generalization power and provided to the algorithm.

To train the model, we need a collection of output values that are connected to the input values—also referred to as class labels—and a set of input values, often referred to as feature values. Images from drones or satellites are used as the input values. The values of features and the corresponding class labels make up training data. For this reason, we are classifying the data points for "FIRE" as positive +1 and "NON-FIRE" as negative -1. In particular, the generated mathematical model is used to predict the class values for given feature values. The suggested model uses the YOLO V8 model to identify the fire's sites of origin in the event that the extracted image is determined to be a wildfire. The pseudocode for the SVM is shown in Figure 3. As seen in Figure 6, a mathematical model for fire detection is created using the SVM classifier tool.

Training a SVM can be illustrated with the following pseudo code:

Algorithm 1 Training an SVM

Require: $\boldsymbol{X}$ and $\boldsymbol{y}$ loaded with training labeled data, $\boldsymbol{\alpha} \Leftarrow 0$ or $\boldsymbol{\alpha} \Leftarrow$ partially trained SVM

```
C ⇐ some value (10 for example)
repeat
    for all {x_i, y_i}, {x_j, y_j} do
        Optimize α_i and α_j
    end for
until no changes in α or other resource constraint criteria met
```

Ensure: Retain only the support vectors ($\boldsymbol{\alpha}_i > 0$)

Figure 6: SVM PseudoCode

3.4 Fire detection through the YoLO model

Object detection can be done using the YOLO (You Only Look Once) neural network model. This refers to the neural network that the program will use to identify things in the picture. In the past, detection frameworks would repurpose image classification techniques and repeatedly examine various regions of the image at varying sizes in order to identify things. This method is laborious and wasteful.

YOLO adopts a very different stance. It uses a single pass over the network to identify objects, and it only looks at the full picture once. Named after that. This is lightning quick. That's why it has become so well-liked.

3.4.1 Training the Wildfire images for YOLO V8

The image is being utilized by the system to achieve the effective identification of the wildfire in the image. This is the first step of the approach where the wildfire needs to be identified in the image for the further process. The wildfire identification module utilizes the Yolov8 approach that effectively performs the object recognition for the wildfire. This model needs to be trained first before being utilized for the wildfire recognition.

The training approach initiates with the installation of the ultralytics for the Yolov8 model and the downloading of the roboflow dataset for the implementation of training. The roboflow is connected using an API key, and the dataset for the wildfire recognition is downloaded through the [1]wildfire. The downloaded dataset is effectively scanned for

[1] URL: https://universe.roboflow.com/search?q=class%3

extracting the list of the files in the directory. The list of the files is then utilized for extraction of the number of files in the directory. The number of images for the training purpose is 1631 along with 102 images for testing.

Following the successful integration of the roboflow data and the effective shuffling of the wildfires dataset, the yolov8 model can be initiated for the yolo task for object recognition. The trained weights are being utilized to initiate the detection model and the training is done the dataset for 200 epochs with the image size as 640 and batch size as 32. The runs of the project are then stored as a zip file in the specified directory after the training of the yolov8 model.

The YOLOv8 is derived as a modification of the Convolutional Neural Network. It utilizes the components of the CNN approach in a unique and effective manner to achieve the object recognition with greater accuracy. The Yolo architecture is made up of 24 convolutional layers with varying parameters which are assisted by 4 max pooling layer and various dropout and batch normalizations to regularize the model and avoid overfitting. The model finally culminates in 2 Fully connected layers.

The initial convolutional layers decompose and reduce the channels which are then max pooled with a stride set as 2 and the kernel size 2x2. These max pooling layers are all identical in all the layers of this model. The subsequent convolutional layers have increasing kernel sized to accommodate for the increase in the information. The activation function being utilized for these layers is the ReLU activation function. The activation function is identical for all the layers except for the fully connected layers that implement a linear activation function to form the .pt file which is the trained data file for Yolo8. This .pt file will be used in the upcoming steps to alert the blind person for the wildfire.

3.4.2 Testing the model for wildfire using YOLO V8

In this step, the image input for the wildfire has been given, which was pulled from NASA's repository. The trained model file.pt is used to detect the wildfire in the live input image; their upper left rectangular positions are obtained from the.pt file. This position is monitoring the stability of the image, and then wildfires are marked in a red color bounding box. Then this is being used by the LSTM model to estimate the tracking path of the wildfire as explained in the coming steps.

Conclusion: By doing all this, it satisfies the objective 2.1 mentioned in Table 2 of Section 1.2 for programmatically obtaining the Roboflow dataset at Google Colab IDE.

3.5. Dataset Segregation and Wildfire Propagation through LSTM *:*

Once the dataset is collected and stored in a worksheet, it is read into a double dimension list using pandas library. This double dimensional list is used to segregate the data into two lists, like X and Y. The X list contains attributes like Date, Temperature, RelativeHumidity, WindSpeed, WindDirection, Precipitation, longitude, latitude, ndvi, SLOPE, ELEVATION, FUEL_COVER , LAND_COVER and Y list contains attribute like Fire_NonFire.

3.5.1 Training and testing Data Creation

The function train_test_split() then uses the two lists, X and Y, to build the test and train sets. This function takes in X, Y, a test size of 0.25, and boolean value shuffling of false since the model does not shuffle the data based on the date. The random train and test data are then generated by this function and assigned to four lists, namely train_x, train_y, test_x, and test_y. Since the train and test lists are mapped to ordinal values, any allocated data types may be handled with ease by the list. The MinMaxScaler() function is used to normalize these test and train lists that were generated. Using the minimum and maximum values as a guide, this approach normalizes all the data. The function reshape( ) is used to reshape the lists into a single dimension after they have been normalized. To predict the coordinates of the spread of wildfires, the reshaped data is fed through various LSTM models.

3.5.2 LSTM Training

Input parameters for this neural network include test_x, test_y, train_x, and a scalar normalization object. The LSTM model with a few parameters is introduced with 20 units of samples, a return sequence of TRUE, and a one-dimensional space with a single feature. A dropout() is also used, with a rate of 20% (or 0.2) between the output layer and the last hidden layer, in addition to the other parameters. Repeat this process of adding parameters and dropouts for 40, 80, and 80 units of rates. Then, a dense layer with a kernel size of 40 and an activation function of "tanh" is added. Activation functions on neurons are used by the dense layer of a densely linked neural network to effectively learn new information. Two dense layers, one with a kernel size of 1 and the other with a kernel size of 40, make up the fundamental LSTM neural network. When building a neural network, a batch size of 10 and 250 epochs are employed. Figure 7 below shows the architecture for the LSTM neural network.

LSTM	
Layer	**Activation**
LSTM 20 Samples,Kernel=1	relu
Dropout 20%	
LSTM 40 Samples,Kernel=1	
Dropout 20%	
LSTM 80 Samples,Kernel=1	
Dropout 20%	
LSTM 80 Samples,Kernel=1	
Dropout 20%	
Dense 40	tanh
Dense 40	tanh
Dense 1	None
Adam Optimizer	
Batch size 10	
Epochs 250	

Figure 7: LSTM Architecture

Once the data is trained, all the trained info is stored in a.h5 file. The Tanh activation function, which was used during the construction of the LSTM neural network, is mentioned in the below equation.

$$tanh = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \quad \text{\_\_\_\_\_\_}(6)$$

3.5.3 LSTM Testing

Once the dataset is trained and the system gets the latitude, longitude, and number of days to predict the propagation, then the model predicts the coordinates using the trained data in the LSTM.h5 file. Then, by using the Python library GPS_Mapper, the coordinates are mapped for the pulled images from NASA's repository. Then these mapped coordinates for the image are used to mark the propagation points in the image. This process is repeated for the given number of days, and finally all the images are merged to get a gif image to show the output.

Conclusion: By doing all this, it satisfies the objectives of 3, 4, 5 and 6 as mentioned in Table 2 of Section 1.2 for the implementation of the SVM model in Section 3.1, the implementation of the YOLO v8 model in Section 3.2, and the implementation of the LSTM model in Section 3.3, along with the complete integration of all the models in a standalone application, which satisfies Objective 6.

4 Data creation and cleansing, Evaluation, Implementation of models and Results in Real-Time Wildfire Progression analysis and prediction using Hybrid Model

4.1 Introduction

This section analyzes the datasets used to train the LSTM model to track wildfire trajectories, as well as their generation and preparation. This part also evaluates the model's wildfire detection and tracking output.

4.2 Dataset Creation and Pre-processing

4.2.1 Collection of Weather Dataset

One of the necessary datasets for our study is weather data, which is needed to forecast the fire's course. The National Oceanic and Atmospheric Administration website, which offers station-by-station historical meteorological data for the California region, is the source of weather data [1] is downloaded, which has the following parameters as depicted in Table 3.

Table 3: Weather data parameters and their description

Field Name	Description	Units	Data Type
DailyAverageDewPointTemperature	Average daily dew point temperature	Fahrenheit (° F)	Int
DailyAverageDryBulbTemperature	The average daily dry bulb temperature recorded	Fahrenheit (° F)	Int
DailyAverageRelativeHumidity	Average daily Humidity	Percentage(%)	Int
DailyAverageSeaLevelPressure	Average daily sea level pressure	Inch of mercury (Hg)	Int
DailyAverageStationPressure	Daily average station pressure	Inch of mercury (Hg)	Int
DailyAverageWetBulbTemperature	Average daily wet bulb temperature	Fahrenheit (° F)	Int
DailyAverageWindSpeed	Average wind speed in mph	Miles Per Hour (mph)	Int
DailyAverageDewPointTemperature	Average daily dew point temperature	Fahrenheit (° F)	Int
DailyAverageDryBulbTemperature	The average daily dry bulb temperature recorded	Fahrenheit (° F)	Int
DailyPeakWindDirection	Daily peak wind direction	Compass Degrees	Int
DailyDryBulbTemperature	Daily dry bulb temperature	Fahrenheit (° F)	Int
DailyPrecipitation	Daily precipitation	Inch(in)	Int
PeakWindDirection	Maximum wind direction	Compass Degrees	Int
DryBulbTemperature	Dry bulb temperature	Fahrenheit (° F)	Int
Date	recorded	MM DD YYYY	Date

[2]4.2.2 Historical Wildfire Dataset

We need historical wildfire data for our study in order to predict the path of the fire. Data about past fires has been gathered from a variety of sources, including GEE, Idaho State University's GIS Training and Research Center, and the United States Forest Services (USFS). This dataset of wildfire[2] contains the following attributes:

Table 4: Attributes of Historical Wildfire Dataset

Field Name	Description	Data Type
X	Latitude	Decimal
Y	Longitude	Decimal
Discover_Year	Year of the fire	Number
Fire_Number	Alternative ID for each fire that occurred	Number
Total_acres_burnt	Number of acres burnt by fire	Decimal
County	Region of the fire incident	Text
Slope	A Integer value indicating slope of the region	Integer
Elevation	A Integer value indicating elevation of the region	Integer
Fire_Name	Name of the fire	Text
Ignition	Start date of the fire	DateTime
Fire_Out	End date of the fire	DateTime
Location	Area of the fire in the county	Text
Fuel_model	Presence of fuel levels in the area	Text
Cover_Class	Type of land cover in the area	Text
Fire_Intensity	Length of the flame of a fire	Text

4.2.3 Remote Sensing Data for Vegetation Index

Data on vegetation is used to forecast the progression of fires. The USGS, NASA's Earth Data, and Google Earth Engine (GEE) have all provided remote sensing data on vegetation. The sample vegetation data collected is displayed in Figure 30. By entering Javascript code in the GEE editor, we may access satellite and remote sensing data using GEE, a geodatabase tool. We must choose a geographic area of our interest, in this example, Shasta County, by either entering the latitude and longitude or by simply choosing the relevant spot on the map. We are gathering information about the chronology of the several fires that have broken out in Shasta in 2019. We are also taking into account the dataset with the least amount of cloud coverage. The GEE collection set that we are using to extract the data must be specified in the code. We must take into account a number of vegetation indices for the vegetation data [3], which must be computed and are listed in the table below.

3

[2] : https://www.ncdc.noaa.gov/cdo-web/datatools/lcd),

[3] http://giscenter.isu.edu/research/Techpg/HFD

Table 5: Different Vegetation Indices

Index	Description	Formula
NDVI	Quantifies the greenness and is useful to understand density and health of the plants	(NIR - R) / (NIR + R)
NDMI	Used to detect the water content in the vegetation	(NIR - SWIR) / (NIR + SWIR)
EVI	Quantifies the greenness and corrects the atmospheric conditions	G * ((NIR - R) / (NIR + C1 * R – C2 * B + L))
NBR	Used to identify the area burnt after the fire	(NIR - SWIR) / (NIR + SWIR)
NBR 2	It modifies the NBR to highlight the water content in the vegetation after the fire	(SWIR1 – SWIR2) / (SWIR1 + SWIR2)

One source for all vegetative indicators is the NOAA [4]dataset. Once all the datasets are collected, they are merged, and needed attributes are selected to form a single dataset that contains the following attributes: Date, Temperature, RelativeHumidity, WindSpeed, WindDirection, Precipitation, longitude, latitude, ndvi, FIRE_OUT, SLOPE, ELEVATION, FUEL_COVER, LAND_COVER and Fire_NonFire.

Conclusion: By doing all this it satisfies the objective 2.2 and 2.3 mentioned in table 2 of section 1.2 for creating vegetation dataset by combining California shashta county datasets and also for selecting the important attribute required for the training of the LSTM model.

4.3 Implementation, Evaluation and Results of YOLO V8

4.3.1 Implementation YOLO V8

Using the Roboflow dataset, the YOLO V8 model is implemented in the Google Colab environment. The wildfire recognition dataset may be obtained from 5 wildfires, and the roboflow can be integrated with an API key. It efficiently scans the downloaded dataset to retrieve the directory's file list. The number of files in the directory can then be extracted using the file list. There are a total of 1631 images, including 102 for training purposes. The Yolov8 model can be started for the Yolo task of object recognition after the roboflow data and wildfires dataset have been successfully integrated and shuffled. With a batch size of 32 and an image size of 640, the detection model is trained for 200 epochs using the trained weights.

4.3.2 Evaluation and results of YOLO V8 for Confusion Matrix

The deployed model is evaluated on a Windows-based, Intel Core i5-powered, 16 GB RAM PC. To test the deployed system, we utilize the following equation to express the confusion matrix's accuracy score parameter. Equations 7, 8, 9 and 10 show the accuracy, precision, recall and F measure , respectively.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad -(7) \qquad \text{Precision(P)} = \frac{TP}{TP+FN} \quad -(8)$$

---

[4] NOAA/VIIRS/001/VNP09GA

$$\mathrm{Recall(R)} = \frac{TP}{TP+FP} \quad -(9) \qquad \mathrm{Macro-F1} = \frac{2*P*R}{P+R} \quad -(10)$$

Here, TP is True positive cases, TN is True Negative cases, FP is False positive cases and FN is False Negative cases.

4.3.3 Evaluation and results of YOLO V8 for F1 measure

This statistic, which is a mean of recall and precision, is sometimes called the F-score. Because it is based on the harmonic mean, it is most accurate when the two numbers are close to one another. F1 = 2 * recall * precision / (precision + recall) is the formula for the F-measure. If there are no relevant wildfire obtained, the F-measure value is 0, and if there are all relevant wildfires retrieved, the F-measure value is 1. With good precision and recall, the F-measure is high; when the two metrics are significantly different, it's close to the one that's worse.

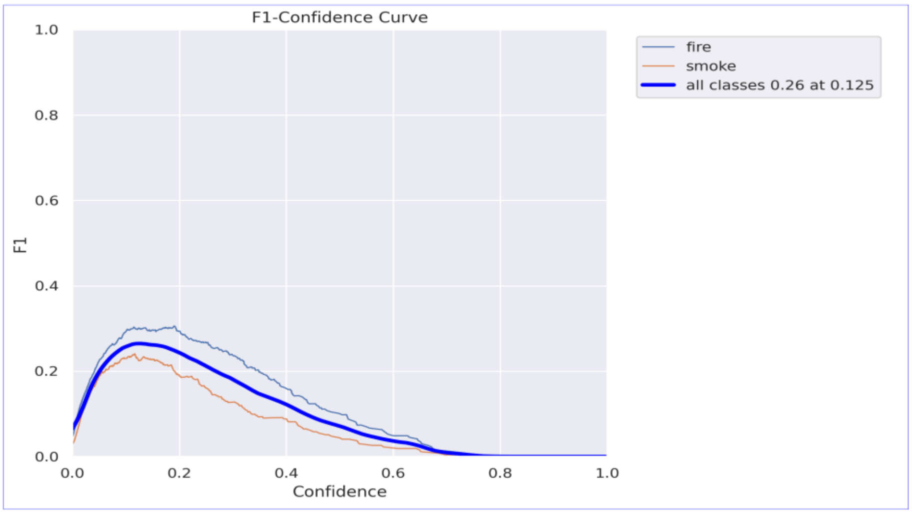

Figure 8: F1 Score for YOLO V8 model on Validation dataset

4.3.4 Evaluation and results of YOLO V8 for Precision

The percentage of wildfire that were successfully identified relative to the total number of objects is measured by this metric.

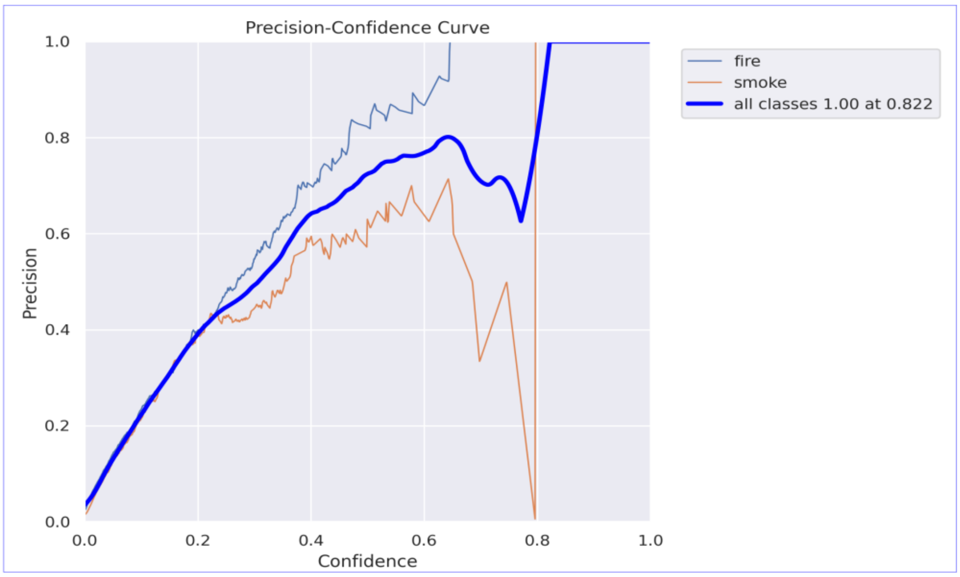

Figure 9: Precision for YOLO V8 model on Validation dataset

4.3.5 Evaluation and results of YOLO V8 for Recall

This statistic counts how many real wildfires were successfully identified as a percentage of all real items.

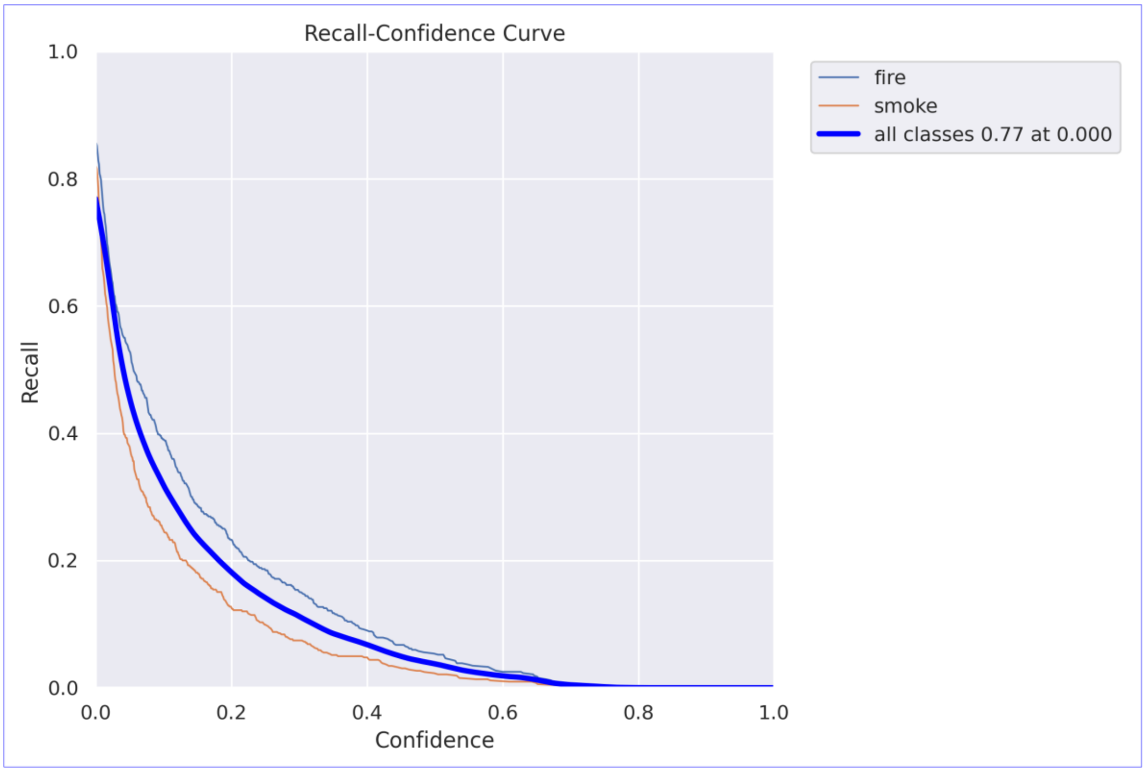

Figure 10: Recall for YOLO V8 model on Validation dataset

4.3.6 Evaluation and results of YOLO V8 for train box loss, train class loss and train accuracy loss

The train box loss, train class loss and train accuracy loss on the training dataset for 50 epochs are shown below.

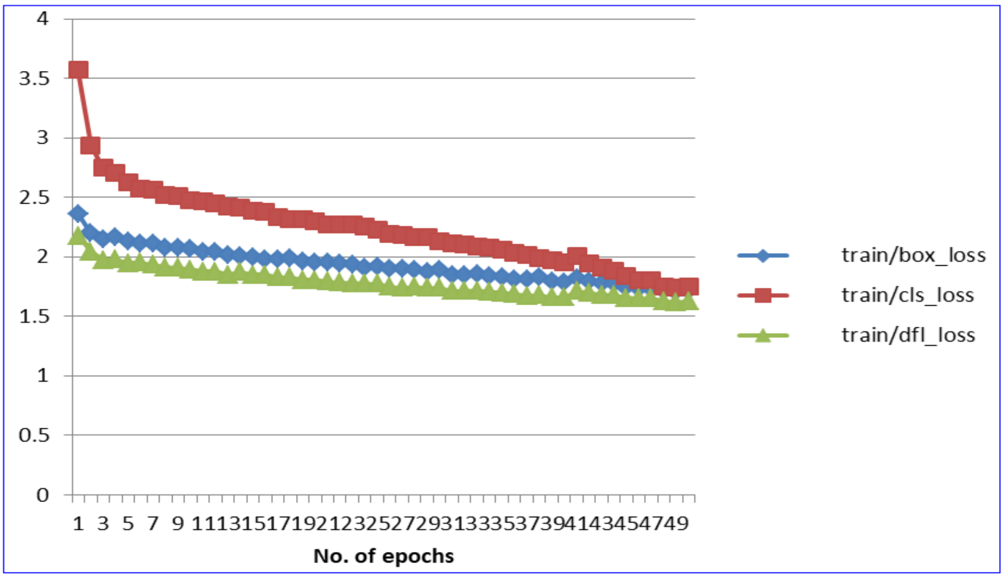

Figure 11: Training class loss, box loss and accuracy loss while training for 50 epochs.

4.3.7 Evaluation and results of YOLO V8 for validation box loss, validation class loss and validation accuracy loss

The validation box loss, validation class loss and validation accuracy loss on the validation dataset for 50 epochs are shown below.

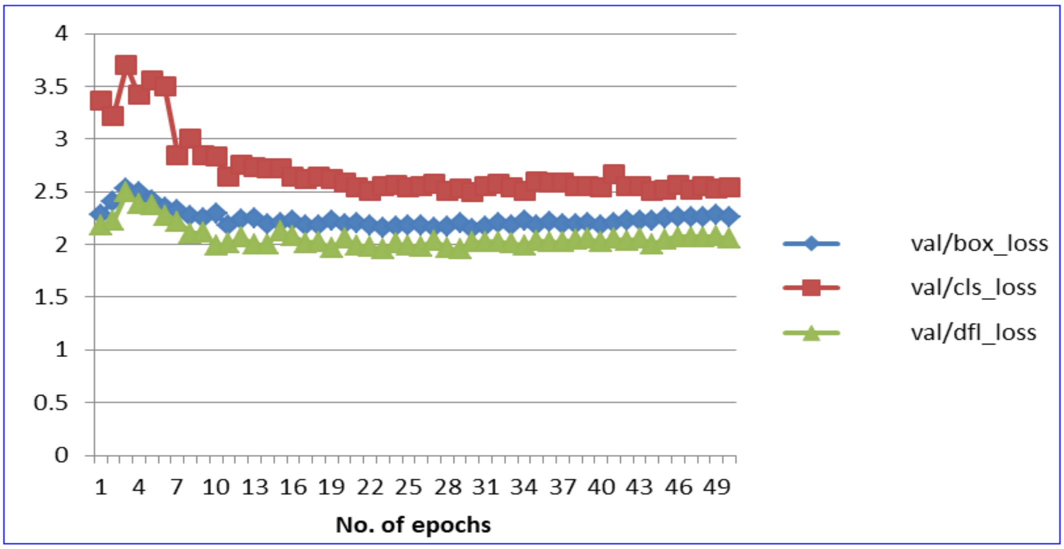

Figure 12: Validation class loss, box loss and accuracy loss while training for 50 epochs on validation dataset

4.3.8 Evaluation and results of YOLO V8 for learning rate with 50 epochs

In an optimization setting where the loss function is minimized, the learning rate (LR) controls the amount by which the weights of the neural network change. Optimizers and loss functions rely on this parameter.

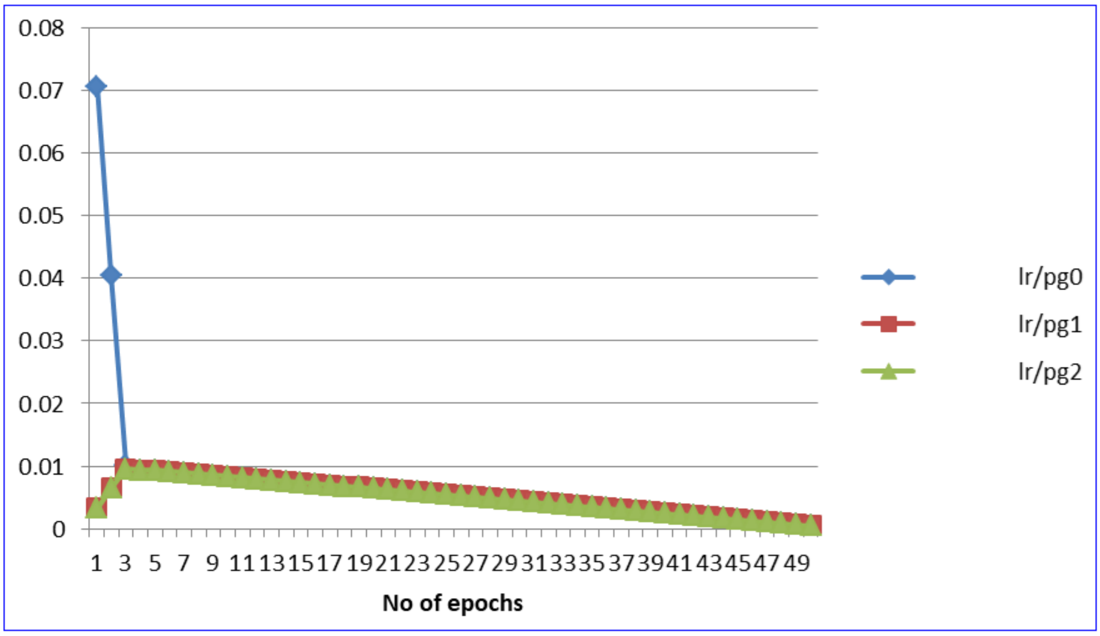

Figure 13: Learning rate for 50 epochs training

4.3.9 Evaluation and results of YOLO V8 for all training parameters

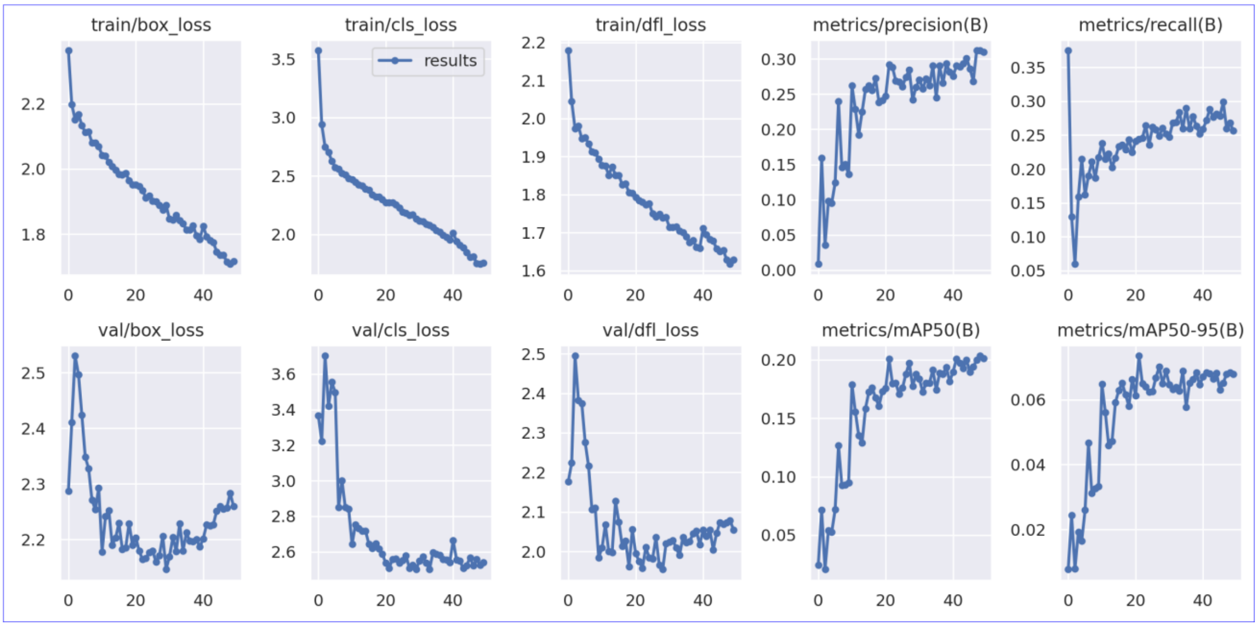

Figure 14: All training parameters for 50 epochs

4.4 Implementation of LSTM

Following data collection and storage in the worksheet, the LSTM Model dataset is read into a pandas library-generated double-dimensional list. By using this two-dimensional list, we may divide the data into two sets, for example, X and Y. Data such as Date, Temperature, RelativeHumidity, WindSpeed, WindDirection, Precipitation, longitude, latitude, ndvi, SLOPE, ELEVATION, FUEL_COVER, and LAND_COVER are all part of the X list. An example of an attribute in the Y list is Fire_NonFire.

After that, the train_test_split() function constructs the test and train sets using the X and Y lists. Since this model does not shuffle the data depending on the date, it accepts X and Y as inputs and returns false for the boolean value shuffling. The test size is 0.25. This function generates sample data for both the train and test sets and stores them in four lists: train_x, train_y, test_x, and test_y. This list can easily manage any allocated data type since the train and test lists are mapped to ordinal values. This generated data for tests and training is normalized using the MinMaxScaler() method. It normalizes all the data by using the minimum and maximum values as references. When the lists are normalized, the method reshape() is utilized to transform them into a single dimension. Using the transformed data, multiple LSTM models are applied to forecast the locations of wildfire spread.

An object for scalar normalization and the variables train_x and train_y are input to this neural network. The one-dimensional space with a single feature, 20 units of data, and a return sequence of TRUE are the initial parameters of the LSTM model. Aside from the extra parameters, a 20% (or 0.2) dropout() is used between the output layer and the final hidden layer within the algorithm. The steps of adding the parameter and dropping out are repeated at rates of 40, 80, and 80 units again. Activation function "tanh" and 40-kernel Dense layer follow. An efficient way for a densely linked neural network's layer to learn new information is by using activation functions on neurons. The two dense layers that make up an LSTM neural network are the one with a 1x1 kernel and the other with a 40x40 kernel. The neural network building procedure makes use of a batch size ranging from 10 to 250 epochs.

4.4.1 Histogram of weather dataset

While we were preparing the data, we also looked for false rows and extra columns in the weather data and eliminated them since they weren't relevant for training. Upon careful examination, the "avg" columns, which represent the average of the highest and lowest values for each feature in our dataset (e.g., Temp_avg, Hum_avg), should be removed to show its histogram in the below figure.

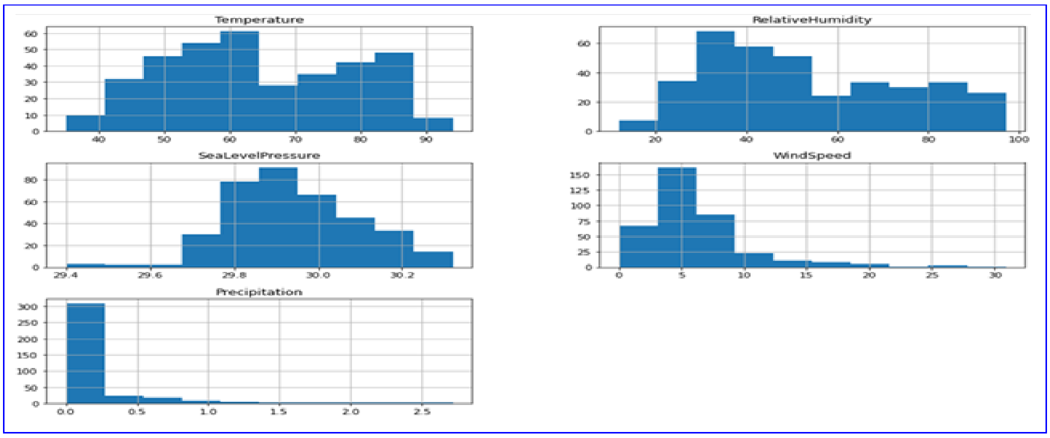

Figure 15: Histogram of weather dataset

4.4.2 Histogram of wildfire dataset

There have been several naming convention changes, such as X, Y being replaced with latitude and longitude, Fuel Model being renamed to Fuel_cover, and cover_class being converted to land_cover. The dataset's OBJECTID field serves as a distinct key for every wildfire occurrence; it is transformed into a string in order to display the histogram in the following manner:

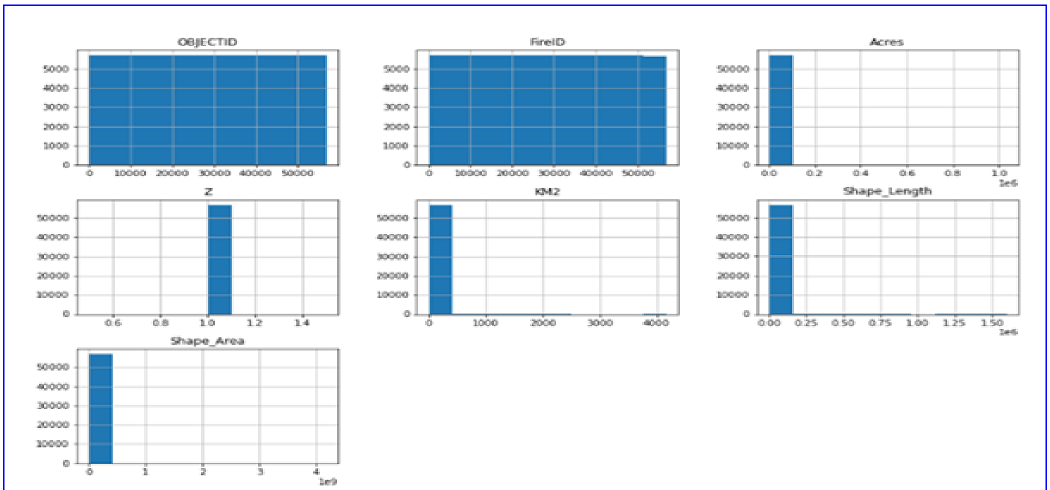

Figure 16: Histogram of Wildfire dataset

4.4.3 Histogram of vegetation dataset

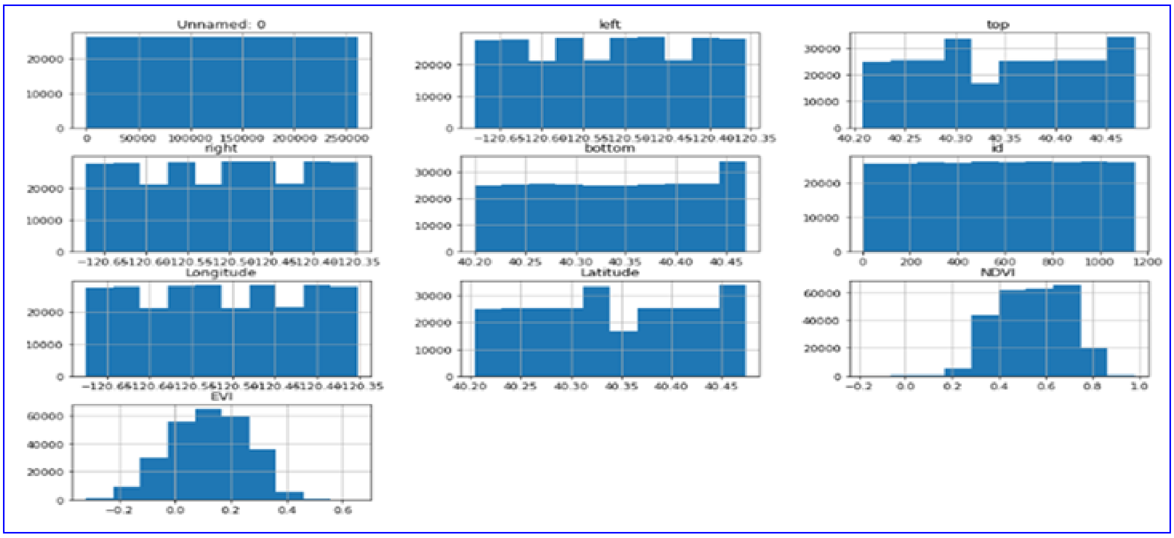

Figure 17: Histogram of vegetation dataset

4.4.4 LSTM training loss and accuracy

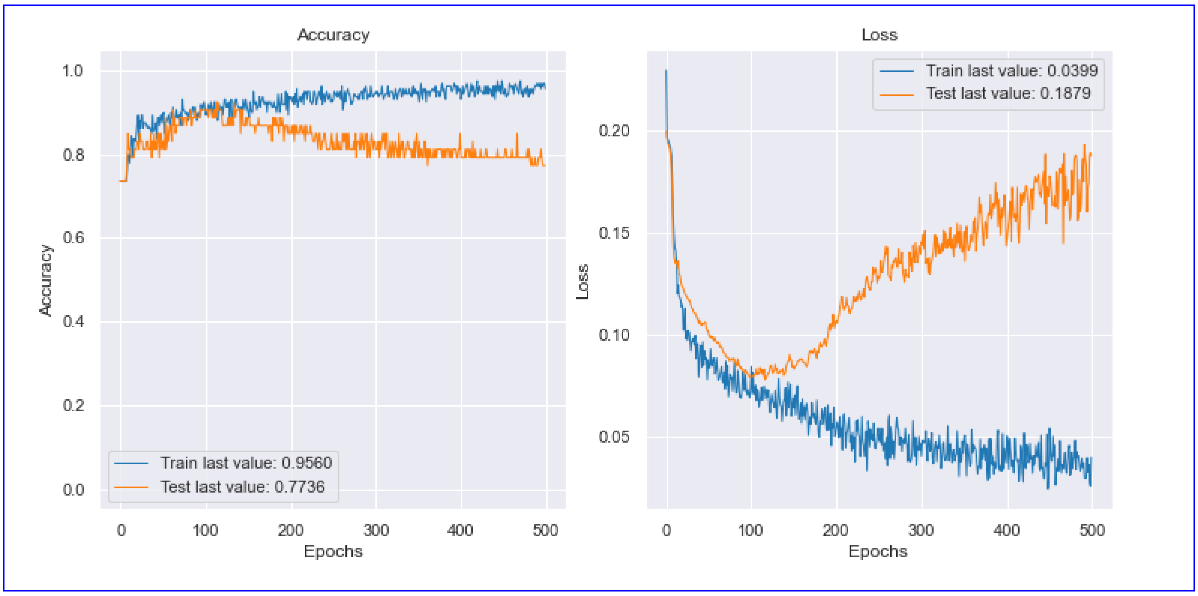

Figure 18: LSTM Accuracy and loss for 500 epochs

5. Discussion and Comparison

5.1 Comparison of Developed Model

The designed approach is used SVM and YOLO model for fire classification and fire detection. The obtained precision and recall are shown in the below figure 19 and 20 respectively.

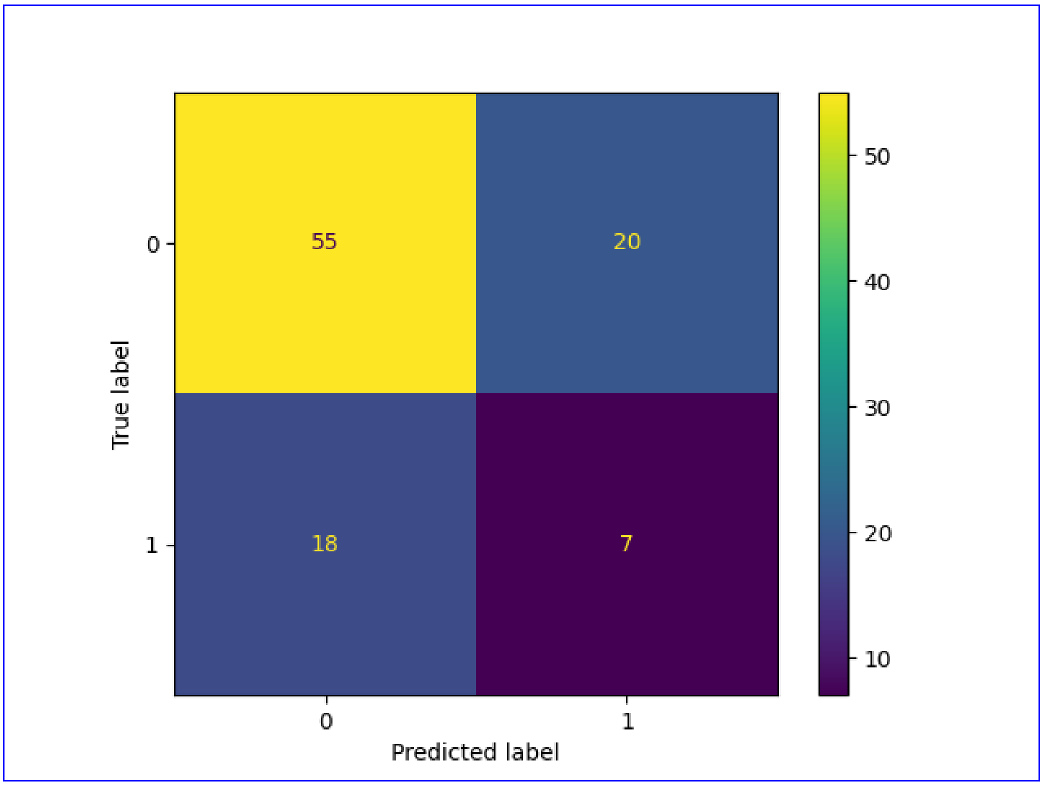

Figure 19: Confusion matrix for SVM model

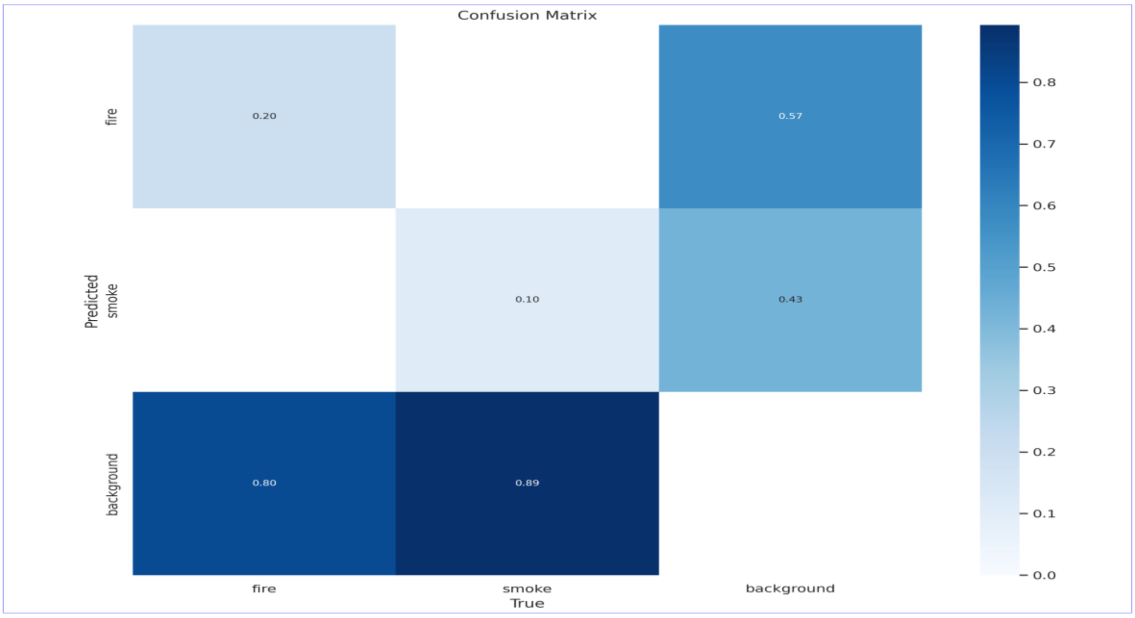

Figure 20: Confusion matrix for YOLO V8 model

Upon examining the confusion matrix of the SVM and YOLO v8 models, we discovered that the precision and recall of the SVM model are 0.75 and 0.733, respectively. On the other hand, for 50 epochs of training, the YOLO V8 yields 0.80 and 0.77 of precision and recall.

This is evident that YOLO v8's performance is better than that of the SVM model, and this can be shown in the below table 6 and figure 21.

Table 6: Precision and Recall value of SVM and YOL V8

Model	Precision	Recall
SVM	0.75	0.73
YOLO V8	0.8	0.77

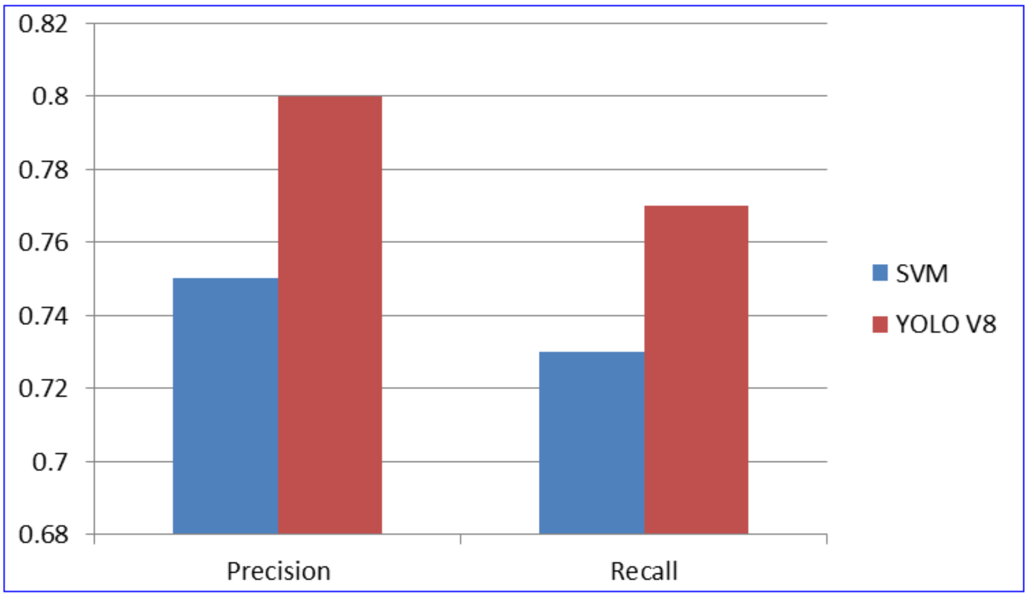

Figure 21 : Precision and Recall comparison between SVM and YOLO V8 Model

5.2 Comparison of Developed Model vs Existing Model

To evaluate the designed model the Root Mean Square Error (RMSE) is used, Which indicates the amount of error done by system while tracking and identifying the fire using LSTM model. The RMSE can be represented by the equation 11.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(x_{1,i} - x_{2,i}\right)^2}{n}} \quad \text{\_ (11)}$$

Where,

Σ - Summation , $(x_1 - x_2)^2$ - Differences Squared for the summation in between the expected no. of results ,n - Number of Trails

Table 7 tabulates the RMSE results for the designed model LSTM and the RMSE score of the methodology discussed in Xufeng Lin et al.(2023). and Figure 22 shows the resulting graph.

Table 7: RMSE Records of different methodologies

Model	RMSE
LSTnet	0.39
RNN	0.46
SVR	0.44
LSTM	0.29

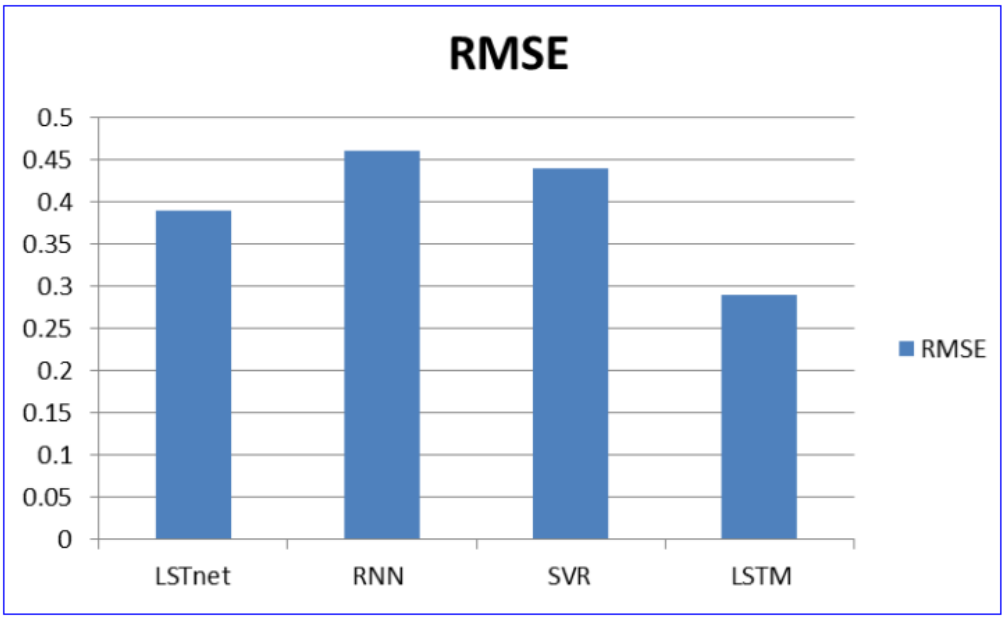

Figure 22: RMSE Comparison of different models

The table and graph clearly show that the LSTM model outperforms all other techniques in terms of RMSE. This is mainly because the model was trained using high-quality data that had been properly scaled, as well as because it was hybridized with another model.

Conclusion : This chapter of Comparison satisfies the objectives of 7 and 8 to compare the developed models among themselves and also with other models.

6 Conclusion and Future Scope

Several literature reviews on current methods for wildfire detection and spread prediction analysis were incorporated into the proposed framework. The proposed model derives its own solution to obtain good outcomes in near real-time from the total research done on wildfire detection and progression prediction analysis. In order to address this particular issue, three distinct phases have been executed. The three steps are as follows: categorization, detection, and prediction. The necessary image data for detection and classification was gathered by the model. Roboflow, Google and the Mendeley datasets were used to gather aerial and satellite images of the wildfire along with the vegetation dataset of Shasta county. The model has classified the image as a wildfire using a pre-trained support vector machine (SVM) model.

The model used the YOLOv8 model to determine exactly where the fire was in the image. In order to train YOLO v8, Model utilized 1631 images for training and 102 images for testing. To achieve the best trained data recorded in the.pt file, the dataset is trained for 50 epochs with an image size of 640 and a batch size of 32 in a Google Colab environment. The test set images performed better when using the YOLOv8 model. The Yolov8 model is well-suited for real-time wildfire detection because it requires minimal testing time. In other words, if aerial images are taken by a drone or satellite and want to identify a fire, the YOLOv8 model is a good option.
The proposed model utilized LSTM to forecast the potential areas of fire spread over the following five days. Data on weather, fires, and vegetation were all input into the model. Fire behavior is highly dependent on environmental factors such as humidity, temperature, and vegetation indices (NDVI, NDMI, acres burned, etc.). Cleanup and format transformation are

performed on the collected datasets. The model is tested with a variety of dispersion metrics, including standard deviation, range, and mean. In order to find a reliable model for real-time wildfire prediction, the model is tuned with the number of epochs while increasing the training accuracy. The final product, the standalone interface, will be a precise, user-friendly, and inexpensive way to fight wildfires. This was put into action by combining our three machine learning models for upcoming wildfire detection and prediction. Firefighters will have ample time to plan and contain wildfires with this approach. Additionally, it is anticipated to assist residents and visitors of places Shasta County, California county side and others in monitoring wildfire occurrences and implementing necessary safety protocols.

Finally, the designed models like SVM and YOLO v8 are compared themselves for precision and recall parameters. YOLO v8 clearly outperforms the SVM. On the other hand, we evaluate the designed model against some existing models for RMSE parameters, and it yields the best RMSE value of 0.29, demonstrating its excellence.

Future work : The long-term goal is to train the models using additional datasets that include terrain, power line, and elevation information. The Proposed model can be enhanced by creating new machine learning models, testing them against current ones, and ultimately making present models more accurate for real time wildfire images obtained from satellites or fast flying drones. The future model may be equipped to deal with all odds in the images, like blur, few shots, etc. by incorporating high end image processing techniques by using transfer learning and others.

Acknowledgement

With heartfelt gratitude, I acknowledge Dr. Catherine Mulwa for her unwavering patience and invaluable guidance throughout this journey. Her meticulous supervision and insightful feedback were crucial in shaping my work. I also extend my sincere thanks to the National College of Ireland for offering essential library resources, which played a significant role in the development of my technical report and literature review. Lastly, I am deeply thankful for the unwavering support, encouragement, and faith of my parents, without whom this accomplishment would not have been possible. Their blessings were my greatest strength.

References

V. K. Singh, C. Singh and H. Raza. (2022). Event Classification and Intensity Discrimination for Forest Fire Inference with IoT, IEEE Sensors Journal.8869-8880.

Burak kizilkaya, enver ever, hakan yekta yatbaz and adnan yazici. (2022). An Effective Forest Fire Detection Framework Using Heterogeneous Wireless Multimedia Sensor Networks, ACM Transaction Multimedia Computer Communication.

Udaya Dampage, Lumini Bandaranayake, RidmaWanasinghe, Kishanga Kottahachchi and Bathiya Jayasanka. (2022). Forest fire detection system using wireless sensor networks and machine learning. Springer nature. Dr. A Chrispin Jiji, Dr. KDV Prasad, Nagarajan Jeyaraman, Dillip Narayan Sahu, Dr A.Yasmine begum and Nitin Jagannath Patil. (2022). iot based automatic forest fire detection based on machine learning approach, annals of forest research 1792-1809.

Avazov K., Hyun A.E, Sami S A.A, Khaitov A, Abdusalomov A.B and Cho Y.I. (2023).Forest Fire Detection and Notification Method Based on AI and IoT Approaches, future internet.

Anshul Sharma, Anand Nayyar, Kiran Jot Singh, Divneet Singh Kapoor, Khushal Thakur and Shubham Mahajan. (2023). An IoT-based forest fre detection system: design and testing, Multimedia Tools and Applications.

Zheng Y, Zhang G and Tan S Feng L. (2023). Research on Progress of Forest Fire Monitoring with Satellite Remote Sensing. Agricultural & Rural Studies.

W. Tao and F. An. (2023). ATSS-Driven Surface Flame Detection and Extent Evaluation Using Edge Computing on UAVs. IEEE open Access Journal.

Xiwen Chen, Bryce Hopkins, Hao Wang, Leo Oneill, Fatemeh Afghah, Abolfazl Razi, Peter Fulé, Janice Coen, Eric Rowell And Adam Watts. (2022). Wildland Fire Detection and Monitoring Using a Drone-Collected RGB/IR Image Dataset. IEEE open Access Journal.

R. Lasaponara, C. Fattore and G. Modica. (2023). Imaging Burned Areas and Fire Severity in Mediterranean Fragmented Ecosystems Using Sentinel-1 and Sentinel-2: The Case Study of Tortoli–Ogliastra Fire (Sardinia). IEEE Geoscience and Remote Sensing journal.

L. Rui, Z. Jiaqing and H. Yang. (2023). Dehazing Model of Extra-High Voltage Converter Station Based on Two-Stage Attention. IEEE Access journal.

S. Gowravaram, Haiyang Chao, Zhenghao Lin, Sheena Parsons, Tiebiao Zhao and Ming Xin.(2023).Prescribed Grass Fire Mapping and Rate of Spread Measurement Using NIR Images From a Small Fixed-Wing UAS. IEEE Journal Remote Sensing.

Tuo Feng, Laura Duncanson, Steven Hancock, Paul Montesano, Sergii Skakun, Michael A. Wulder, Joanne C. White, David Minor and Tatiana Loboda.(2024).Characterizing Fire-Induced Forest Structure and Aboveground Biomass Changes in Boreal Forests Using Multitemporal Lidar and Landsat. IEEE Journal Remote Sensing.

Z. Hong, Zhizhou Tang, Haiyan Pan, Yuewei Zhang, Zongsheng Zheng, Ruyan Zhou, Yun Zhang, Yanling Han, Jing Wang and Shuhu Yang. (2024). Near Real-Time Monitoring of Fire Spots Using a Novel SBT-FireNet Based on Himawari-8 Satellite Images. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.

D. Q. Tran, M. Park, Y. Jeon, J. Bak and S. Park. (2022). Forest-Fire Response System Using Deep-Learning-Based Approaches With CCTV Images and Weather Data. IEEE open Access journal.

L. Wang, H. Zhang, Y. Zhang, K. Hu and K. An. (2023). A Deep Learning-Based Experiment on Forest Wildfire Detection in Machine Vision Course. IEEE open Access journal.

Veerappampalayam Easwaramoorthy Sathishkumar, Jaehyuk Cho, Malliga Subramanian and Obuli Sai Naren. (2023). Forest fre and smoke detection using deep learning-based learning without forgetting. Springer open.

S. F. Rubab, A. A. Ghaffar and G. S. Choi. (2024). FireDetXplainer: Decoding Wildfire Detection With Transparency and Explainable AI Insights. IEEE open Access journal.

Z. Dong, C. Zheng, F. Zhao, G. Wang, Y. Tian and H. Li. (2024). A Deep Learning Framework: Predicting Fire Radiative Power From the Combination of Polar-Orbiting and Geostationary Satellite Data During Wildfire Spread. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.

Xikun Hu, Wenlin Liu, HaoWen, Ka-Veng Yuen, Tian Jin, Alberto Costa Nogueira and Ping Zhong. (2024). AF-Net: An Active Fire Detection Model Using Improved Object-Contextual Representations on Unbalanced UAV Datasets. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.

W. Li, J. Sun, Z. Chen, K. Liu and Z. Zhang. (2023). Smoke and Flame Identification Method for the Entire Process of Grassland Fire Based on YOLOv5m-D and Static and Dynamic Characteristics. IEEE open Access journal.

X. Sun, Ning Li, Duoqi Chen, Guang Chen, Changjun Sun, Mulin Shi, Xuehong Gao, Kuo Wang And Ibrahim M. Hezam. (2024). A Forest Fire Prediction Model Based on Cellular Automata and Machine Learning. IEEE open Access journal.

Y. Yu, L. Liu, Z. Chang, Y. Li and K. Shi. (2024). Detecting Forest Fires in Southwest China From Remote Sensing Nighttime Lights Using the Random Forest Classification Model. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.

Ahmad Alkhatib, Khalid Mohammad Jaber, Hassan Alzo, Mohammad Abdallah and Mousa Salah. (2024). Exploring Progress in Forest Fire Detection, Prediction, and Behavior: An In-Depth Survey. International Journal of Computing and Digital Systems.

Swaraj Singh, Manish Jha, Dr.Bijal Talati and Aditi Jaiswal. (2024). Forest fire detection using CNN.

ResearchGate Xufeng Lin , Zhongyuan Li, Wenjing Chen, Xueying Sun and Demin Gao. (2023). Forest Fire Prediction Based on Long- and Short-Term Time-Series Network. MDPI. Forests