Project Report

PROJECT NAME: Emerging Methods for Early Detection Of Forest Fires

TEAM ID : PNT2022TMID01211

The Report is submitted by:

THE TEAM :

TEJA YANDAPALLI – TEAM LEAD

POOJA YANDAPALLI

R. YASHIKA

YOGALAKSHMI

1.INTRODUCTION

1.1 Project overview:

Forest fires are a major environmental issue, creating economic and ecological damage while endangering human lives. There are typically about 100,000 wildfires in the United States every year. Over 9 million acres of land have been destroyed due to treacherous wildfires. It is difficult to predict and detect Forest Fire in a sparsely populated forest area and it is more difficult if the prediction is done using ground-based methods like Camera or Video-Based approach. Satellites can be an important source of data prior to and also during the Fire due to its reliability and efficiency. The various real-time forest fire detection and prediction approaches, with the goal of informing the local fire authorities.

1.2 Purpose:

Monitoring of the potential risk areas and an early detection of fire can **significantly shorten the reaction time and also reduce the potential damage as well as the cost of fire fighting**.By detecting a fire quickly and accurately (i.e., by not sacrificing speed or causing false alarms) and providing early warning notification, a fire-detection system can **limit the emission of toxic products created by combustion, as well as global-warming gases produced by the fire itself**.**The forest fires destroys the wildlife habitat, damages the environment, affects the climate, spoils the biological properties of the soil, etc**. So the forest fire detection is a major issue in the present decade. At the same time the forest fire have to be detected as fast as possible.

2. Literature survey:

2.1 Existing problem:

In recent history and even the present day, several forest fire detection methods have been implemented, such as watchtowers, satellite image processing methods, optical sensors, and digital camera-based methods[2], although there are many drawbacks, such as inefficiency, power consumption, latency, accuracy and implementation costs. To address these drawbacks, a forest fire detection system using wireless sensor networks is proposed in this paper.

2.2 References:

[1] A. A. A. Alkhatib, “A Review on Forest Fire Detection Techniques:,” http://dx.doi.org/10.1155/2014/597368, vol. 2014, Mar. 2014, doi: 10.1155/2014/597368.
[2] “Copernicus: Mediterranean region evolves into wildfire hotspot, while fire intensity reaches new records in Turkey | Copernicus.” https://atmosphere.copernicus.eu/copernicusmediterranean-region-evolves-wildfirehotspot-while-fire-intensity-reaches-newrecords (accessed Jan. 01, 2022).
[3] “From wildfires to floods, the Mediterranean bears the brunt of catastrophic climate change.” https://www.rfi.fr/en/international/20210813- from-wildfires-to-floods-the-mediterraneanbears-the-brunt-of-catastrophic-climatechange (accessed Sep. 18, 2021).
[4] “Fires, forests and the future: a crisis raging out of control? | WWF.” https://wwf.panda.org/discover/our_focus/fore sts_practice/forest_publications_news_and_reports/fires_forests/ (accessed Sep. 18, 2021).
[5] R. Szeliski, Computer vision: algorithms and applications. Springer Science \& Business Media, 2010.
[6] M. T. Islam, B. M. N. Karim Siddique, S. Rahman, and T. Jabid, “Image Recognition with Deep Learning,” 2018 Int. Conf. Intell. Informatics Biomed. Sci. ICIIBMS 2018, pp. 106–110, Nov. 2018, doi: 10.1109/ICIIBMS.2018.8550021.
[7] L. Perez and J. Wang, “The Effectiveness of Data Augmentation in Image Classification using Deep Learning,” Dec. 2017, Accessed: Jan. 12, 2022. [Online]. Available: https://arxiv.org/abs/1712.04621v1.
[8] A. Mikołajczyk and M. Grochowski, “Data augmentation for improving deep learning in image classification problem,” 2018 Int. Interdiscip. PhD Work. IIPhDW 2018, pp. 117–122, Jun. 2018, doi: 10.1109/IIPHDW.2018.8388338. [9] A. Kherraki, M. Maqbool, and R. El Ouazzani, “Traffic Scene Semantic Segmentation by Using Several Deep Convolutional Neural Networks,” 2021 3rd IEEE Middle East North

[10] Uratek, “Urafire is a software for fire forest early detection,” 2013, http://www.uratek.com/applications.php?5. [11] P. Guillemant and J. Vicente, “Real-time identification of smoke images by clustering motions on a fractal curve with a temporal embedding method,” France, 2001.
[12] NGNS, “Forest Fire Finder,” 2013, http://www.ngns-is.com/ eng/FFF eng.html.
[13] D. Schroeder, “Evaluation of Three Wildfire Smoke Detection Systems,” FERIC5, 2004.
[14] ALASIA Marketing, “Fire Hawk ForestWatch,” 2013, http:// www.firehawk.co.za/.
[15] FireWatch, “An Early Warning System for Forest Fires, successfully in the global use,” 2013, http://www.fire-watch.de/systemoverview.
[16] J. Lloret, M. Garcia, D. Bri, and S. Sendra, “A wireless sensor network deployment for rural and forest fire detection and verification,” Sensors, vol. 9, no. 11, pp. 8722–8747, 2009.
[17] B. Son, Y. Her, and K. Kim, “A Design and Implementation of Forest-Fires Surveillance System based on Wireless Sensor Networks for South Korea Mountains,” International Journal of Computer Science and Network Security, vol. 6, no. 9, pp. 124– 130, 2006
[18] Rong, J., et al., Fire flame detection based on GICA and target tracking. Optics & Laser Technology, 2013. 47: p. 283-291.
[19] Celik, T. and H. Demirel, Fire detection in video sequences using a generic color model. Fire Safety Journal, 2009. 44(2): p. 147-158.
[20] Chunyu, Y., et al., Video fire smoke detection using motion and color features. Fire

technology, 2010. 46(3): p. 651-663.
[21] Günay, O., et al., Video based wildfire detection at night. Fire Safety Journal, 2009. 44(6): p. 860-868.
[22] Zhou, Z., et al., Wildfire smoke detection based on local extremal region segmentation and surveillance. Fire Safety Journal, 2016. 85: p. 50-58.
[23] Gubbi, J., S. Marusic, and M. Palaniswami, Smoke detection in video using wavelets and support vector machines. Fire Safety Journal, 2009. 44(8): p. 1110-1115.
[24] Millan-Garcia, L., et al., An early fire detection algorithm using IP cameras. Sensors, 2012. 12(5): p. 5670-5686. [25] Khatami, A., et al., A new PSO-based approach to fire flame detection using K-Medoids clustering. Expert Systems with Applications, 2017. 68: p. 69-80.
[26] Hackner, A., et al., Heterogeneous sensor arrays: Merging cameras and gas sensors into innovative fire detection systems. Sensors and Actuators B: Chemical, 2016. 231: p. 497-505.
[27] Ko, B.C., K.-H. Cheong, and J.-Y. Nam, Fire detection based on vision sensor and support vector machines. Fire Safety Journal, 2009. 44(3): p. 322-329.
[28] Muhammad, K., J. Ahmad, and S.W. Baik, Early Fire Detection using Convolutional Neural Networks during Surveillance for Effective Disaster Management. Neurocomputing, 2017.
[29] Zhang, D., et al. Image based forest fire detection using dynamic characteristics with artificial neural networks. in Artificial Intelligence, 2009. JCAI'09. International Joint Conference on. 2009. IEEE
[30] Soliman, H., K. Sudan, and A. Mishra. A smart forest-fire early detection sensory system: Another approach of utilizing wireless sensor and neural networks. in Sensors, 2010 IEEE. 2010. IEEE.
[31] Kolesenkov, A., et al. Anthropogenic situation express monitoring on the base of the fuzzy neural networks. in Embedded Computing (MECO), 2014 3rd Mediterranean Conference on. 2014. IEEE.
[32] Chauhan, A., S. Semwal, and R. Chawhan. Artificial neural network-based forest fire detection system using wireless sensor network. in India Conference (INDICON), 2013 Annual IEEE. 2013. IEEE.
[33] Satir, O., S. Berberoglu, and C. Donmez, Mapping regional forest fire probability using artificial neural network model in a Mediterranean forest ecosystem. Geomatics, Natural Hazards and Risk, 2016. 7(5): p. 1645-1658.
[34] Filizzola, C., et al., RST-FIRES, an exportable algorithm for earlyfire detection and monitoring: Description, implementation, and field validation in the case of the MSG-SEVIRI sensor. Remote Sensing of Environment, 2017. 192: p. e2-e25.
[35] Coppo, P., Simulation of fire detection by infrared imagers from geostationary satellites. Remote Sensing of Environment, 2015. 162: p. 84-98.
[36] Koltunov, A., et al., The development and first validation of the GOES Early Fire Detection (GOES-EFD) algorithm. Remote Sensing of Environment, 2016. 184: p. 436-453.
[37] Oliva, P. and W. Schroeder, Assessment of VIIRS 375 m active fire detection product for direct burned area mapping. Remote Sensing of Environment, 2015. 160: p. 144-155.
[38] Krüll, W., et al., Early forest fire detection and verification using optical smoke, gas and microwave sensors. Procedia Engineering, 2012. 45: p. 584-594.
[39] Dennison, P.E. and D.A. Roberts, Daytime fire detection using airborne hyperspectral data. Remote Sensing of Environment, 2009. 113(8): p. 1646-1657.
[40] Allison, R.S., et al., Airborne optical and thermal remote sensing for wildfire detection

and monitoring. Sensors, 2016. 16(8): p. 1310.
[41] Cruz, H., et al., Efficient forest fire detection index for application in unmanned aerial systems (UASs). Sensors, 2016. 16(6): p. 893.
[42] Tomkins, L., et al., Use of night vision goggles for aerial forest fire detection. International journal of wildland fire, 2014. 23(5): p. 678-685.
[43] Ko, B., J.-H. Jung, and J.-Y. Nam, Fire detection and 3D surface reconstruction based on stereoscopic pictures and probabilistic fuzzy logic. Fire Safety Journal, 2014. 68: p. 61-70.
[44] Bolourchi, P. and S. Uysal. Forest fire detection in wireless sensor network using fuzzy logic. in Computational Intelligence, Communication Systems and Networks (CICSyN), 2013 Fifth International Conference on. 2013. IEEE.
[45] Bousack, H., et al., Towards Improved Airborne Fire Detection Systems Using Beetle Inspired Infrared Detection and Fire Searching Strategies. Micromachines, 2015. 6(6): p. 718-746.
[46] Sahin, Y.G. and T. Ince, Early forest fire detection using radioacoustic sounding system. Sensors, 2009. 9(3): p. 1485-1498.
[47] Kumar, S., et al. Optimal sleep-wakeup algorithms for barriers of wireless sensors. in Broadband Communications, Networks and Systems, 2007. BROADNETS 2007. Fourth International Conference on. 2007. IEEE
[48] G. D. Georgiev, G. Hristov, P. Zahariev, and D. Kinaneva, “Forest Monitoring System for Early Fire Detection Based on Convolutional Neural Network and UAV imagery,” 28th Natl. Conf. with Int. Particip. TELECOM 2020 - Proc., pp. 57–60, Oct. 2020, doi: 10.1109/TELECOM50385.2020.9299566.
[49] A. Rajagopal et al., “Fine-Tuned Residual Network-Based Features with Latent Variable Support Vector Machine-Based Optimal Scene Classification Model for Unmanned Aerial Vehicles,” IEEE Access, vol. 8, pp. 118396– 118404, 2020, doi: 10.1109/ACCESS.2020.3004233.
[50] Y. Zhao, J. Ma, X. Li, and J. Zhang, “Saliency Detection and Deep Learning-Based Wildfire Identification in UAV Imagery,” Sensors 2018, Vol. 18, Page 712, vol. 18, no. 3, p. 712, Feb.2018, doi: 10.3390/S18030712
. [51] J. Zheng, C. Xianbin, Z. Baochang, Y. Huang, and Y. Hu, “Bi-heterogeneous Convolutional Neural Network for UAV-based dynamic scene classification,” ICNS 2017 - ICNS CNS/ATM Challenges UAS Integr., Aug. 2017, doi: 10.1109/ICNSURV.2017.8011932.
[52] S. Wang, J. Zhao, N. Ta, X. Zhao, M. Xiao, and H. Wei, “ORIGINAL RESEARCH PAPER A real - time deep learning forest fire monitoring algorithm based on an improved Pruned + KD model,” no. 0123456789, 2021, doi: 10.1007/s11554-021-01124-9
[53] A. Shamsoshoara, F. Afghah, A. Razi, L. Zheng, P. Z. Fulé, and E. Blasch, “Aerial imagery pile burn detection using deep learning: The FLAME dataset,” Comput. Networks, vol. 193, p. 108001, Jul. 2021, doi: 10.1016/J.COMNET.2021.108001.

[54] Z. Jiao et al., “A YOLOv3-based Learning Strategy for Real-time UAV-based Forest Fire Detection,” Proc. 32nd Chinese Control Decis. Conf. CCDC 2020, pp. 4963–4967, Aug. 2020, doi: 10.1109/CCDC49329.2020.9163816. [55] H. Wu, H. Li, A. Shamsoshoara, A. Razi, and F. Afghah, “Transfer Learning for Wildfire Identification in UAV Imagery,” 2020 54th Annu. Conf. Inf. Sci. Syst. CISS 2020, Mar. 2020, doi: 10.1109/CISS48834.2020.1570617429.
[56] G. V. S. Lohit and D. Bisht, “Seed Dispenser using Drones and Deep Learning Techniques for Reforestation,” Proc. - 5th Int. Conf. Comput. Methodol. Commun. ICCMC 2021, pp. 1275–1283, Apr. 2021, doi: 10.1109/ICCMC51019.2021.9418227.

[57] G. V. S. Lohit, "Reforestation Using Drones and Deep Learning Techniques," 2021 7th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2021, pp. 847–852, Mar. 2021, doi: 10.1109/ICACCS51430.2021.9442053.
[58] D. Q. Tran, M. Park, D. Jung, and S. Park, "Damage-Map Estimation Using UAV Images and Deep Learning Algorithms for Disaster Management System," Remote Sens. 2020, Vol. 12, Page 4169, vol. 12, no. 24, p. 4169, Dec. 2020, doi: 10.3390/RS12244169.
[59] S. Zheng, W. Wang, Z. Liu, and Z. Wu, "Forest Farm Fire Drone Monitoring System Based on Deep Learning and Unmanned Aerial Vehicle Imagery," Math. Probl. Eng., vol. 2021, pp. 1– 13, Nov. 2021, doi: 10.1155/2021/3224164. [60] Z. Jiao et al., "A Deep learning based forest fire detection approach using uav and yolov3," 1st Int. Conf. Ind. Artif. Intell. IAI 2019, Jul. 2019, doi: 10.1109/ICIAI.2019.8850815.

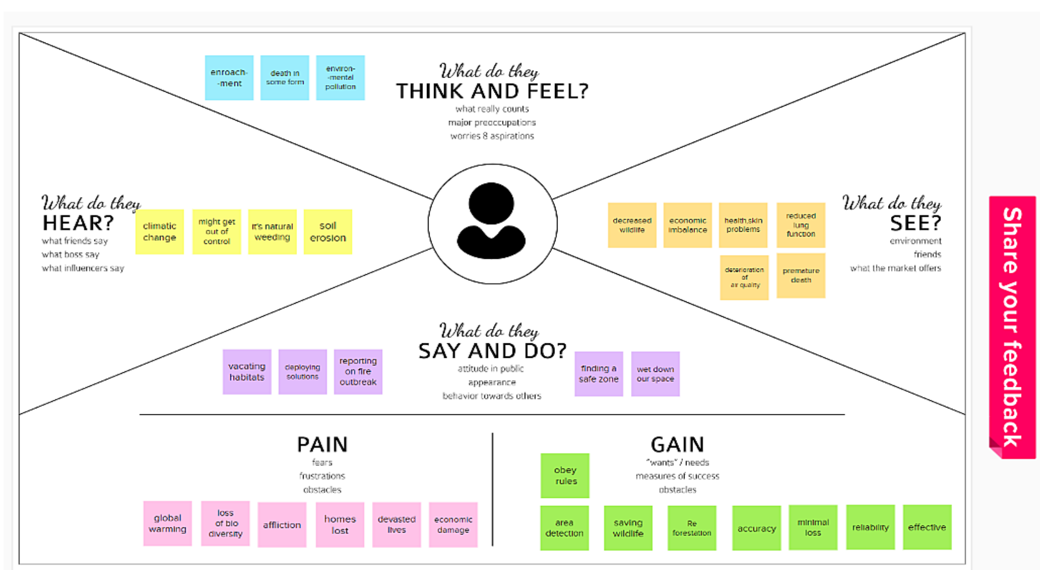
2.3 Problem statement:

Forest fires are a major environmental issue, creating economic and ecological damage while endangering human lives. There are typically about 100,000 wildfires in the United States every year. Over million acres of land have been destroyed due to treacherous wildfires. It is difficult to predict and detect Forest Fire in a sparsely populated forest area and it is more difficult if the prediction is done using ground-based methods like Camera or Video-Based approach and also using satellites.

3.IDEATION & PROPOSED SOLUTION

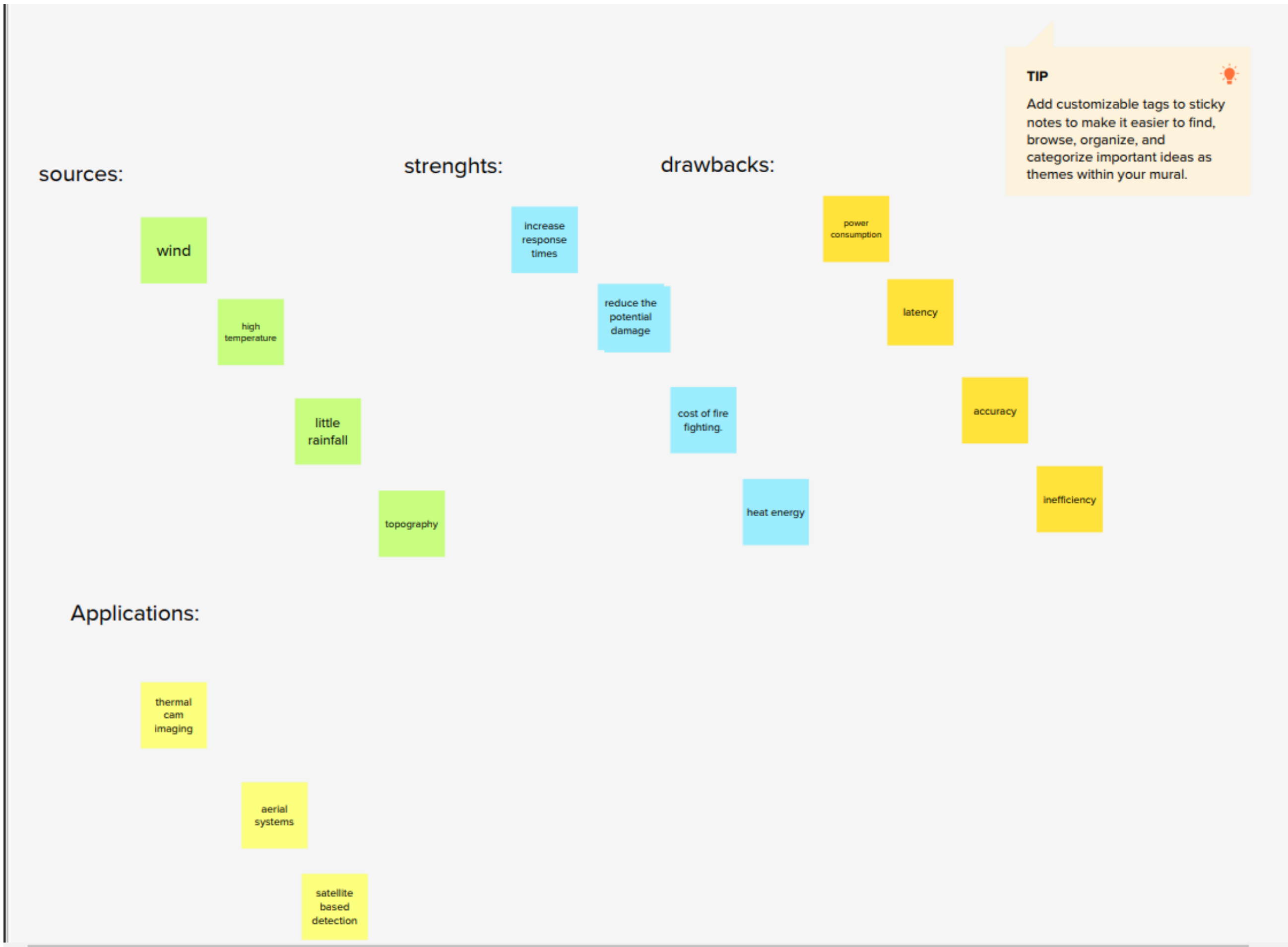
-

3.1 Empathy canvas map:



3.2 Brainstorming:

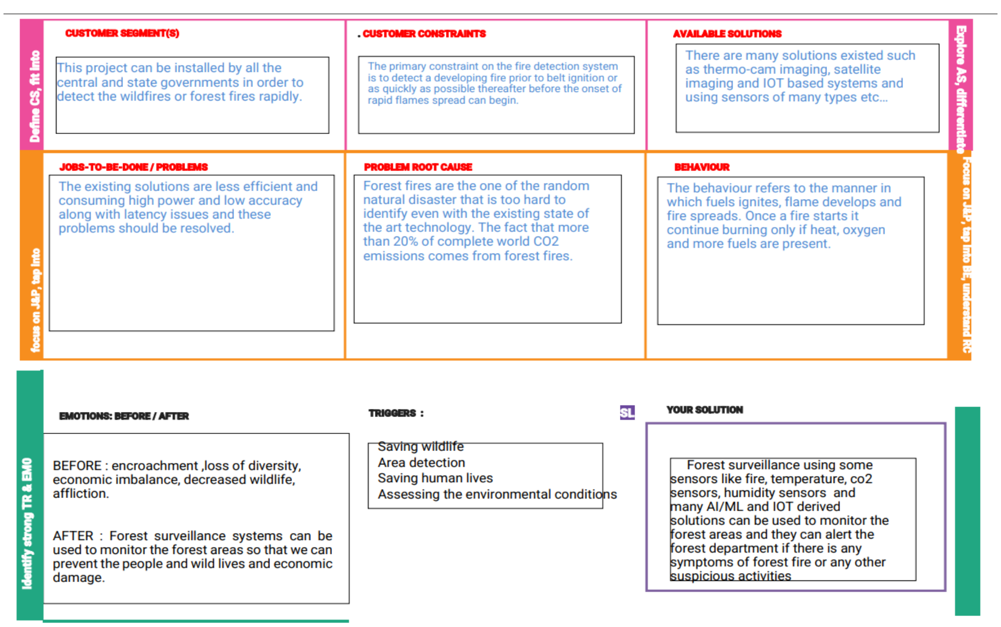
3.2 Group ideas & prioritisation:



3.3 Proposed solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To solve the problem arising in forest fires can be solved by banning the clear-cutting of forests, by following government regulations,by reducing consumption of deforestation prone products and by educating others etc.,
2.	Idea / Solution description	Artificial neural networks are used to predict forest fires.It can predict new inputs through knowledge collected by self-learning methods.
3.	Novelty / Uniqueness	Artificial neural networks are created to digitally mimic the human brain and these networks can be used to design the next generation of computers.
4.	Social Impact / Customer Satisfaction	The forest fires create significant impacts on the mental and emotional health of survivors,including increased anxiety,depression etc.,
5.	Business Model (Revenue Model)	The annual losses from forest fires in India for the entire country have been moderately estimated at Rs.440 crores(US$107 million).To overcome this ,we can use Artificial neural networks.
6.	Scalability of the Solution	With the advancements in wireless technology and digital electronics some tiny devices have started to be used in numerous areas in daily life which are capable of sensing computing and communicating.Development of MEMS wireless networks systems are expected to be in wide use.

3.4 Proposed solution fit:



4.REQUIREMENT ANALYSIS

4.1 Functional requirements:

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Sensors and detectors	The sensors for Temperature,Humidity,Heat etc..,should be installed for detection.
FR-2	cameras	Cameras should be placed to capture the scenario of wild fire.
FR-3	Data collection	The data shoul be collected and stored in databases for assessments.
FR-4	Data Analyst	The data so collected should be assesed to proceed for further actions.
FR-5	Alarming system	An alarming should be installed such that it will alert the corresponding departments to take action.
FR-6	Prediction systems	Sometimes an prediction system place a key rolesin detection of forest fires and can reduce the damage that can be occurred.

4.2 Non -functional requirements:

Non-functional Requirements:

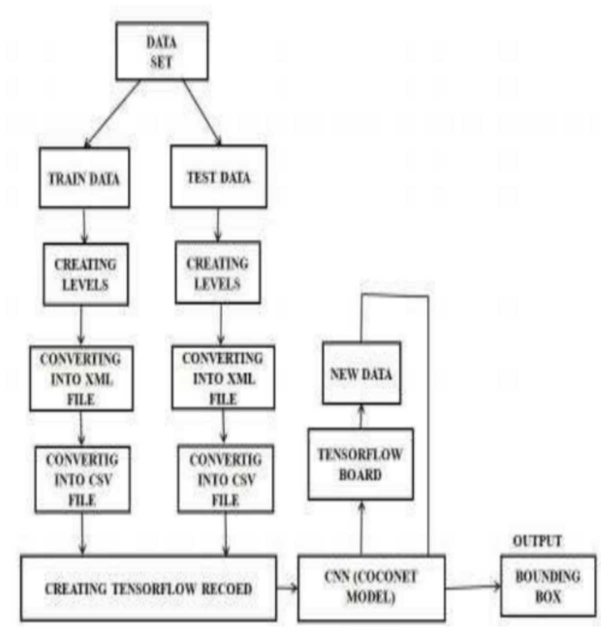
Following are the non-functional requirements of the proposed solution.

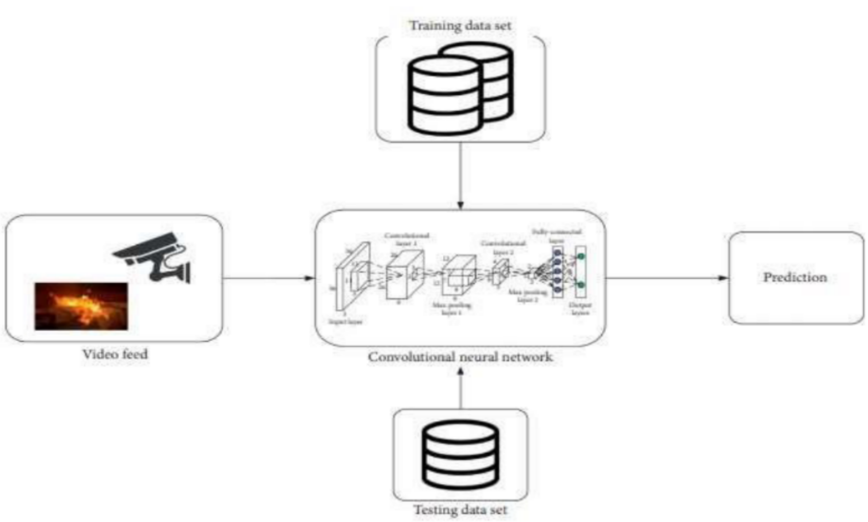
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Usage of sensors and cameras to detect and sense the wild fire.
NFR-2	**Security**	Sensors and all ather requirements are installed for performing in all extreme conditions.
NFR-3	**performance**	All the sensing systems installed will detect and results are sent to the end users with much accuracy.
NFR-4	**Reliability**	The technologies used and the sensing and detecting systems will run to long time and desidned to be reliable in manner.
NFR-5	**Availability**	The technological necessities so far used are available in the current modern world and are used by the various developers more oftenly.
NFR-6	**Scalability**	The single sensing systems onc einstalled will cover the one fourth of an forest as fire eventually spread eventhough..,

5.PROJECT DESIGN

5.1 DataFlow diagrams:

Data Flow Diagrams:



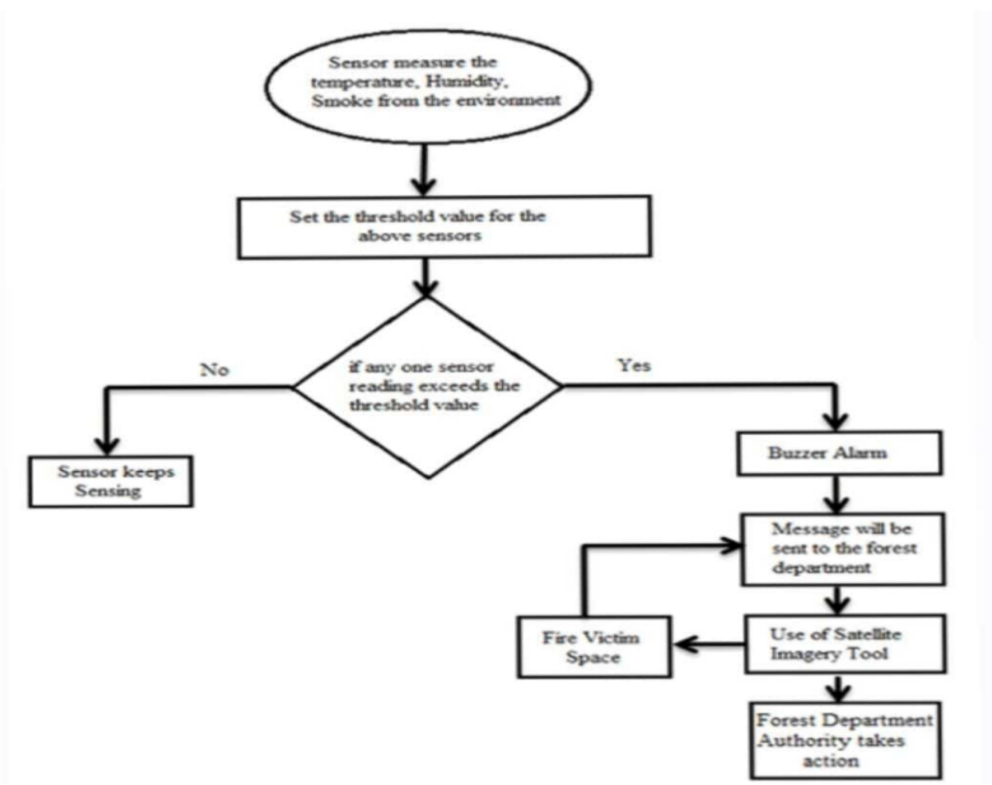


5.2 Technical architecture:

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	The user interacts with the application.	Python
2.	Application Logic-1	The Logic for performance of the process to execute the desired output	Python
3.	Application Logic-2	IBM is used to deploy the CNN model	IBM service
4.	Database	(Pictures) Composite Data Type	MySQL.
5.	Cloud Database	Database Service on Cloud	IBM Cloudant etc.
6.	File Storage	Files like dataset for the use of training and testing can be stored in local system.	IBM Block Storage or Other Storage Service or Local Filesystem
7.	External API-1	Purpose of External API used	IBM API, twilio rest API
8.	Machine Learning Model	Purpose of Machine Learning Model is it allows the user to feed a computer algorithm an immense amount of data and have the computer analyze and make data-driven recommendations and decisions based on only the input data.	Object Recognition Model, CNN
9.	Infrastructure (Server / Cloud)	Application Deployment on Cloud	IBM Cloud

5.2 Solution architecture:



5.3 User stories:

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Data Collection	Collect the data	USN-1	It is necessary to collect the data of the forest which includes temperature , humidity, soil erosion ,wind and rainfall of the forest	It is necessary to collect the right data else the prediction may become wrong	Medium	Sprint-1
Technologists	Assessing technology	USN-2	Identify algorithms that can be used for prediction	To collect the algorithm to identify the accuracy level of each algorithms	High	Sprint-2
Data Analysts	To check algorithm works fare enough in all situations	USN-3	Identify the accuracy of each algorithms	Accuracy of each algorithm-calculated so that it is easy to obtain the most accurate output	Low	Sprint-2
	To correlate with previous data	USN-4	Evaluate the Dataset	Data is evaluated before processing	Medium	Sprint-1
		USN-5	Identify accuracy,precision,recall of each algorithms	These values are important for obtaining the right output	High	Sprint-3
End users	To get the accomplished outcome of the project	USN-6	Outputs from each algorithm are obtained	It is highly used to predict the effect and to take precautionary measures.	High	Sprint-4

6.PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION:

Activity List:

ACTIVITY LIST

Activity Number	Activity	Sub Activity	Assigned To	Status
1.	PROJECT OBJECTIVES		All Members	Completed
2.	PROJECT FLOW		All Members	Completed
3.	PRE-REQUISITES		All Members	Completed
4.	DATA COLLECTION	4.1 Download the Dataset	TEJA YANDAPALLI	Completed
5.	IMAGEPREPROCESSING	5.1 Import the ImageDataGenerator Library. 5.2 Define the Parameters/Arguments forImageDataGenerator class. 5.3 Applying ImageDataGenerator Functionality totrainset andtestset.	TEJA YANDAPALLI	Completed
6.	MODEL BUILDING	6.1 Importing the model buildinglibraries. 6.2 Initializing themodel. 6.3 Adding CNNlayers. 6.4 Adding denselayers. 6.5 Configuringthe	TEJA YANDAPALLI	Completed

		learning process. 6.6 Training themodel. 6.7 Saving themodel. 6.8 Predictions.		
7.	VIDEO ANALYSIS	7.1 OpenCV for video processing. 7.2 Creating an account in Twilioservice. 7.3 Sendingalert message.	TEJA YANDAPALLI	Completed
8.	TRAIN CNN MODELON IBM	8.1 Train image classificationmodel. 8.2 Register for IBM cloud.	TEJA YANDAPALLI	In Progress
9.	IDEATION PHASE	9.1 LiteratureReview. 9.2 Empathymap. 9.3 Ideation.	All Members	Completed
10.	PROJECT DESIGNPHASE – I	10.1 ProposedSolution. 10.2 Problem solutionfit. 10.3 Solutio n Architecture.	All Members	Completed
11.	PROJECT DESIGNPHASE -II	11.1 Customerjourney. 11.2 Functiona l requirement. 11.3 Data flowDiagrams. 11.4 Technolog y Architecture.	All Members	Completed
12.	PROJECT PLANNINGPHASE	12.1 Preparemilestone and activitylist. 12.2 Sprint deliveryplan.	All Members	In Progress
13.	PROJECTDEVE LOPMENTPHAS E	13.1 Project development-Delivery of Sprint-1. 13.2 Project development-Delivery of Sprint-2. 13.3 Project development-Delivery of Sprint-3. 13.4 Project development-Delivery of Sprint-4.	All Members	In Progress

Milestone List:

MILESTONE LIST

Milestone Name	Milestone Number	Description	Mandatory	
Project Objectives	M-01	We will be able to learn to prepare dataset, image processing, working with CNN layers, read images using OpenCV and CNN for computer vision AI	Yes	-
Project Flow	M-02	A project management process flowchart is a graphical aid, designed to visualize the sequence of steps to be followed throughout the project management process	Yes	
Pre-Requisites	M-03	To complete this project we should have known following project such as Keras, Tensorflow, Python ,Anaconda, OpenCV, Flask, Scikit-learn etc…	Yes	
Prior Knowledge	M-04	One should have knowledge on the Supervised Learning ,CNN and Regression Classification and Clustering, ANN	Yes	
Data collection	M-05	We can collect dataset from different open sources like kaggle.com, UCI machine learning etc	Yes	
Image Preprocessing	M-06	Importing the ImageDataGenerator libraries, Define Parameters/Arguments for ImageDataGenerator class, Applying Image Data Generator Functionality to trainset and testset	Yes	
Model Building	M-07	Importing the model building libraries, Initializing the model, Adding CNN layers, Adding Dense layers, Configuring the learning Process,Train the model,Save the model,Predictions.	Yes	
Video Analysis	M-08	Opencv for video processing, creating an account in twilio service and sending alert message	Yes	
Train CNN model	M-09	Register for IBM Cloud and train Image Classification Model	Yes	
Ideation Phase	M-10	Prepare Literature Survey on the selected Project and Information Gathering, empathy map and ideation	Yes	
Project Design Phase-I	M-11	Prepare Proposed solution , problem-solution fit and Solution Architecture	Yes	
Project Design Phase-II	M-12	Prepare Customer journey ,functional requirements,Data flow diagram and Technology Architecture	Yes	
Project Planning Phase	M-13	Prepare Milestone list , Activity list and Sprint Delivery Plan	Yes	
Project DevelopmentPhase	M-14	Project Development delivery of Sprint 1, Sprint 2, Sprint 3, Sprint 4	Yes	

6.2 SPRINT DELIVERY SCHEDULE:

Sprint Delivery Plan:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	20	High	TEJA YANDAPALLI POOJA YANDAPALLI YASHIKA.R YOGALAKSHMI.V
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application usage.	20	High	TEJA YANDAPALLI POOJA YANDAPALLI YASHIKA.R YOGALAKSHMI.V
Sprint-2	Input	USN-3	When ever the fire is detected ,the information is given to the database.	20	High	TEJA YANDAPALLI POOJA YANDAPALLI YASHIKA.R YOGALAKSHMI.V
Sprint-2		USN-4	When it is the wildfire then the alarming system is activated.	20	High	TEJA YANDAPALLI POOJA YANDAPALLI YASHIKA.R YOGALAKSHMI.V
Sprint-3	Output	USN-5	And the alarm also sent to the corresponding departments and made them know that the wildfire is erupted.	20	High	TEJA YANDAPALLI POOJA YANDAPALLI YASHIKA.R YOGALAKSHMI.V
Sprint-4	Action	USN-6	Required actions will be taken in order to controlled erupted wildfire by reaching as early as possible to the destination with the help of detecting systems.	20	High	TEJA YANDAPALLI POOJA YANDAPALLI YASHIKA.R YOGALAKSHMI.V

Project Tracking:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

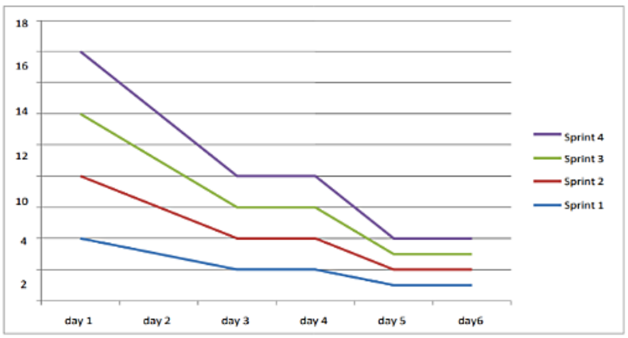
6.3 REPORTS FROM JIRA:

Sprint Schedule Chart:

Burndown Chart:

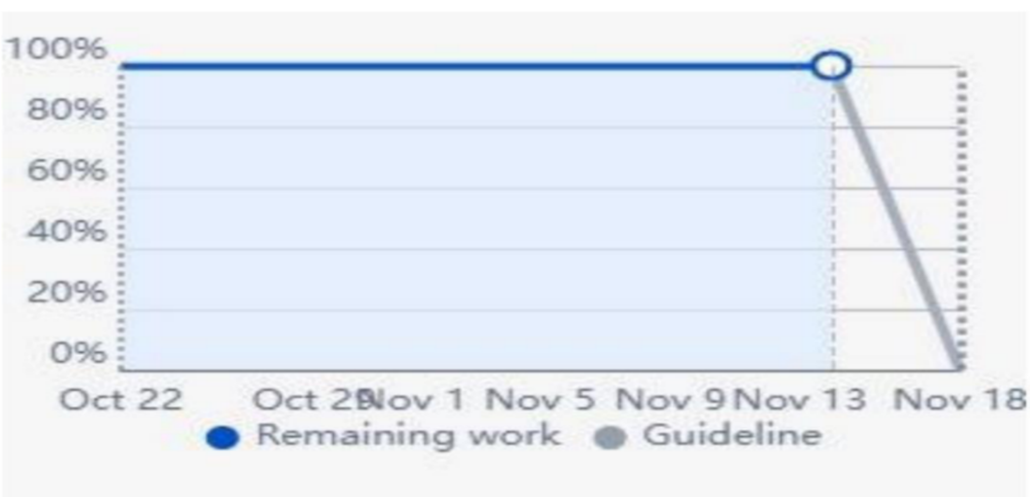
A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

SPRINT SCHEDULE CHART:

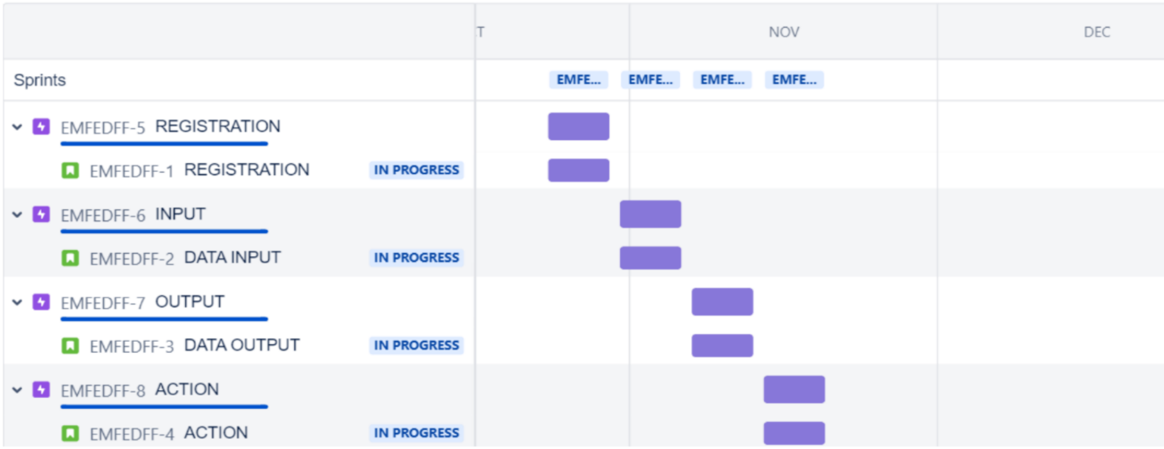


Burndown chart:

SPRINT BURNDOWN CHART:



JIRA FILES



7.CODING & SOLUTIONING:

7.1 FEATURE1:

Dataset:

https://www.kaggle.com/arbethi/forest-fire?select=Dataset

7.2 FEATURE2:

Image Preprocessing:

```
# -*- coding: utf-8 -*-
"""imageprocessing.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1YG_pSrXMz1wAjQ7Oz1UqrUNX5FbNEqXP
"""


# split imbalanced dataset into train and test sets without stratification
from collections import Counter
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
# create dataset
X, y = make_classification(n_samples=100, weights=[0.94], flip_y=0, random_state=1)
print(Counter(y))
# split into train test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.50, random_state=1)
print(Counter(y_train))
print(Counter(y_test))

import keras
from keras.preprocessing.image import ImageDataGenerator

train_datatype=ImageDataGenerator(rescale=1./255,shear_range=0.2,rotation_range=180,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)

import keras
from keras.preprocessing.image import ImageDataGenerator
```

7.3 DATABASE SCHEMA :

Model building:

Model Building:

import model building libraries

In [ ]:
```
from keras.models import Sequential
from keras.layers import Convolution2D,MaxPooling2D,Dense,Flatten
import warnings
warnings.filterwarnings('ignore')
```

Initializing the Model

Adding Convolutional Layer

Adding Dense Layers

In [ ]:
```
model = Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(units=256,activation='relu'))
model.add(Dense(units=1,activation='sigmoid'))
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 126, 126, 32)      896

 max_pooling2d (MaxPooling2D  (None, 63, 63, 32)       0
 )

 flatten (Flatten)           (None, 127008)            0

 dense (Dense)               (None, 256)               32514304

 dense_1 (Dense)             (None, 1)                 257

=================================================================
Total params: 32,515,457
Trainable params: 32,515,457
Non-trainable params: 0
```

Configuring The Learning Process

In [ ]:
```
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy','mse'])
```

Training the Model

In [ ]:
```
y = model.fit_generator(train,steps_per_epoch=14,epochs=15,validation_data=test,validation_steps=4)
```

```
Epoch 1/15
14/14 [==============================] - 24s 2s/step - loss: 3.1720 - accuracy: 0.6032 - mse: 0.3361 - val_loss: 0.3162 - val_accuracy: 0.8512 - val_ms
e: 0.1027
Epoch 2/15
14/14 [==============================] - 20s 1s/step - loss: 0.4173 - accuracy: 0.7844 - mse: 0.1395 - val_loss: 0.2734 - val_accuracy: 0.8512 - val_ms
e: 0.0937
Epoch 3/15
14/14 [==============================] - 21s 1s/step - loss: 0.2720 - accuracy: 0.8739 - mse: 0.0865 - val_loss: 0.1265 - val_accuracy: 0.9669 - val_ms
e: 0.0358
Epoch 4/15
14/14 [==============================] - 20s 1s/step - loss: 0.2030 - accuracy: 0.9128 - mse: 0.0630 - val_loss: 0.0851 - val_accuracy: 0.9752 - val_ms
e: 0.0238
Epoch 5/15
14/14 [==============================] - 21s 2s/step - loss: 0.1813 - accuracy: 0.9197 - mse: 0.0555 - val_loss: 0.0773 - val_accuracy: 0.9835 - val_ms
e: 0.0209
Epoch 6/15
14/14 [==============================] - 22s 2s/step - loss: 0.1876 - accuracy: 0.9128 - mse: 0.0615 - val_loss: 0.0866 - val_accuracy: 0.9752 - val_ms
e: 0.0247
Epoch 7/15
14/14 [==============================] - 20s 1s/step - loss: 0.1649 - accuracy: 0.9335 - mse: 0.0512 - val_loss: 0.0932 - val_accuracy: 0.9835 - val_ms
e: 0.0253
Epoch 8/15
14/14 [==============================] - 20s 1s/step - loss: 0.1601 - accuracy: 0.9312 - mse: 0.0495 - val_loss: 0.0836 - val_accuracy: 0.9504 - val_ms
e: 0.0256
Epoch 9/15
14/14 [==============================] - 21s 1s/step - loss: 0.1584 - accuracy: 0.9220 - mse: 0.0497 - val_loss: 0.0558 - val_accuracy: 0.9835 - val_ms
e: 0.0145
Epoch 10/15
14/14 [==============================] - 20s 1s/step - loss: 0.1782 - accuracy: 0.9243 - mse: 0.0553 - val_loss: 0.0678 - val_accuracy: 0.9835 - val_ms
e: 0.0183
Epoch 11/15
```

predictions:

Predictions

In [ ]:

```
from keras.models import load_model
import cv2
import numpy as np
from PIL import Image
from keras.utils import img_to_array
model = load_model('/content/drive/MyDrive/IBM PROJECT/ffd_model.h5')
def prediction(img_path):
    i = cv2.imread(img_path)
    i = cv2.cvtColor(i, cv2.COLOR_BGR2RGB)
    img = Image.open(img_path)
    img = img.resize((128,128))
    x = img_to_array(img)
    x = np.expand_dims(x,axis=0)
    pred = model.predict(x)
    plt.imshow(i)
    print("%s"%("FOREST FIRE DETECTED! SMS SENT!" if pred==[[1.]] else "NO FOREST FIRE DETECTED"))
```

video processing:

In [3]:

```
#import opencv librariy
import cv2
#import numpy
import numpy as np
#import image function from keras
from keras.preprocessing import image
#import load_model from keras
from keras.models import load_model
#import client from twilio API
from twilio.rest import Client
#imort playsound package
from playsound import playsound
```

```
WARNING:playsound:playsound is relying on another python subprocess. Please use `pip install pygobject` if you want playsound to run more efficiently.
```

In [5]:

```
model = load_model(r'/content/drive/MyDrive/IBM PROJECT/ffd_model.h5')
#define video
video = cv2.VideoCapture(0)
#define the features
name = ['forest','with forest']
```

In [6]:

```
account_sid = 'AC381739ada733d1ba2fcee2548f10eef0'
auth_token = '928561042fab0f4b80ca038a4d7447f2'
client = Client(account_sid, auth_token)

message = client.messages \
    .create(
         body='FOREST FIRE IS DETECTED, STAY ALERT',
         from_='+14246228559',
          to = '+918919689576'
     )

print(message.sid)
```

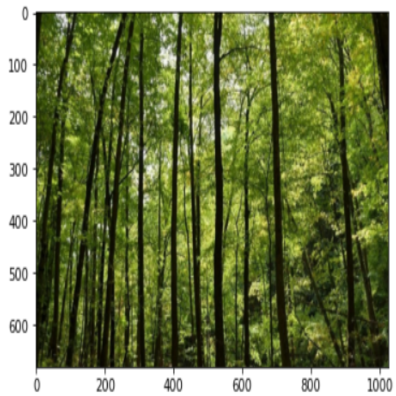
```
SMb82de2eb481d77eadb829c25c1b47246
```

8.Testing:

8.1 TEST CASES:

```
In [ ]: prediction(r'/content/drive/MyDrive/IBM PROJECT/Dataset/test_set/forest/0.48007200_1530881924_final_forest.jpg')
```

```
1/1 [==============================] - 0s 112ms/step
NO FOREST FIRE DETECTED
```



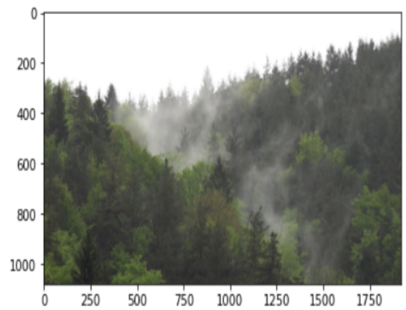
```
In [ ]: prediction(r'/content/drive/MyDrive/IBM PROJECT/Dataset/test_set/with fire/180802_CarrFire_010_large_700x467.jpg')
```

```
1/1 [==============================] - 0s 29ms/step
FOREST FIRE DETECTED! SMS SENT!
```

In [ ]:

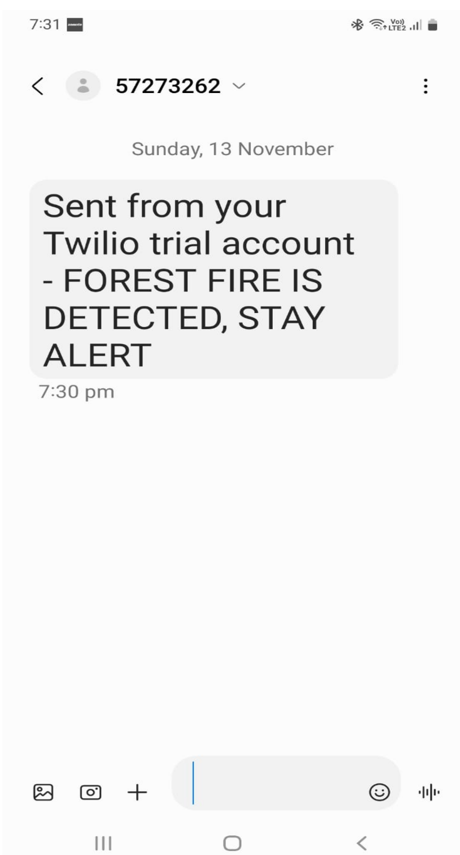
```
prediction(r'/content/drive/MyDrive/IBM PROJECT/Dataset/train_set/forest/with_fire (1).png')
```

```
1/1 [==============================] - 0s 27ms/step
NO FOREST FIRE DETECTED
```

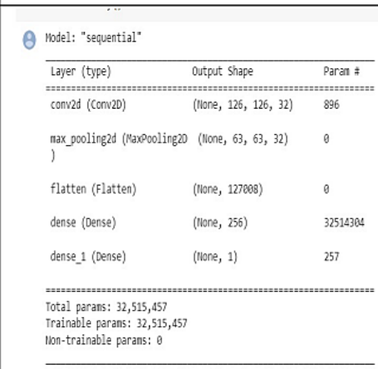
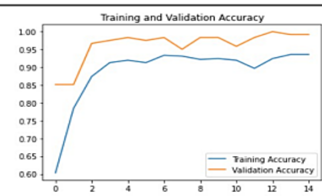


9.RESULTS:

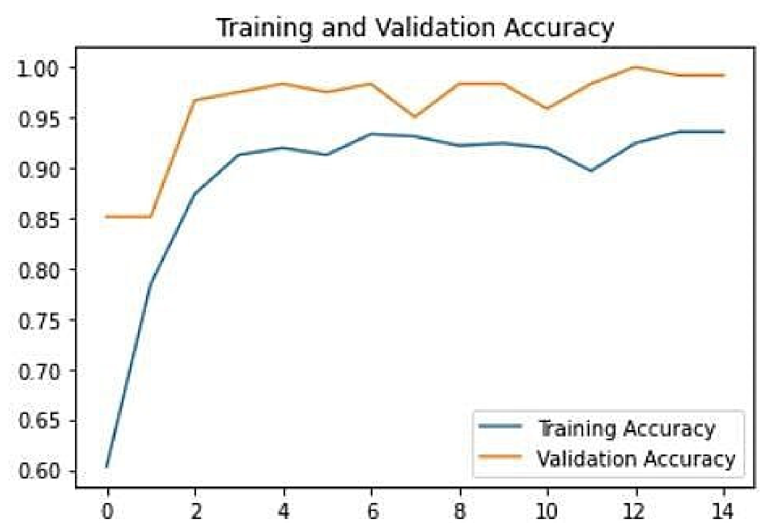
SENDING ALERT MESSAGE:



Performance testing:

S.No.	Parameter	Values	Screenshot
1.	Model Summary	**Model - Sequential model Layers: Conv2D-(None,62,62,32) MaxPooling2D-(None,31,31, 32) Flatten-(None,30752) Dense-(None,200) Dense_1 -(None,9)**	
2.	Accuracy	Training Accuracy - 0.94 Validation Accuracy -0.99	
3.	Confidence Score (Only Yolo Projects)	Class Detected - N/A Confidence Score - N/A	N/A

Accuracy:



110.Advantages:

1. It can also gauge temperature, humidity, and air pressure to create a climate map of the forest.
2. This map provides the means to assess the risk of fire.
3. It also serves to monitor the quality of the forest, prevent diseases and droughts, and optimize tree growth.
4. Fire detection systems increase response times, as they are able to alert the correct people in order to extinguish the fire.
5. This thus reduces the amount of damage to the property.
6. Fire detection systems can be connected to sprinklers that will automatically respond when a fire is detected.
7. A very huge area of forest is destroyed by fire every year.
8. Monitoring of the potential risk is sand an early detection of fire can significantly shorten the reaction time and

also reduce the potential damage as well as the cost of firefighting.

1. Benefits of Fire are the sounds and smells of fire bring different images to different people. ...
2. Cleaning the Forest Floor.
3. Fire removes low-growing underbrush, cleans the forest floor of debris, opens it up to sunlight, and nourishes the soil. ...
4. Providing Habitat. ...
5. Killing Disease. ...
6. New Generations.

10.Disadvantages:

1. At earlier times, forest fires were detected using watchtowers, which were not efficient because they were based on human observations.
2. In recent history and even the present day, several forest fire detection methods

have been implemented.

1. such as watchtowers, satellite image processing methods, optical sensors,

and digital camera-based methods.

1. although there are many drawbacks, such as inefficiency, power consumption, latency, accuracy and implementation costs.
2. Wildfires can disrupt transportation, communications, power and gas services, and water supply.
3. They also lead to a deterioration of the air quality, and loss of property, crops, resources, animals and people.
4. A person can burn.
5. a factory can be destroyed if fire catches.
6. chemical reactions may harm it if it is more supplied.
7. a gas can blast.
8. a nuclear factory can harm in large scale.

11.Conclusion:

1. From this project we came to the conclusion that the decision tree has a remarkable accuracy of 99% in predicting fires in forest areas. This reduces the chances of false alarm to a great extent.Our system is able to differentiate various forest fire scenarios, from initial case (no fire) to detection of fire,fairly accurately. It can accurately determine the growth of fire. This will help in early stages of fire detection and help to confine fire to limited areas before much damage occurs. The system will be very effective in preventing occurrence of false alarms. We aim to monitor the forests without constant human supervision.

12.Future Scope:

1. This project carries a broad perspective for the future.Moreover there is a need for great research to be done in this field in the coming years. In the future, our project can be extended towards finding an efficient way of localization of the fire, gravity of fire, direction of spread, area burnt and many more. In our experiment,the process of simulation of forest fire was done by burning the dried leaves directly. We could come up with ways to make this simulation more close to actual forest fires. Moreover, we can include the region specific meteorological data in the dataset for generating models for prediction. The nodes can be improved by making them efficient enough to have a better sensing distance, resistant to the harsh forest conditions, and energy efficient. A focused research can be done in devising ways of forest coverage with the nodes.

13.APPENDIX:

SOURCE CODE:-

Model Building:

import model building libraries

In [ ]:

```
from keras.models import Sequential
from keras.layers import Convolution2D,MaxPooling2D,Dense,Flatten
import warnings
warnings.filterwarnings('ignore')
```

Initializing the Model

Adding Convolutional Layer

Adding Dense Layers

In [ ]:

```
model = Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(units=256,activation='relu'))
model.add(Dense(units=1,activation='sigmoid'))
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 126, 126, 32)      896

 max_pooling2d (MaxPooling2D  (None, 63, 63, 32)       0
 )

 flatten (Flatten)           (None, 127008)            0

 dense (Dense)               (None, 256)               32514304

 dense_1 (Dense)             (None, 1)                 257

=================================================================
Total params: 32,515,457
Trainable params: 32,515,457
Non-trainable params: 0
```

Configuring The Learning Process

In [ ]:

```
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy','mse'])
```

Training the Model

In [ ]:

```
y = model.fit_generator(train,steps_per_epoch=14,epochs=15,validation_data=test,validation_steps=4)
```

```
Epoch 1/15
14/14 [==============================] - 24s 2s/step - loss: 3.1720 - accuracy: 0.6032 - mse: 0.3361 - val_loss: 0.3162 - val_accuracy: 0.8512 - val_ms
e: 0.1027
Epoch 2/15
14/14 [==============================] - 20s 1s/step - loss: 0.4173 - accuracy: 0.7844 - mse: 0.1395 - val_loss: 0.2734 - val_accuracy: 0.8512 - val_ms
e: 0.0937
Epoch 3/15
14/14 [==============================] - 21s 1s/step - loss: 0.2720 - accuracy: 0.8739 - mse: 0.0865 - val_loss: 0.1265 - val_accuracy: 0.9669 - val_ms
e: 0.0358
Epoch 4/15
14/14 [==============================] - 20s 1s/step - loss: 0.2030 - accuracy: 0.9128 - mse: 0.0630 - val_loss: 0.0851 - val_accuracy: 0.9752 - val_ms
e: 0.0238
Epoch 5/15
14/14 [==============================] - 21s 2s/step - loss: 0.1813 - accuracy: 0.9197 - mse: 0.0555 - val_loss: 0.0773 - val_accuracy: 0.9835 - val_ms
e: 0.0209
Epoch 6/15
14/14 [==============================] - 22s 2s/step - loss: 0.1876 - accuracy: 0.9128 - mse: 0.0615 - val_loss: 0.0866 - val_accuracy: 0.9752 - val_ms
e: 0.0247
Epoch 7/15
14/14 [==============================] - 20s 1s/step - loss: 0.1649 - accuracy: 0.9335 - mse: 0.0512 - val_loss: 0.0932 - val_accuracy: 0.9835 - val_ms
e: 0.0253
Epoch 8/15
14/14 [==============================] - 20s 1s/step - loss: 0.1601 - accuracy: 0.9312 - mse: 0.0495 - val_loss: 0.0836 - val_accuracy: 0.9504 - val_ms
e: 0.0256
Epoch 9/15
14/14 [==============================] - 21s 1s/step - loss: 0.1584 - accuracy: 0.9220 - mse: 0.0497 - val_loss: 0.0558 - val_accuracy: 0.9835 - val_ms
e: 0.0145
Epoch 10/15
14/14 [==============================] - 20s 1s/step - loss: 0.1782 - accuracy: 0.9243 - mse: 0.0553 - val_loss: 0.0678 - val_accuracy: 0.9835 - val_ms
e: 0.0183
Epoch 11/15
```

Predictions

```
In [ ]:
from keras.models import load_model
import cv2
import numpy as np
from PIL import Image
from keras.utils import img_to_array
model = load_model('/content/drive/MyDrive/IBM PROJECT/ffd_model.h5')
def prediction(img_path):
    i = cv2.imread(img_path)
    i = cv2.cvtColor(i, cv2.COLOR_BGR2RGB)
    img = Image.open(img_path)
    img = img.resize((128,128))
    x = img_to_array(img)
    x = np.expand_dims(x,axis=0)
    pred = model.predict(x)
    plt.imshow(i)
    print("%s"%("FOREST FIRE DETECTED! SMS SENT!" if pred==[[1.]] else "NO FOREST FIRE DETECTED"))
```

```
In [3]:
#import opencv librariy
import cv2
#import numpy
import numpy as np
#import image function from keras
from keras.preprocessing import image
#import load_model from keras
from keras.models import load_model
#import client from twilio API
from twilio.rest import Client
#imort playsound package
from playsound import playsound
```

```
WARNING:playsound:playsound is relying on another python subprocess. Please use `pip install pygobject` if you want playsound to run more efficiently.
```

```
In [5]:
model = load_model(r'/content/drive/MyDrive/IBM PROJECT/ffd_model.h5')
#define video
video = cv2.VideoCapture(0)
#define the features
name = ['forest','with forest']
```

```
In [6]:
account_sid = 'AC381739ada733d1ba2fcee2548f10eef0'
auth_token = '928561042fab0f4b80ca038a4d7447f2'
client = Client(account_sid, auth_token)

message = client.messages \
    .create(
         body='FOREST FIRE IS DETECTED, STAY ALERT',
         from_='+14246228559',
          to = '+918919689576'
     )

print(message.sid)
```

```
SMb82de2eb481d77eadb829c25c1b47246
```

DEMO LINK:

https://drive.google.com/file/d/11NKVsW0-kyzG45V0oi1jyQ3d54clUIXN/view?usp=sharing

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-5377-1658761176