Project Final Deliverable

Report Document

Date	19 November 2022
Team ID	PNT2022TMID48059
Project Name	Project – EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

Team ID: PNT2022TMID48059

Batch: B2-2M4E

TEAM LEADER:

Name: P. DHINESH KUMAR

Register Number: 912419104007

TEAM MEMBERS:

Name: R. SELVENDRAN

Register Number: 912419104028

Name: P. ARUN

Register Number: 912419104002

Name: R. KARTHIK

Register Number: 912419104013

EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

PROJECT REPORT

Submitted by,

P.DHINESH KUMAR	912419104007
R.SELVENDRAN	912419104028
P.ARUN	912419104002
R.KARTHIK	912419104013

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SHANMUGANATHAN ENGINEERING COLLEGE
ARASAMPATTI – 622507

ANNA UNIVERSITY:: CHENNAI 600 025

NOVEMBER 2022

ABSTRACT

The forest is one of the most important wealth ofevery country. The forest fires destroys the wildlife habitat,damages the environment, affects the climate, spoils thebiological 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.

In the proposed method, a color spatial segmentation, temporal segmentation,global motion compensation, Support Vector Machine (SVM)classifications are used to detect the fire and to segment the fire from the video sequence. The method is implemented over the two real time data sets. The proposed method is most suitable for segmenting fire events over unconstrained videos in real time.

Wild fires are actual risk to individual life, eco-friendly system as well as in infrastructure. There are many economical fire recognition sensor tools available, although all are different in their response time delay, require higher maintenance, high expenditure and additional problems to be applied in large open areas such as forests.

we propose forest fire detection along with the following phases. Initially, convert the video into frames. Secondly, these segmented moving regions (RGB) were converted into YCbCr color space and then seven fire finding rules were applied to separate candidate fire pixels. At last, temporal variation (motion detection) is used to distinguish among fire and fire like color objects. Nine internet video data sets are used in proposed method. Final result shows that up to 99 percent of true detection rates are achieved by the proposed method.

TABLE OF CONTENT

LIST OF FIGURES

1.INTRODUCTION

1.1 Project Overview

One of the natural resources that must be protected and explored wisely and securely, as well as sustainability concepts, is the tropical forest. However, poor exploitation has resulted in certain issues. Forest fires are one type of calamity. This calamity not only destroys forest natural resources, but also causes additional problems such as fogging and ecosystem disruption. On the other hand, research in intelligence systems, particularly machine learning aided by image sensing algorithms and neurofuzzy, results in a design system capable of monitoring real data, proceeding with a rule base derived from human reasoning, and producing prediction systems and supporting information for decisionmaking. Forest fires (also known as wildfires) are a major environmental hazard that threaten forest preservation, cause economic and ecological damage, and cause human suffering.

The main objective of the work is to apply image sensing and machine learning in the aim of forest fire prediction with the input stream of images. It uses image processing, background subtraction and special wavelet analysis. We also use SVM for classifying the candidate region to either real fire or non- fire. We use the faster R-CNN object detection model, which has full image convolutional features. The algorithm uses YCbCr color space after processing the RGB color space.

The fast improvement of computerized camera innovation has brought about a quick expansion in picture quality and diminished expense of the cameras. Further more, the way that computerized cameras can cover bigger regions with astounding outcomes, assists us with picking them over sensors. And sensors have the ability to get themselves perished in the disaster situation easily when compared to the latest digital cameras.

A web-based application system to help users to predict whether a forest fire will occur or not from the input stream of images which is user- friendly, cost-effective and mobile. Comparison between existing and proposed model is carried out. Faster R-CNN is the object detection model which is used. The alternate models are Single Shot Multibox Detector (SSMD) and FCN. The predictions will be given after image processing and can be used for automatic forest-fire alarm systems.

The technique to fire detection hasn't altered in the last 40 years – a system will see smoke, smell smoke, or detect heat. While this is undeniably a sound strategy, there is always the possibility of a game-changing breakthrough that will turn the industry on its head. Fire detectors are becoming increasingly reliable, but they are also being modified to reduce the risk of false alerts. False alarms are currently one of the most serious problems in fire protection. Not only is this clearly a waste of resources that diverts assistance away from potentially life-threatening situations, but it also has an impact on the fire strategies of different fire departments across the country.

1.2 Purpose

The purpose of the project is to solve the existing problem of unreliable fire detection systems used to detect the forest fire. The project is aimed at using surveillance cameras in order to detect and monitor the occurrence of fire. Since the cameras are already installed in places, this system is aimed at diminishing the disadvantages of false alarm, making the system cost effective and fast method of detecting fire. The system uses Open Source Computer Vision, also known as OpenCV, is an open source freeware which is aimed at computer vision.

Then the Unmanned aerial vehicles (UAVs) are promising options to patrol the forest by making them fly over the region. The proposed model deployed on an onboard UAV has achieved 1.24 seconds of classification time with an accuracy of 91% and an F1 score of 0.91. The onboard CPU is able to make a 3D model of the forest fire region and can transmit the data in real time to the ground station. The proposed model is trained on both dense and rainforests in detecting and predicting the chances of fire. The proposed model out performs the traditional methods such as Bayesian classifiers, random forest, and support vector machines.

Over the last few decades, forest fires are increased due to deforestation and global warming. Many trees and animals in the forest are affected by forest fires. Technology can be efficiently utilized to solve this problem. Forest fire detection is inevitable for forest fire management. The purpose of this work is to propose deep learning techniques to predict forest fires, which would be cost-effective a new approach for fire detection and control, in which modern technologies are used. This allows them to use computer vision methods for recognition and detection of smoke or fire, based on the capturing images or the video input from the camera sensors and drone cameras. These solutions mainly aim to mitigate the damage caused by the fires, using methods for their early detection.

2.LITERATURE SURVEY

A. Emerging methods for early detection of forest fires using unmanned aerial vehicles and LORAWAN sensor networks:

Forest fires are occurring throughout the year with an increasing intensity in thsummer and autumn periods. These events are mainly caused by the actions of humans, but different nature and environmental phenomena, like lightning strikes or spontaneous combustion of dried leaves or sawdust, can also be credited for their occurrence. Regardless of the reasons for the ignition of the forest fires, they usually cause devastating damage to both nature and humans. Forest fires are also considered as a main contributor to the air pollution, due to the fact that during every fire huge amounts of gases and particle mater are released in the atmosphere. In this paper we will discuss and present two different emerging solutions for early detection of forestfires.

B. A Review on Early Forest Fire DetectionSystems Using Optical Remote Sensing:

The environmental challenges the world faces nowadays have never been greater or more complex. Global areas covered by forests and urban woodlands are threatened by natural disasters that have increased dramatically during the last decades, in terms of both frequency and magnitude. Large-scale forest fires are one of the most harmful natural hazards affecting climate change and life around the world. Thus, to minimize their impacts on people and nature, the adoption of well-planned and closely coordinated effective prevention, early warning, and response approaches are necessary. This paper presents an overview of the optical remote sensing technologies used in early fire warning systems and provides an extensive survey on both flame and smoke detection algorithms employed by each technology. Three types of systems are identified, namely terrestrial, airborne, and spaceborne-based systems, while various models aiming to detect fire occurrences with high accuracy in challenging environments are studied. Finally, the strengths and weaknesses of fire detection systems based on optical remote sensing are discussed aiming to contribute to future research projects for the development of early warning firesystems.

C. Forest Fire DetectionSystem:

The world is burning. As global warming continues to display a statistical rise in global average temperatures and various environmental factors continue to contribute to the rise in forest fires, the need for a wireless detection system to recognize these fire hazards and that can successfully alert the necessary first responders is becoming more and more apparent. Such a detection and alert system would be able to potentially save billions of dollars in property, infrastructure, and environmental costs and damages, preserve wildlife habitats and ecosystems that are directly affected by forest fires, and prevent the displacement of countless families from their homes that neighbor forested areas and regions. Therefore, we have come together as an engineering team to propose and develop a prototype solution to these issues using our acquired technical knowledge as senior electrical engineering students for our senior design project this semester. Our project idea entitled, "Forrest Fire Detection System," will be comprised of multiple systems working in tandem: a LoRa antennae system that will wirelessly transmit sensor data to an accessible website, a solar PV power supply, and a data retrieval gateway and alert system.In summary, we aim to reduce the social, economical, and environmental impacts brought on by forest fires.

2.1 EXISTING PROBLEM

The existing system for detecting fire are smoke alarms and heat alarms. The main disadvantage of the smoke sensor alarm and heat sensor alarms are that just one module is not enough to monitor all the potential fire prone places.The only way to prevent a fire is to be cautious all the time. Even if they are installed in every nook and corner, it just is not sufficient for an efficient output consistently.

As the number of smoke sensor requirement increase the cost will also increase to its multiple. The proposed system can produce consistent and highly accurate alerts within seconds of accident of the fire. It reduces cost because only one software is enough to power the entire network of surveillance. Research is active on this field by data scientists and machine learning researchers. The real challenge is to minimize the error in detection of fire and sending alerts at the right.

2.2 REFERENCE

[1] Saponara, S., Elhanashi, A. & Gagliardi, A. Real-time video fire/smoke detection based on CNN in antifire surveillance systems.J Real-Time Image Proc 18, 889–900 (2021).

[2] Qingjie Zhang, Jiaolong Xu, Liang Xu and Haifeng Guo, Deep Convolutional Neural Networks for Forest Fire Detection. IFMEITA 2016.

[3] Pragati, S. S. (2019-2020). International Journal Of Advance Scientific Research. Forest Fire Detection Using Machine Learning, 1,2.

[4] A Arul, R. S. (2021, May). Fire Detection System Using Machine Learning. Retrieved from ResearchGate: https://www.researchgate.net/publication/351926970_Fire_Detection_System_Using_Machine_Learning.

[5] Yuanbin Wang, L. D. (2019). Journal of algorithms and Computational technology. Forest fire image recognition based on convolutional neural network,

[6] Kim, Byoungjun, and Joonwhoan Lee. 2019. "A Video-Based Fire Detection Using Deep Learning Models" Applied Sciences 9, no. 14: 2862

[7] Xu, R.; Lin, H.; Lu, K.; Cao, L.; Liu, Y. A Forest Fire Detection System Based on Ensemble Learning. Forests 2021, 12, 217. https://doi.org/10.3390/f1202021

[8] BoWFire Dataset. Available online: https://bitbucket.org/gbdi/bowfire-dataset/downloads/ (accessed on 1 January 2021). 29.

[9] FD-Dataset. Available online: http://www.nnmtl.cn/EFDNet/ (accessed on 1 January 2021).

[10]Forestry Images Available online: https://www.forestryimages.org/browse/subthumb.cfm?sub=740 (accessed on 1 January 2021). 31. [10].

2.3 PROBLEM STATEMENT DEFINITION

A problem statement is an explanation in research that describes the issue that is in need of study. What problem is the research attempting to address? Having a Problem Statement allows the reader to quickly understand the purpose and intent of the Research.

Customer Problem Statement:

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love. A well-articulated customer problem statement allows you and your team to find the ide al solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

Example:

2.3.1 Problem Statement

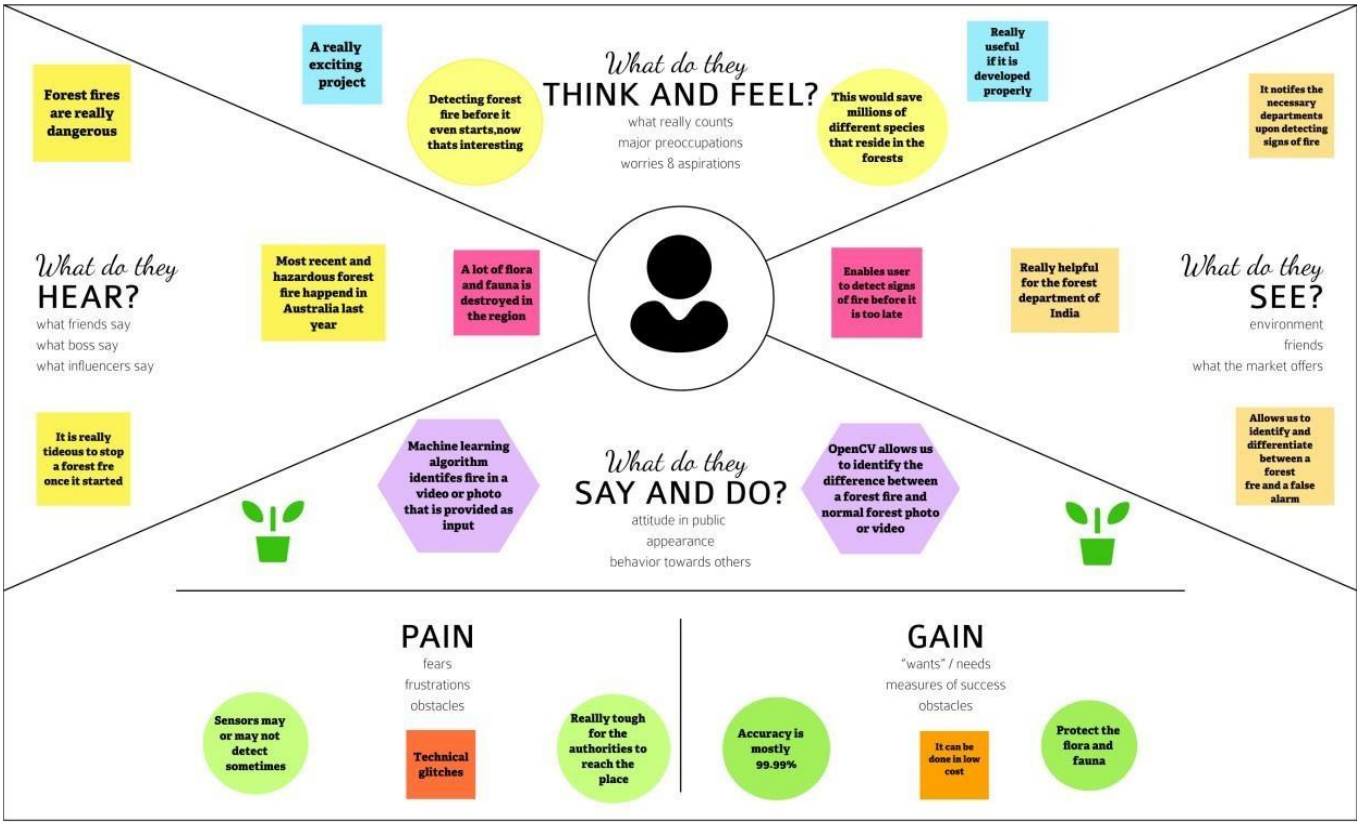
Problem Statement (PS)	I am	I'm trying to	But	Because	Which makes me feel
Problem Statement-1	Aerologist	predict and analyze the composition of Earth's atmosphere	It's a challenge to predict & analyze the damage of atmosphere by forest fire	Lack of information about the forest fire	frustrated
Problem Statement-2	Forest ecologist	Predict & analyze the forest health	It's too hard to predict the absolute forest fire	We can't determine the parameter covered by forest fire	disappointed
Problem Statement-3	Forest Ranger	Protect the forest from forest fires	It's too hard to predict the resources required to extinguish the forest fires	Without having a knowledge about amount of fire covered in forest	confused

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

An empathy map canvas helps brands provide a better experience for users by helping teams understand the perspectives and mindset of their customers. Using a template to create an empathy map canvas reduces the preparation time and standardizes the process so you create empathy map canvases of similar quality.

It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



3.1.1 Empathy Map Canvas

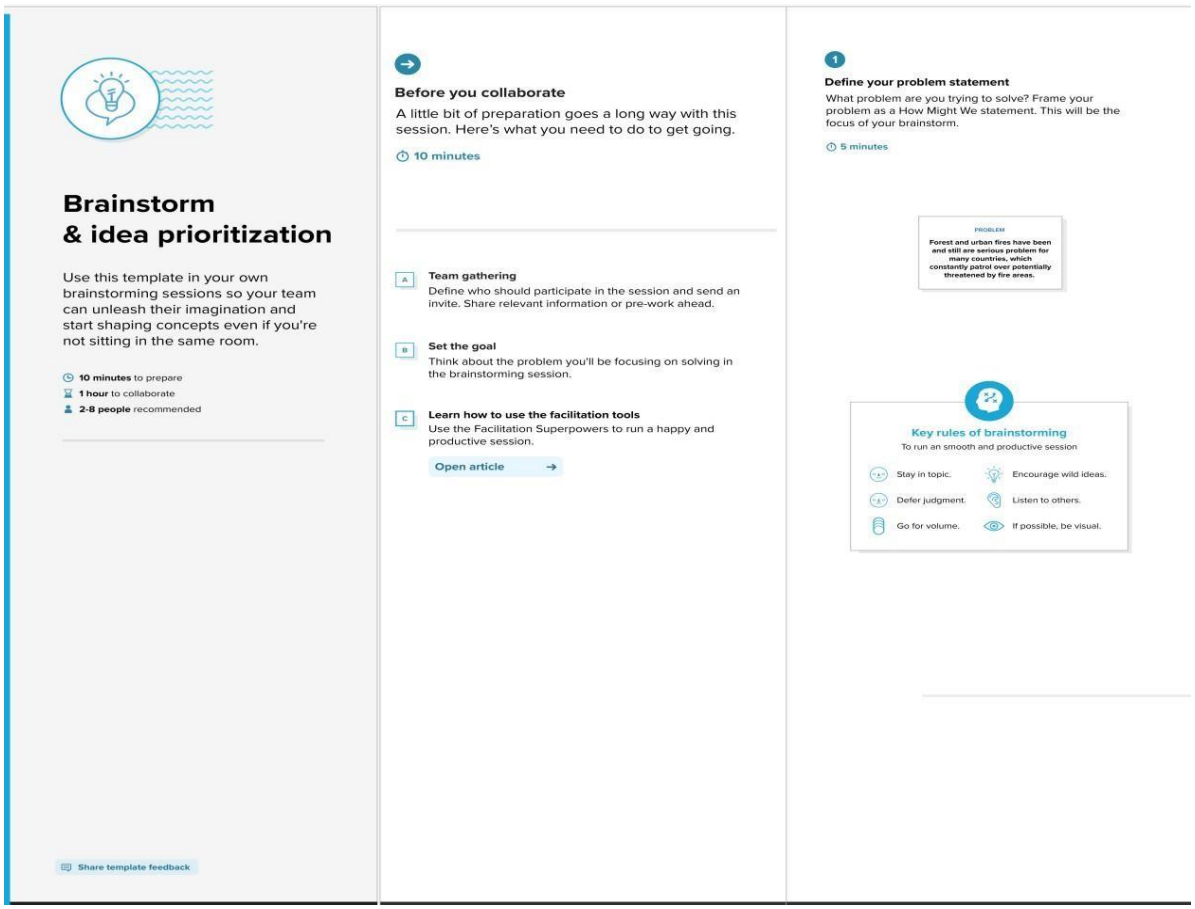
3.2 IDEATION & BRAINSTORMING

Brainstorm & Idea Prioritization Template :

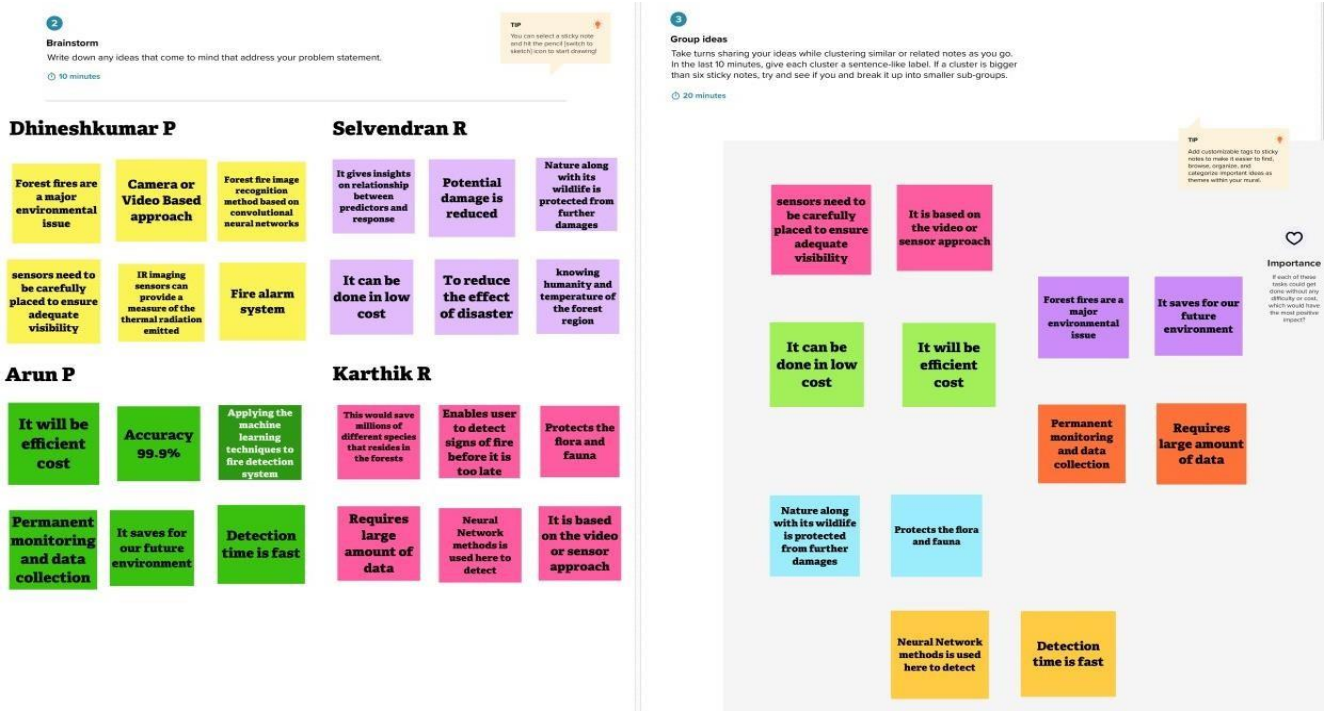
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

People are often stuck in their ways of thinking because of the patterns they see, which is extremely bad for innovation.Creative thinking helps to challenge our assumptions, discover new things, see from new perspectives and keep us mentally sharp.We Use this template in our own brainstorming sessions so your team can unleash theirimagination and start shaping concepts.

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping

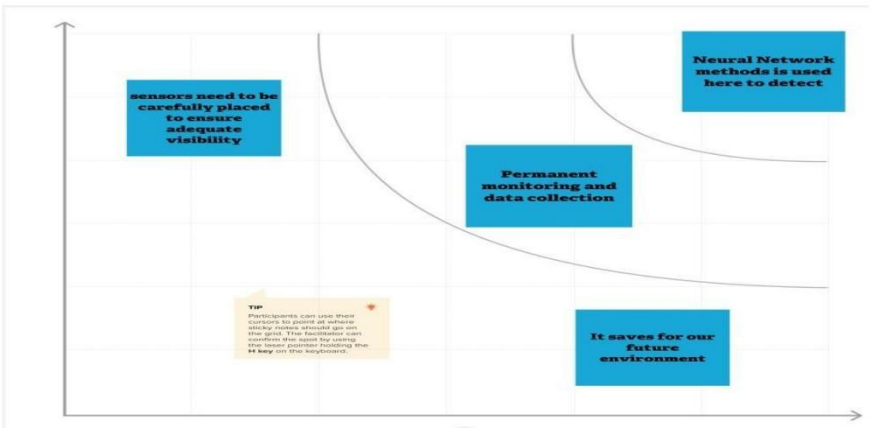


3.2.1 Brainstroming Ideas

Similar Points:

- Forest fires are a major environmental issue.
- Potential damage is reduced.
- It saves for our future environment.
- Requires large amount of data.
- Neural network method is used here to detect.

Step-3: Idea Prioritization



3.2.2 Top 3 Ideas

Top Ideas :

- Sensors need to be carefullyplaced to ensure adequate visibility.
- Permanent monitoring and data collection.
- It saves for our future environment.
- Neural network method is used here to detect.

3.3 PROPOSED SOLUTION

In this proposed system instead of analyzing characteristics parameters of fire i.e color, area, motion, smoke individually, all the parameters are examined simultaneously to reduce the false alarm rates which was present in a previous detection systems. The main part of this system is the flow that will be used to estimate the amount of motion undergone by an object while moving from one frame to another.

The proposed system will give the combine result at the output whether smoke and fire is present or not in the image and video analysing. The system performance can be improved with the use of optimal algorithms for detecting motion and area and extracting features of fire. The enhanced system will performed well than the existing system in terms of detection rate. In this project we have developed a system to detect an occurrence of fire.

The parameters are:

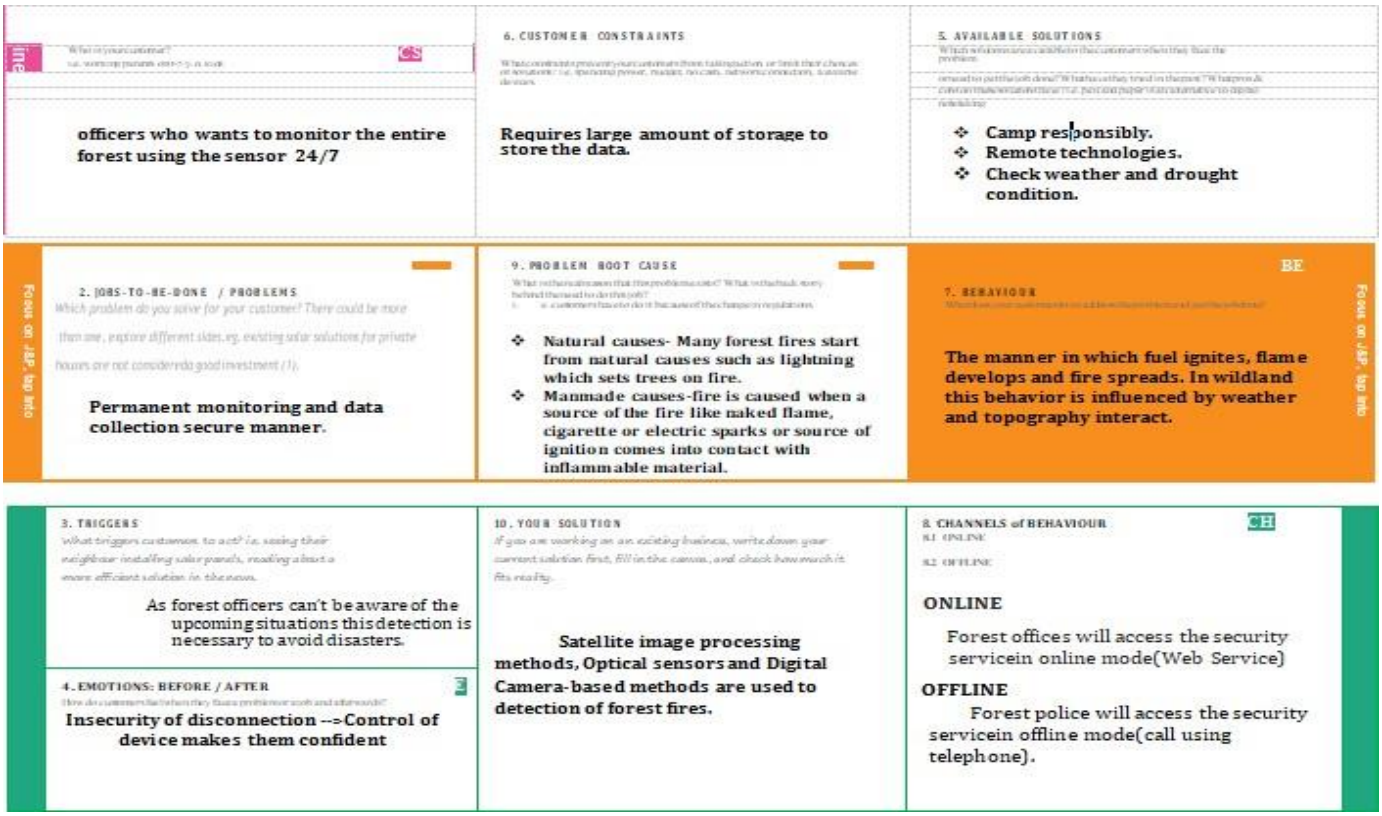
- Problem Statement (Problem to be solved) .
- Idea / Solution description.
- Novelty / Uniqueness.
- Social Impact / Customer Satisfaction.
- Business Model (Revenue Model).
- Scalability of Solution.

S.No	Parameter	Description
1.	Problem Statement (Problem to besolved)	Forest and urban fires are still a serious problem for many countries in the world. The (UAVs), which constantly patrol overpotentially threatened by fire areas. The UAVs also utilize the benefits from Artificial intelligence (AI) and are equipped with on board processing capabilities.
2.	Idea / Solution description	Recent advances in computer vision, machine learning, and remote sensing technologies offer new tools for detecting and monitoring forest fires, while the development of new materials and microelectronics have allowed sensors to be more efficient in identifying active forest fires.
3.	Novelty / Uniqueness	Permanent monitoring, data collection and processing. Terrestrial-based early detection systems consist of either individual sensors (fixed, PTZ, or 360◦ cameras) or networks of ground sensors.
4.	Social Impact / Customer Satisfaction	Growing public alarm at the problem of large-scale forest fires, is evident from an assessment of their past and present repercussions on the population in general.
5.	Business Model (Revenue Model)	Forest sector has strong importance for the economic, social and environmental issues. Portuguese forestry sector is of great importance for the added value creation, for the jobs creation.
6.	Scalability of the Solution	There are several factors that affect the evolution of a wild land fire. It is well known that wind is one of the key parameters to understand the forest fire propagation. Intuitively, the meteorological wind speed tends to drive the main direction of forest fire spread.

3.4 PROBLEM SOLUTION FIT

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem.

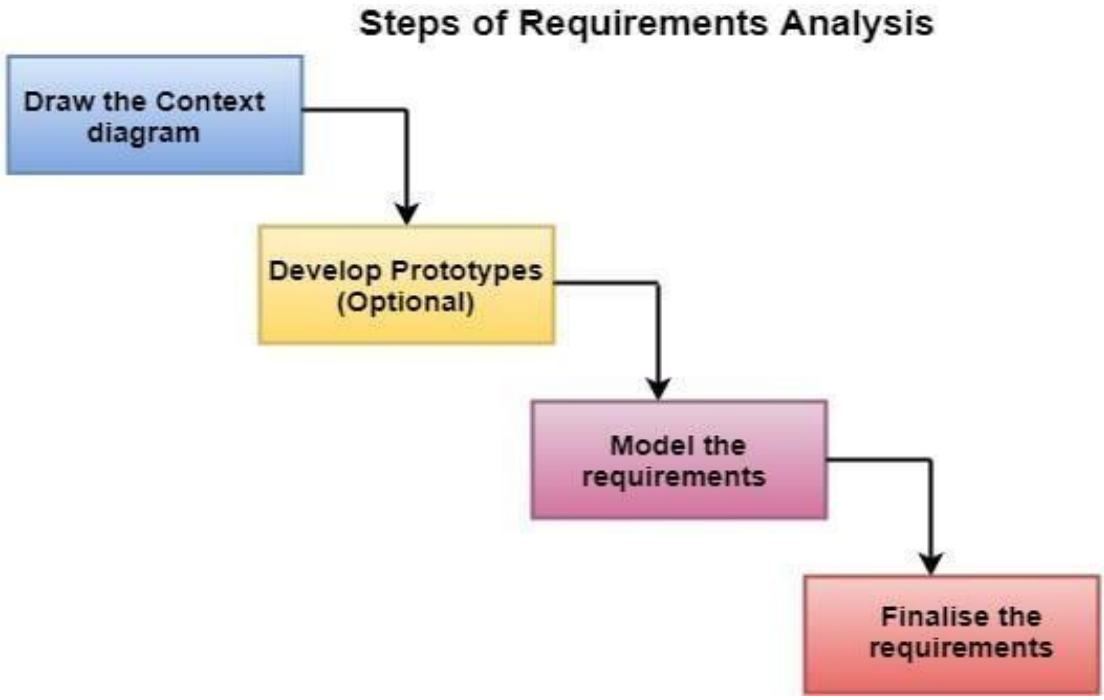
With this template you will be able to take important information into consideration at an earlier stage and look at problem solving in depth. It increases your



3.4.1Chances of finding problem-solution and product-market fit.

4. REQUIREMENT ANALYSIS

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications.



4.1 Steps of Requirements Analysis

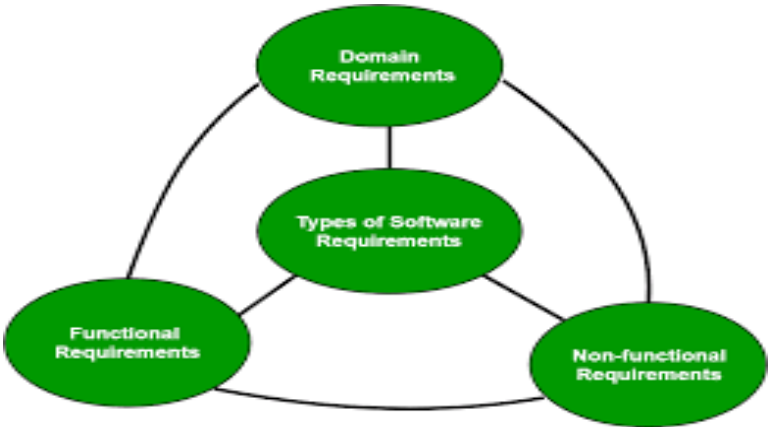
Requirements analysis is a team effort that demands a combination of hardware, software and human factors engineering expertise as well as skills in dealing with people. Here are the main activities involve in requirement analysis:

- Identify customer's needs.
- Evaluate system for feasibility.
- Perform economic and technical analysis.
- Allocate functions to system elements.
- Establish schedule and constraints.
- Create system definitions.

TYPES OF REQUIREMENT ANALYSIS

1.Functional Requirement

2.Non-Functional Requirement



4.2 Types of Requirement Analysis

4.1 FUNCTIONAL REQUIREMENT

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

1. User Registration.
2. User Confirmation.
3. User Login.
4. User Search.
5. User Profile.
6. User Application.

Description of Functional Requirement

1. **User Registration**- Registration through Gmail.
2. **User Confirmation**- Confirmation via Email and Confirmation via OTP.
3. **User Login**- Login using credentials.

4. **User Search**- Search for Info on forest fire occurrence.

5. **User Profile-** User shall be given a live feed of the forest and alert them with notification message.

6. **User Application**- User is alerted if there is an forest fire occurrence in their surroundings.

4.2 NON-FUNCTIONAL REQUIREMENT

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other.

They are also called non-behavioural requirements.

1. Usability.
2. Security.
3. Reliability.
4. Performance.
5. Availability.
6. Scalability.

Description of Non-Functional Requirements

1. **Usability**- Alerts-according to the user location.

2. **Security**- Instant monitoring live feed with alert of the situation.

3. **Reliability-** The prediction of the forest fire is 88%accuratewithefficiency.

4. **Performance**- The live feed and the alert message is an immediate action without a time delay.

5. **Availability**- The application gives alerts message and live feed24/7.

6. **Scalability**- All the data can be stored in the database, If any problem arise we can solve using stores video.

5. PROJECT DESIGN

Project design is an early phase of the project lifecycle where ideas, processes, resources, and deliverables are planned out. A project design comes before a project plan as it's a broad overview whereas a project plan includes more detailed information.

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves thesystem, what changes the information, and where data is stored.

How are data flow diagrams used?

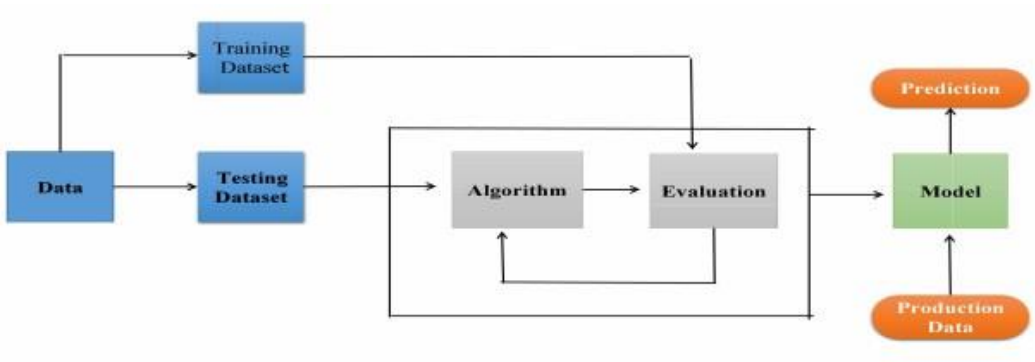
A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled.

They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually "say" things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That's why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented softwareor systems.

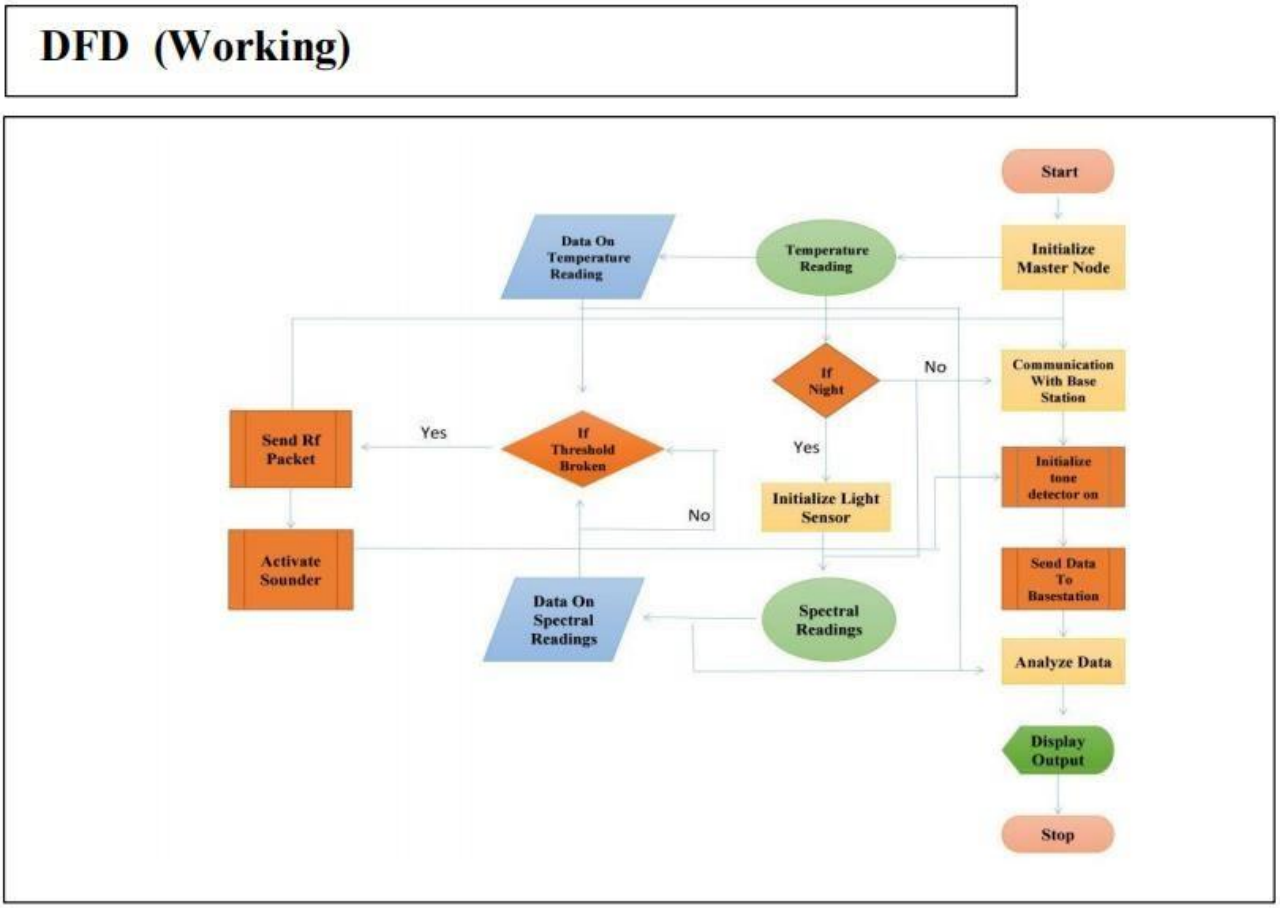
A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

Why is dataflow diagram useful?

Data flow diagrams provide a straightforward, efficient way for organizations to understand, perfect, and implement new processes or systems. They're visual representations of your process or system, so they make it easy to understand and prune.



5.1.1 DATA FLOW DIAGRAM



5.1.2 DATA FLOW DIAGRAM

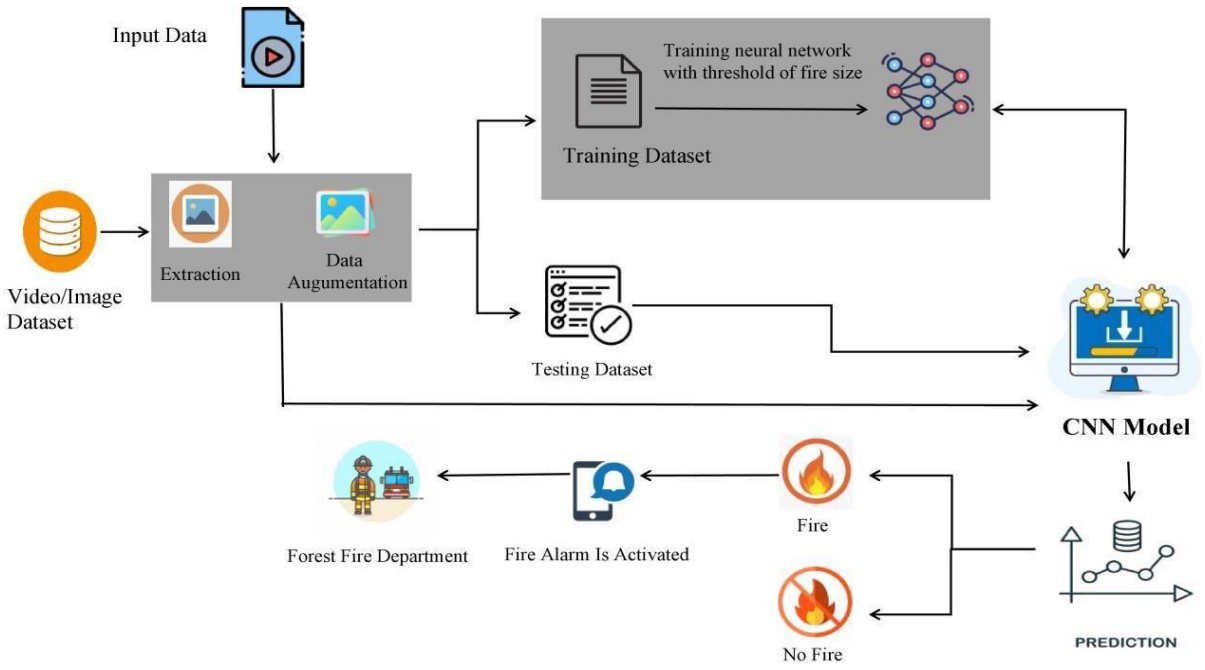
5.2 SOLUTION & TECHNICAL ARCHITECTURE

Solution Architecture

A **solution architecture (SA)** is an architectural description of a specific solution. SAs combine guidance from different enterprise architecture viewpoints (business, information and technical), as well as from the enterprise solution architecture (ESA).

Solution architecture is the process of developing solutions based on predefined processes, guidelines and best practices with the objective that the developed solution fits within the enterprise architecture in terms of information architecture, system portfolios, integration requirements and many more.

It can then be viewed as a combination of roles, processes and documentation that are intended to address specific business needs, requirements or problems through the design and development of applications and information systems.



5.2.1 SOLUTION ARCHITECTURE

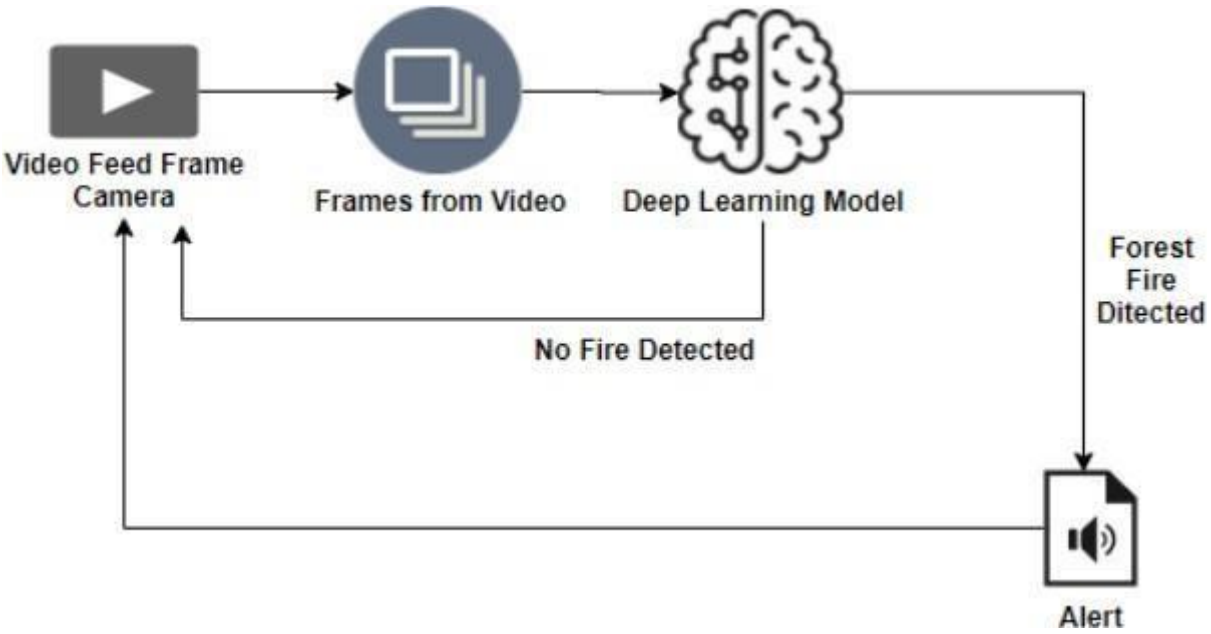
Technical Architecture

Technical Architecture (TA) is **a form of IT architecture that is used to design computer systems**. It involves the development of a technical blueprint with regard to the arrangement, interaction, and interdependence of all elements so that system-relevant requirements are met.

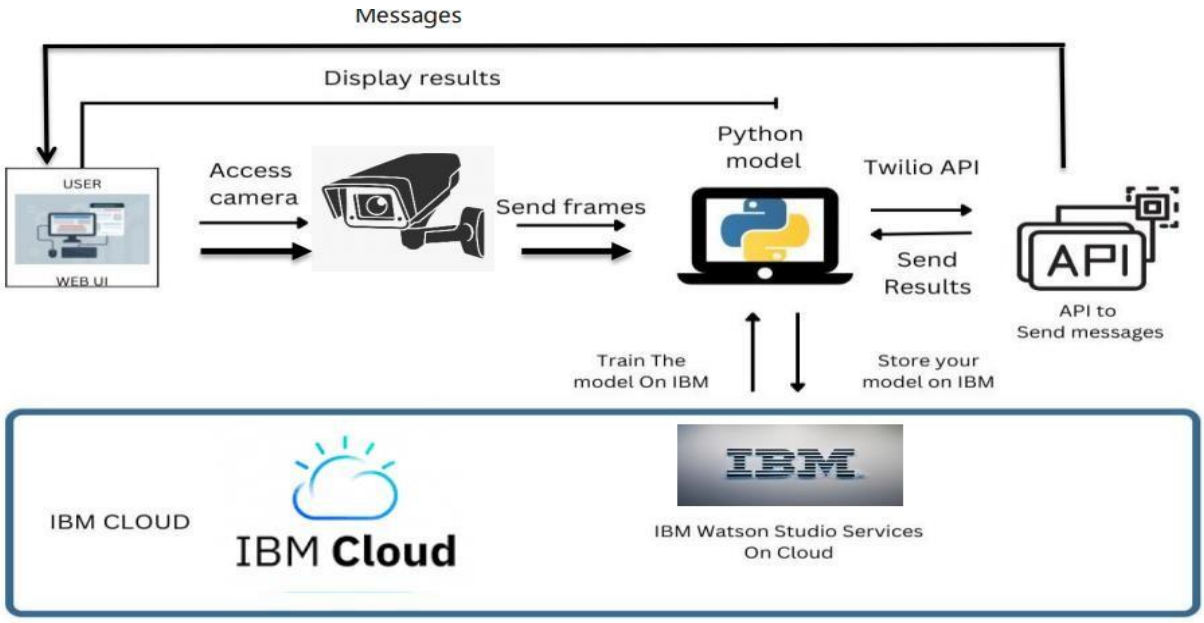
Technical architecture encompasses a variety of different roles and disciplines that are sometimes difficult to tell apart. This is largely due to the highly dynamic nature of IT, and its widespread adoption throughout all industries and businesses that have developed their own practices. In general, there's a differentiation between enterprise architecture, solution architecture, and technology architecture. In order to understand what technology architecture means, it's helpful to examine the term architecture on its own.

The role of a technical architect

1. Depending on the size of the business, technical architects can manage largeteams of developers and IT specialists.
2. They act as technical project managers who analyze design specifications forsingle applications and therein define best practices.
3. This involves tracking its effectiveness by conducting post-execution reviews withsolution architects.



5.1.3 TECHNICAL ARCHITECTURE



5.1.4 TECHNICAL ARCHITECTURE

5.3 USER STORIES

A user story is **an informal, general explanation of a software feature written from the perspective of the end user or customer**. The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer.

Registration

Registration is the process by which a company files required documents with the Securities and Exchange Commission (SEC), detailing the particulars of a proposed public offering. The registration typically has two parts: the prospectus and private filings.

Environmentalist

Collect the data

As a user 1, As an Environmentalist,it is necessary to collect the data of the forest which includes temperature,humidity,wind and rain of the forest .

As a user 2, Identify algorithms that can be used for prediction.

As a user 3, Identify the accuracy of each algorithms.

As a user 4, Evaluate the Dataset.

As a user 5, Identify accuracy,precision,recall of each Algorithms.

As a user 6, Outputs from each algorithm are obtained.

Acceptance criteria

As a user 1, It is necessary to collect the right data else the prediction may become wrong

As a user 2, To collect the algorithm to identify the accuracy level of each algorithm

As a user 3, Accuracy of each algorithm-calculated so that it is easy to obtain the most accurate output

As a user 4, Data is evaluated before processing

As a user 5, These values are important for obtaining the right output

As a user 6, It is highly used to predict the effect and to take precautionary measures.

User Type	Functional Requirement(Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Environmentalist	Collect the data	USN-1	As an Environmentalist,it is necessary to collect the data of the forest which includes temperature,humidity,wind and rain of the forest	It is necessary to collectthe right data else the prediction may becomewrong	High	Sprint-1
		USN-2	Identify algorithms that can be used forprediction	To collect the algorithm to identify the accuracy level of each algorithm	Medium	Sprint-2
		USN-3	Identify the accuracy of each algorithms	Accuracy of each algorithm-calculated so that it is easy to obtain the most accurate output	High	Sprint-2
		USN-4	Evaluate the Dataset	Data is evaluated beforeprocessing	Medium	Sprint-1
		USN-5	Identify accuracy,precision,recall of eachalgorithms	These values are important for obtaining theright output	High	Sprint-3
		USN-6	Outputs from each algorithm are obtained	It is highly used to predictthe effect and to take precautionary measures.	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint planning is an event in scrum that kicks off the sprint. The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved. Sprint planning is done in collaboration with the whole scrum team. Sprint have splinted into the two or three tasks and each sprint have twenty story points. Sprint tasks are assigned into the team members.

Download and collecting the dataset are included in sprint-1. From the collected data are categorized on the basis of parameters set to identify. To train the model, CNN is used to test repeatedly by storing the datasets in server are included in sprint-2. Importing the Model Building libraries to initializing the model and Adding convolution layer, Pooling layer, Flattening layer and also Dense layers such as Hidden layer and Output layer. Configuring the Learning process is accomplished with a call to the compile(). Save the model and Predictions are included in sprint-3. Using OpenCV library function for Video Processing and Creating an account in Twilio service to sending the alert message. Register for IBM Cloud, Creating Watson Studio, Creating the Watson Machine Learning service to train and deploying the model and store the model in the Cloud object storage are included in sprint-4.

Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-1	Collecting Dataset	USN-1	To analyze the fire prone areas and to set the surveillance camera to collect and observe the region continuously for early detection.	2	High	Dhineshkumar Selvendran
Sprint-2	Training & Testing of Model	USN-2	The collected data are categorized on the basis of parameters set to identify. To train the model, CNN is used to test repeatedly by storing the datasets in server.	1	High	Karthik Arun

Sprint-3	Reviewing the model	USN-3	The main task is to check that the model is efficient to work in real time. Therefore, smallest of error decoded needed to be corrected to avoidfuture lags.	1	Medium	Dhineshkumar Karthik
Sprint-4	Implementing themodel	USN-4	The model after testing all it's functionalities is been implemented at forest management offices to get quick responses from the model.	2	High	Selvendran Arun
Sprint-4	Connecting it with API	USN-5	The model should connect with API named Twilio, which receives & sends the management with messages.	2	High	Dhineshkumar Karthik Selvendran Arun

6.2 SPRINT DELIVERY SCHEDULE

A sprint schedule is a document that outlines sprint planning from end to end. It's one of the first steps in the agile sprint planning process—and something that requires adequate research, planning, and communication.

The sprint workflow is intended to help team members evaluate their work and communicate with each other throughout the entire process. The workflow is followed for each sprint. The process includes:

- **Backlog** - A list of set tasks that must be completed before the product is released. The backlog is built by the product owner. The product owner gives a backlog of prioritized items to the scrum master and scrum team. The backlog is based on user stories, which focus on features that consider the type of end user, what they want and why.
- **Sprint planning** - The team discusses top priority user stories and decides what can be delivered in the sprint.
- **Sprint backlog** - Agreed upon by the entire team, this list finalizes and defines what the development team will complete during the sprint.
- **Sprint** – The time frame in which the work must be completed – often 30 days.
- **Daily scrum** – Lead by the scrum master, the team comes together for short daily meetings, in which they discuss what they have completed, what they are working on and any issues that are blocking the work.

- **Outcome** - The outcome of a sprint is a hypothetically usable product. The product owner can decide if the product is ready or if additional features are needed.
- **Sprint end** - At the end of a sprint, two meetings are held:
    - **Sprint review** – The team shows their work to the product owner.
    - **Sprint retrospective** – The team discusses what they can do to improve processes. An important goal is continuous improvement.

All sprint in the total story points twenty then each sprint duration in six days. A sprint-1 start date is 24 October 2022 sprint-1 planned end date for 29 October 2022 the sprint-1 actual release date in 29 October 2022. Next sprint-2 start date is 31 October 2022 sprint-2 planned end date for 05 November 2022 the sprint-2 actual release date in 04 November 2022. Next sprint-3 start date is 07 November 2022 sprint-3 planned end date for 12 November 2022 the sprint-3 actual release date in 11 November 2022. Next sprint-2 start date is 14 November 2022 sprint-4 planned end date for 19 November 2022 the sprint-4 actual release date in 18 November 2022. Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

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	04 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	11Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	18 Nov 2022

Velocity:

Imagine we have a 6-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)

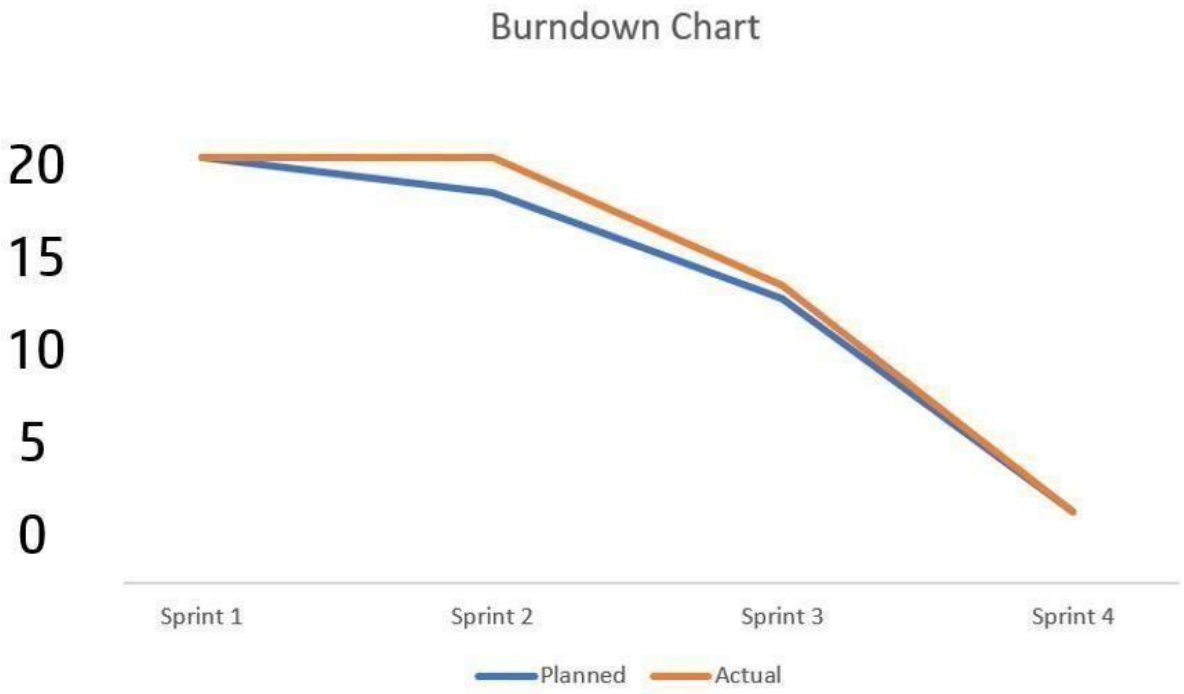
AV = Velocity/Sprint duration

AV = 20/6

AV =3.34

Burndown Chart:

A burn down chart is a graphical representation of work left 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.



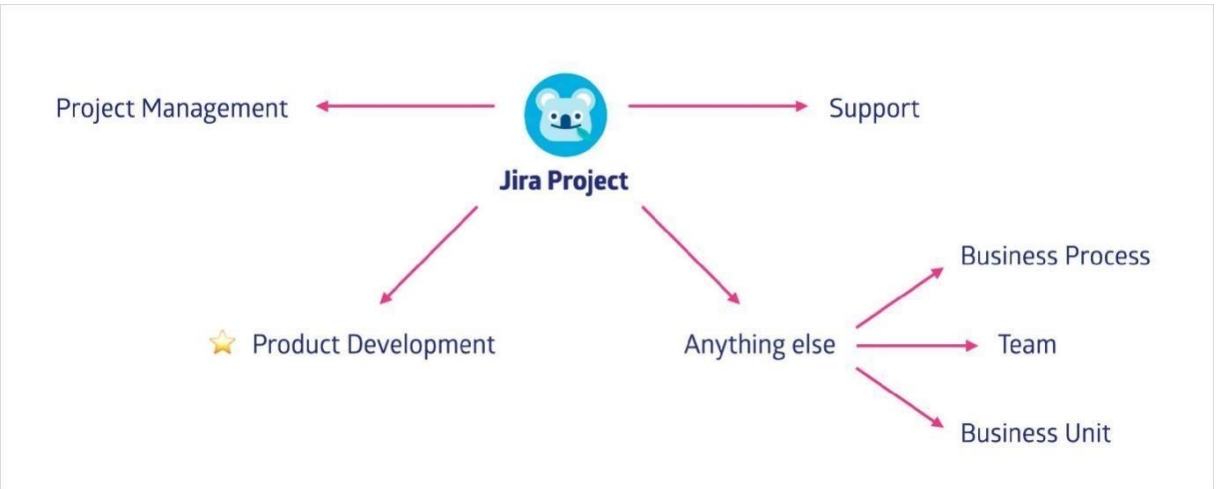
X- Axis: Story Points Y-Axis: Sprints Stages

6.2.1 Burndown Chart

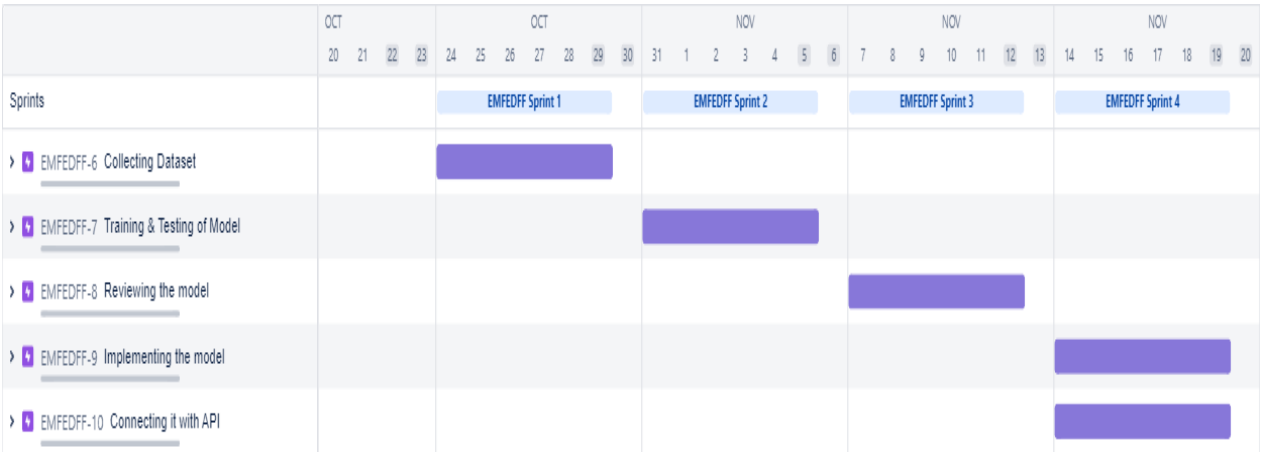
6.3 REPORTS FROM JIRA

Reporting helps you track and analyse your team's work throughout a project. Jira Software has a range of reports that you can use to show information about your project, versions, epics, sprints, and issues.

Create the JIRA account, collaborate the team members and assign the tasks to complete move the in-progress then move the tasks in reviews.

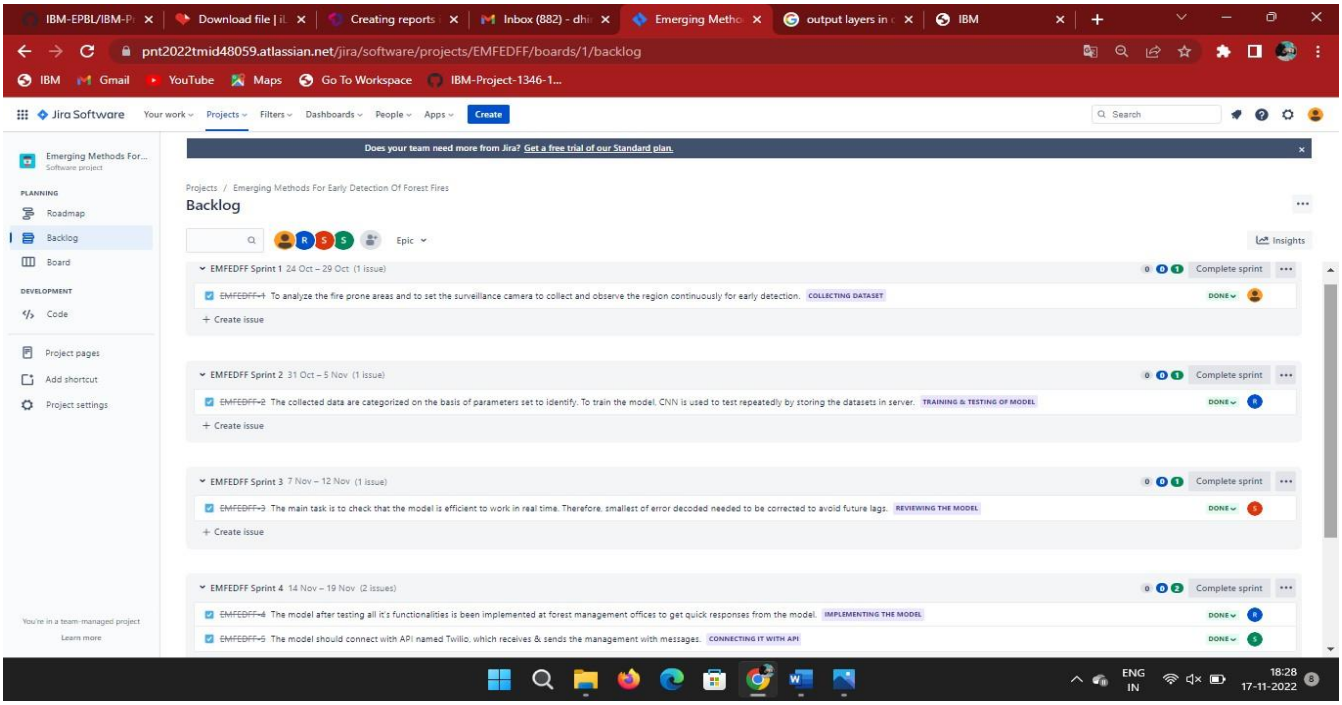


6.3.1 Jira Project Planning

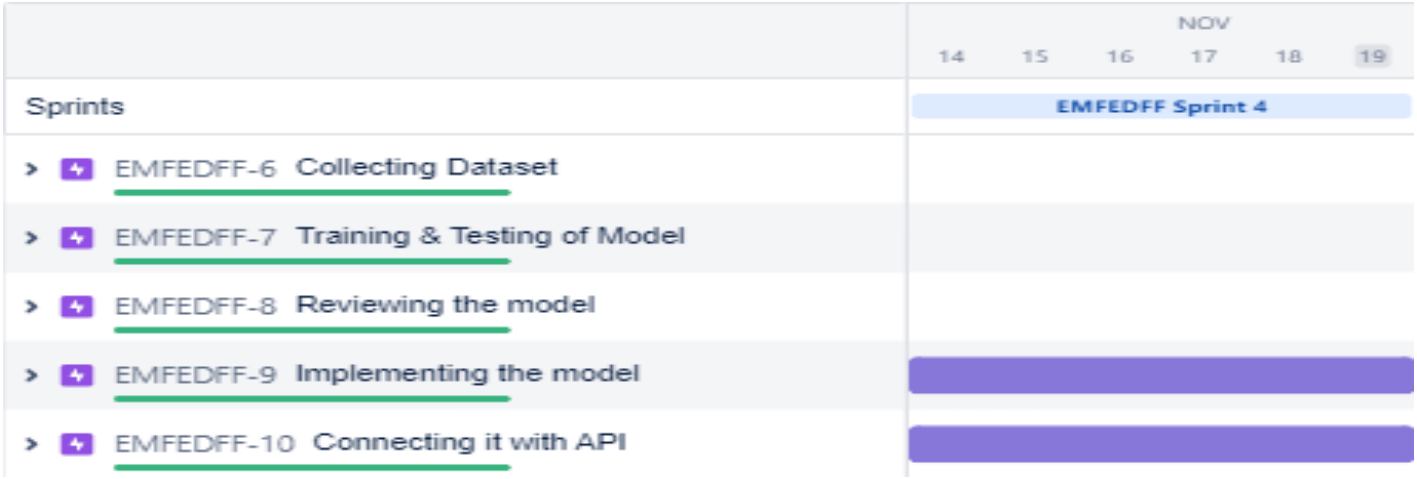


6.3.2 Monthly Task

Sprint delivery chart using Jira software tools



6.3.3 Backlog



6.3.4 Completion of Sprint Delivery

7. CODING & SOLUTIONING

7.1 Feature 1

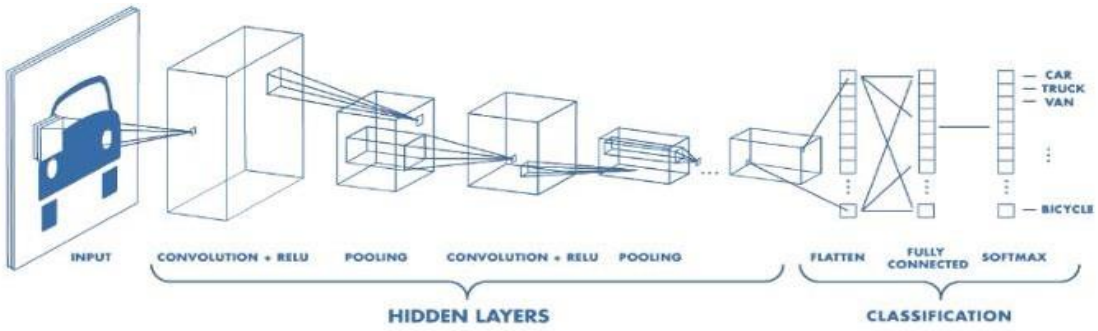
CNN is a type of neural network model which allows us to extract higher representations for the image content. Unlike the classical image recognition where you define the image features yourself, CNN takes the image's raw pixel data, trains the model, then extracts the features automatically for better classification.

Principles of CNN

Convolution Layer

A convolution sweeps the window through images then calculates its input and filter dot product pixel values. This allows convolution to emphasize the relevant features.

This layer performs a dot product between two matrices, where one matrix is the set of learnable parameters otherwise known as a kernel, and the other matrix is the restricted portion of the receptive field. The kernel is spatially smaller than an image but is more in-depth. This means that, if the image is composed of three (RGB) channels, the kernel height and width will be spatially small, but the depth extends up to all three channels.



7.1.1 Convolutional Architecture

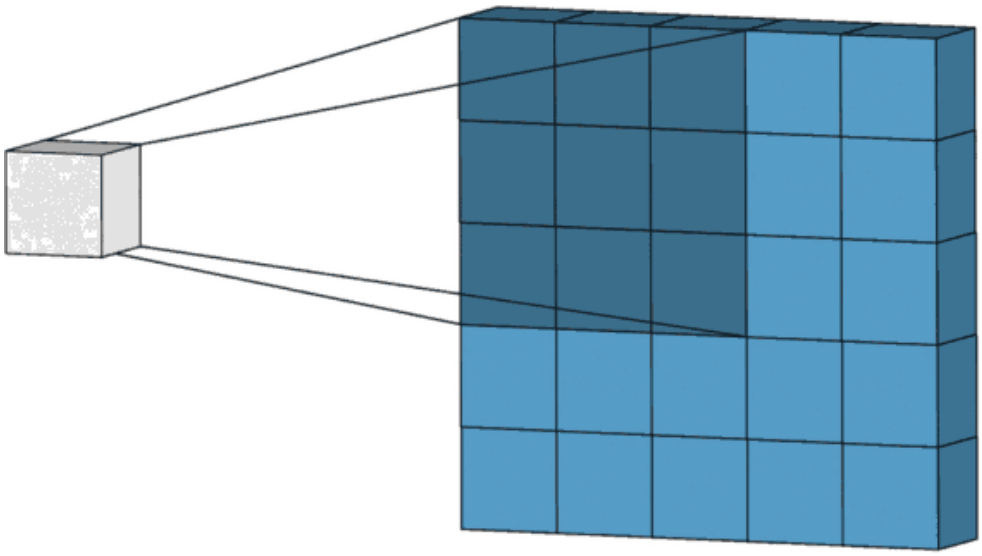
During the forward pass, the kernel slides across the height and width of the image-producing the image representation of that receptive region. This produces a two-dimensional representation of the image known as an activation map that gives the response of the kernel at each spatial position of the image. The sliding size of the kernel is called a stride.

we have an input of size W x W x D and Dout number of kernels with a spatial size of F with stride S and amount of padding P, then the size of output volume can be determined by the following formula:

$$W_{out} = \frac{W - F + 2P}{S} + 1$$

Formula for Convolution Layer

This will yield an output volume of size *Wout* x *Wou*t x *Dout.*

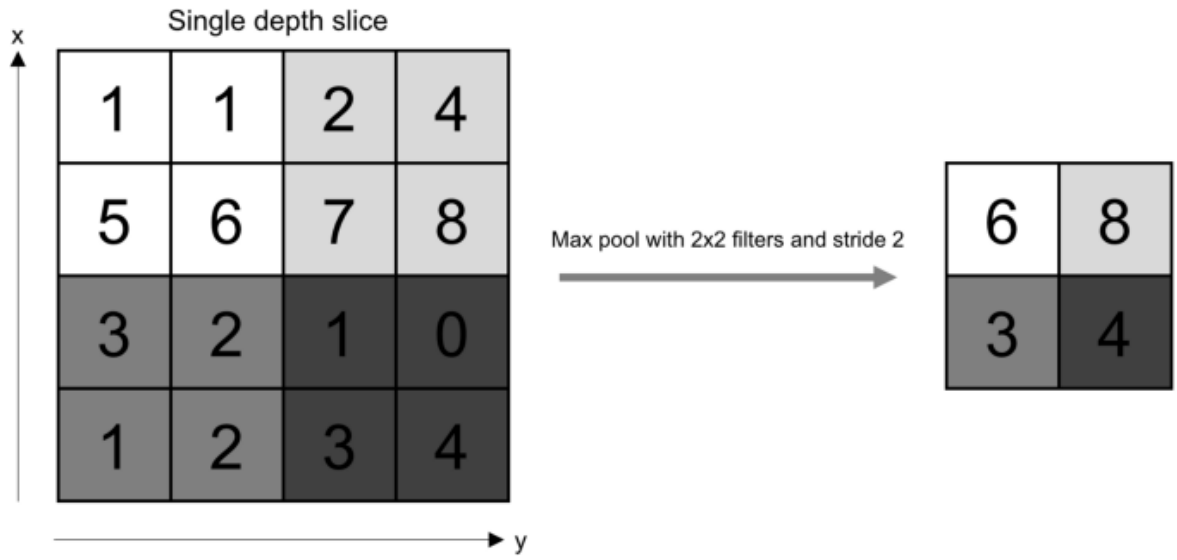


7.1.2 Matrix Multiplication

Pooling Layer

The pooling layer replaces the output of the network at certain locations by deriving a summary statistic of the nearby outputs. This helps in reducing the spatial size of the representation, which decreases the required amount of computation and weights. The pooling operation is processed on every slice of the representation individually.

There are several pooling functions such as the average of the rectangular neighborhood, L2 norm of the rectangular neighborhood, and a weighted average based on the distance from the central pixel. However, the most popular process is max pooling, which reports the maximum output from the neighborhood.



7.1.3 Pooling Operation

If we have an activation map of size *W* x *W* x *D*, a pooling kernel of spatial size *F*, and stride *S*, then the size of output volume can be determined by the following formula:

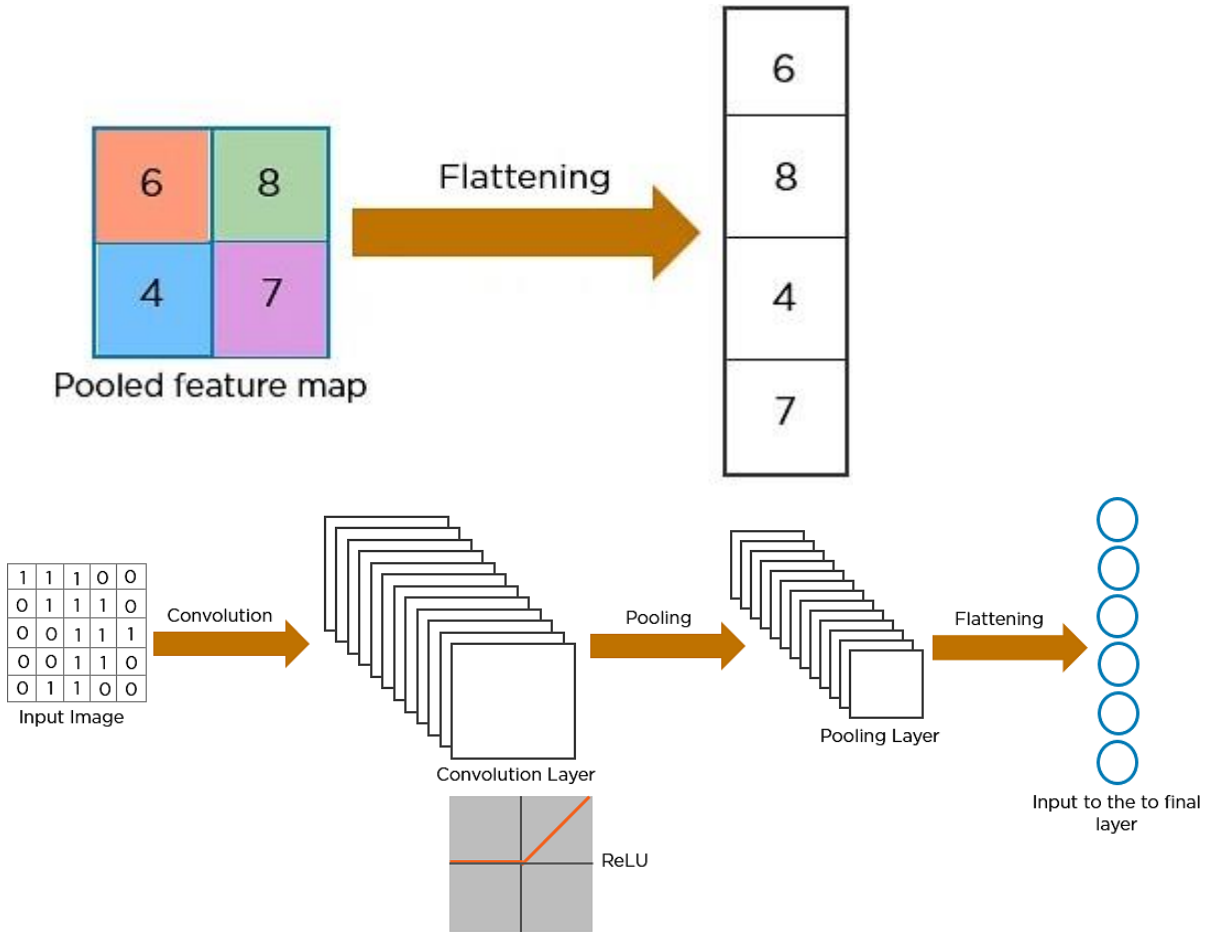
$$W_{out} = \frac{W - F}{S} + 1$$

Formula for Padding Layer

This will yield an output volume of size *Wout* x *Wout* x *D*.

Flattening

After we apply the convolution operation to our image and then we apply pooling to the results of the convolution which is the convolved image.



7.1.4 Flattening

7.2 FEATURE 2

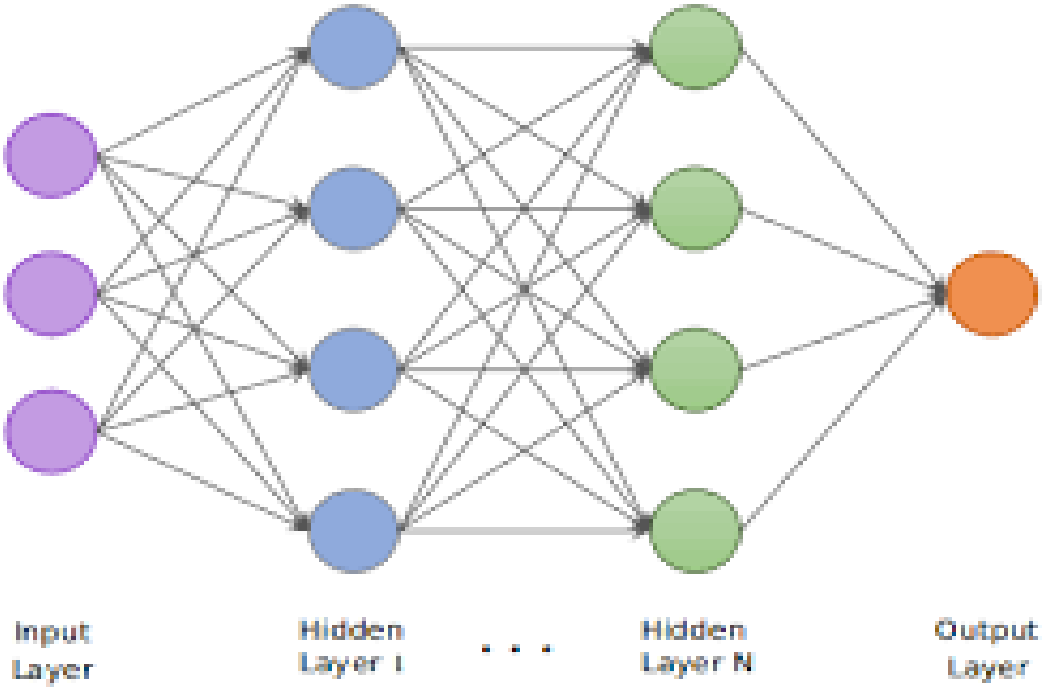
Dense Layers

The name suggests that layers are fully connected (dense) by the neurons in a network layer. Each neuron in a layer receives input from all the neurons present in the previous layer. Dense is used to add the layers.

Hidden Layer

The hidden layers in the network provide a basic building block to transform the data (input layer or the output of the previously hidden layer). Most of the commonly used hidden layers (not all) follow a pattern. It begins with applying a function to its input, moving onto pooling, normalization, and finally applying the activation before it can be fed as input to the next layer.

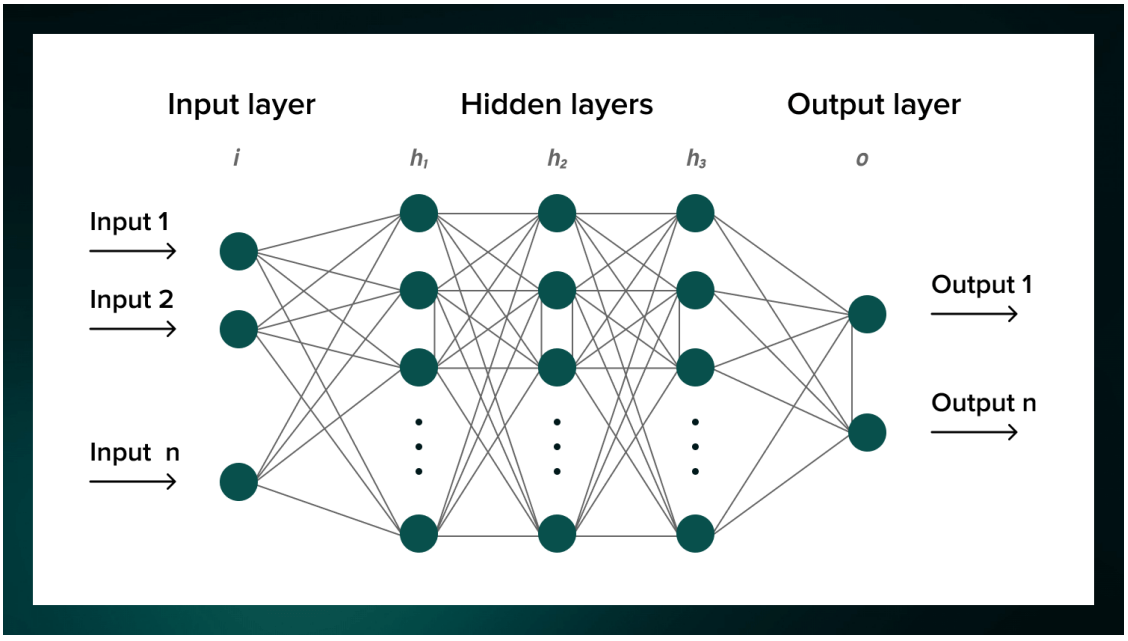
Hidden layers are very common in neural networks, however their use and architecture often varies from case to case. As referenced above, hidden layers can be separated by their functional characteristics. For example, in a CNN used for object recognition, a hidden layer that is used to identify wheels cannot solely identify a car, however when placed in conjunction with additional layers used to identify windows, a large metallic body, and headlights, the neural network can then make predictions and identify possible cars within visual data.



7.2.1 Hidden Layer

Output Layer

After multiple layers of convolution and padding, we would need the output in the form of a class. The convolution and pooling layers would only be able to extract features and reduce the number of parameters from the original images. However, to generate the final output we need to apply a fully connected layer to generate an output equal to the number of classes we need. It becomes tough to reach that number with just the convolution layers. Convolution layers generate 3D activation maps while we just need the output as whether or not an image belongs to a particular class. The output layer has a loss function like categorical cross-entropy, to compute the error in prediction. Once the forward pass is complete the backpropagation begins to update the weight and biases for error and loss reducton.



7.2.2 Output Layer

Video Analysis

OpenCV For Video Processing

OpenCV is an open-source library that provides us with the tools to perform almost any kind of image and video processing.

Capture Video from Camera

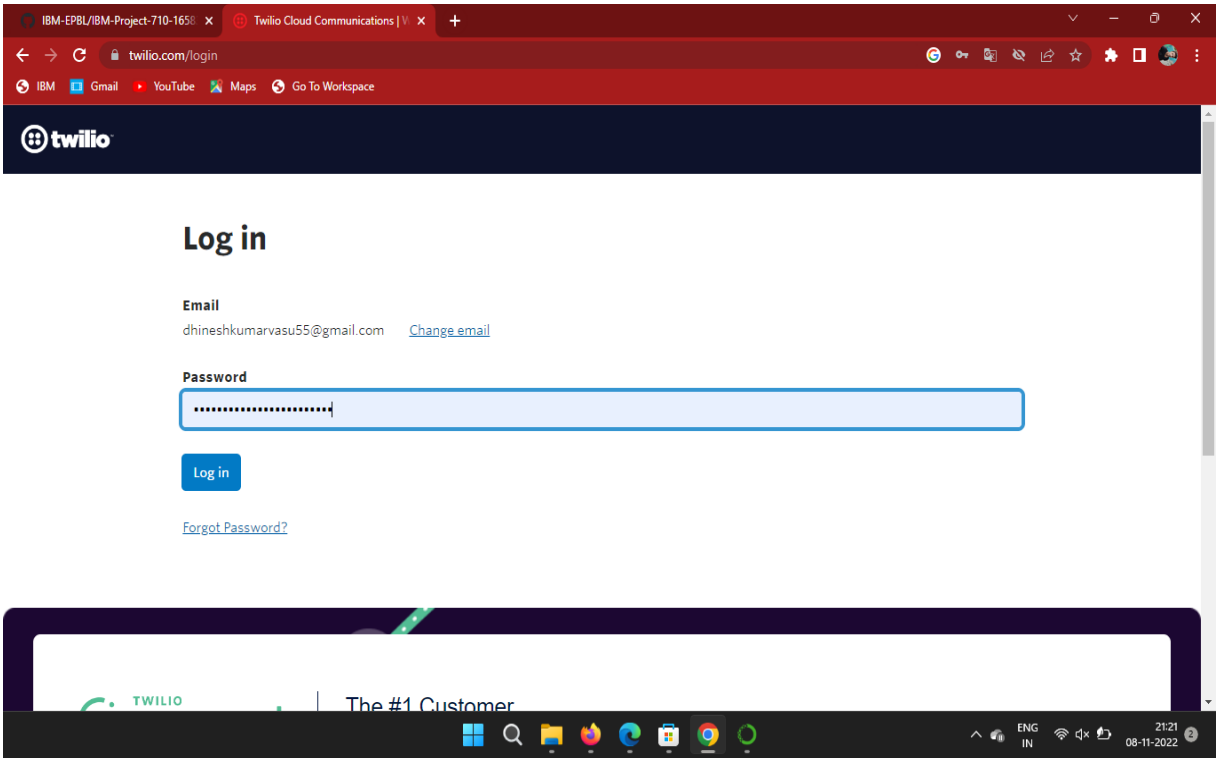
Often, we have to capture the live stream with a camera. OpenCV provides a very simple interface to this. Let's capture a video from the camera (I am using the in-built webcam of my laptop), convert it into grayscale video, and display it.

To capture a video, you need to create a Videocapture object. Its argument can be either the device index or the name of a video file. The device index is just the number to specify which camera. Normally one camera will be connected (as in my case). So I simply pass 0 (or -1). You can select the second camera by passing 1 and so on. After that, you can capture frame-by-frame. But in the end, don't forget to release the capture. To read web cam will see the code.

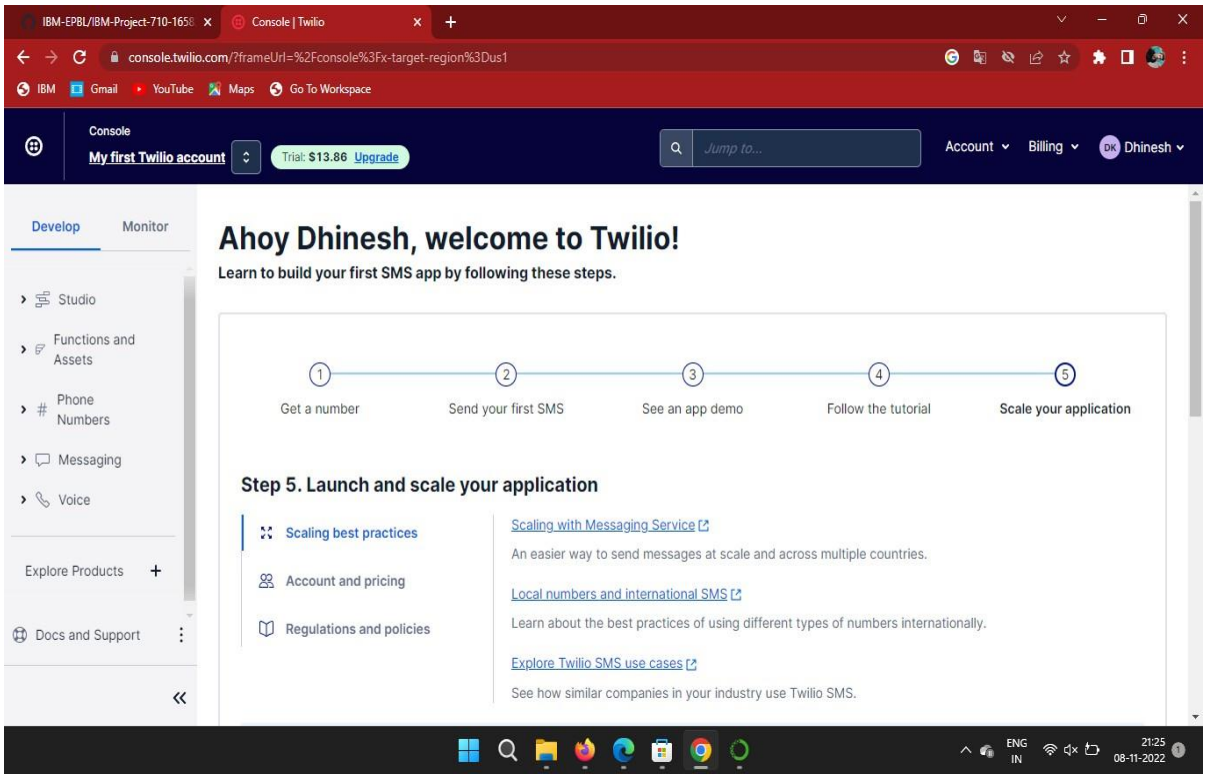
Creating An Account In Twilio Service

Creating an account in twilio service for sending alert message when fire is detected in forest. After created the account login into home page or dashboard. There can show option like get a number and Also there will be a account_sid and auth_token code copy it apply on coding.

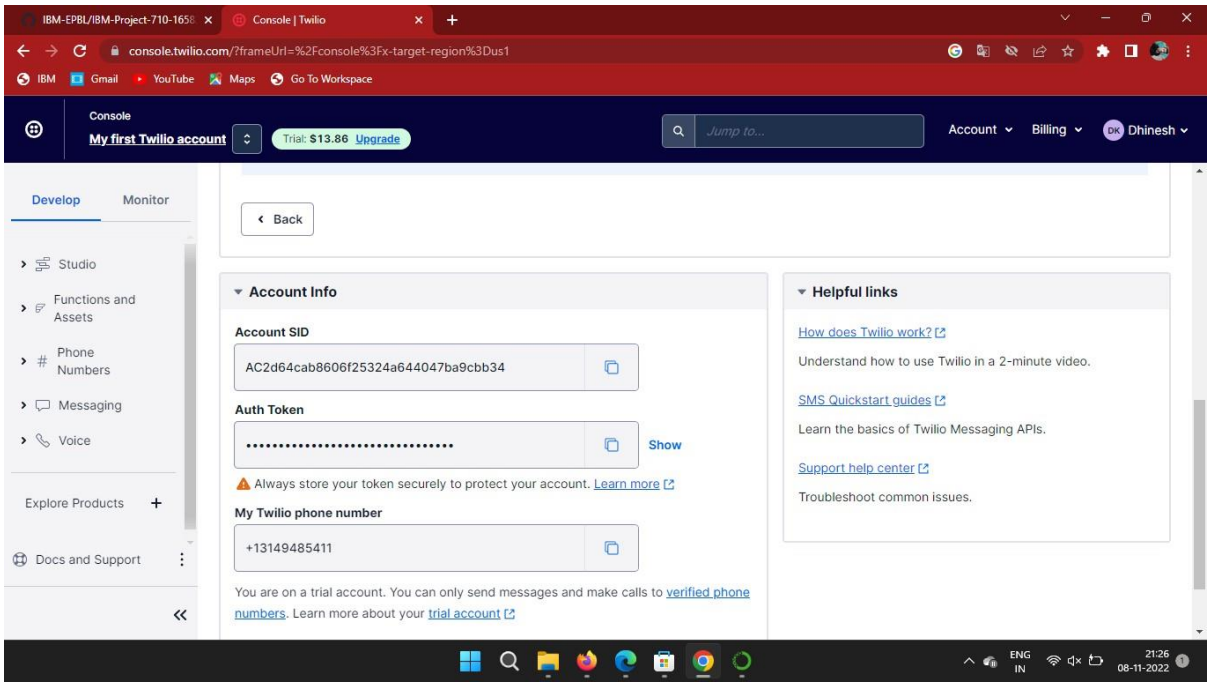
Creating ab account an twilio service and enter into login page



7.2.3 Login Page



7.2.4 Dashboard page and Get a number



7.2.5 Collect Account SID, Auth Token, My Twilio phone number

Coding for sprint

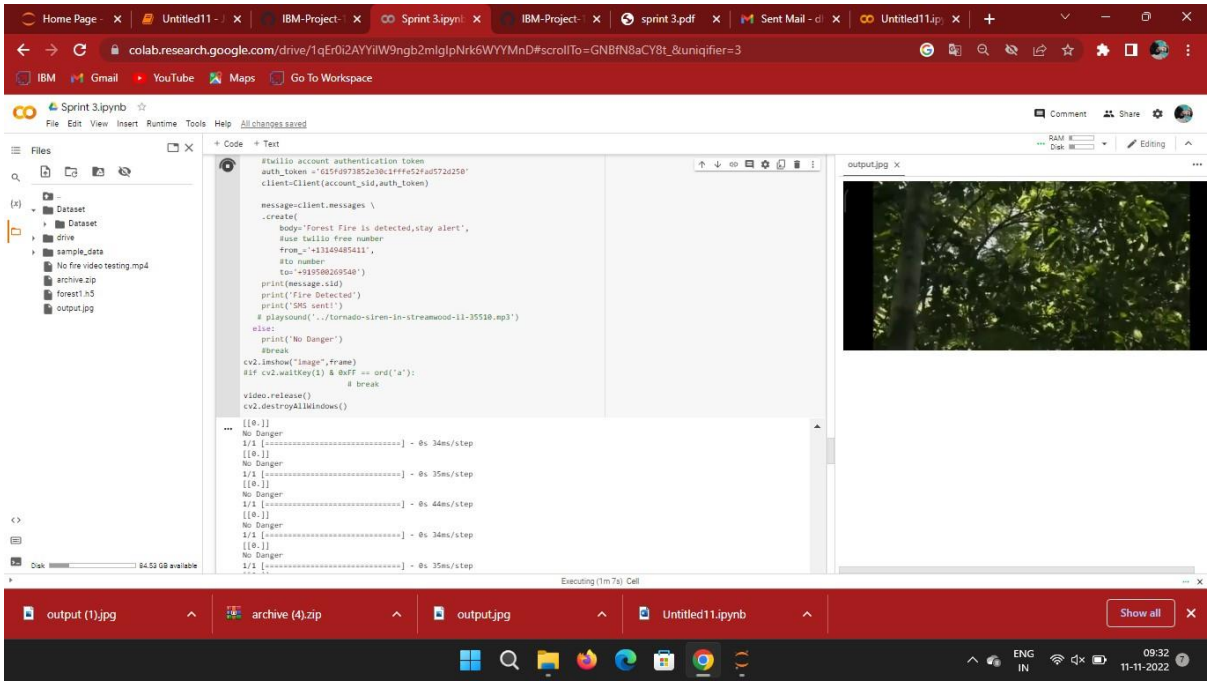
```
import numpy as np
import keras
from keras.preprocessing.image import ImageDataGenerator
#Define the parameters/arguments for ImageDataGenerator class
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,rotation_range=180,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
#Applying ImageDataGenerator functionality to trainset
x_train=train_datagen.flow_from_directory(r'C:\Users\dhine\Downloads\Dataset\Dataset\Dataset\train_set',target_size=(128,128),batch_size=32, class_mode='binary')
#Applying ImageDataGenerator functionality to testset
x_test=test_datagen.flow_from_directory(r'C:\Users\dhine\Downloads\Dataset\Dataset\Dataset\test_set',target_size=(128,128),batch_size=32,
class_mode='binary')
x_train.class_indices
#import model building libraries
#To define Linear initialisation import Sequential
from keras.models import Sequential
#To add layers import Dense
from keras.layers import Dense
#To create Convolution kernel import Convolution2D
from keras.layers import Convolution2D
#import Maxpooling layer
from keras.layers import MaxPooling2D
#import flatten layer
from keras.layers import Flatten
```

```
import warnings
warnings.filterwarnings('ignore')
model = keras.Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Convolution2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Convolution2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Convolution2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(150,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
model.summary()
model.compile(loss = 'binary_crossentropy',optimizer = "adam",
metrics = ["accuracy"])
r = model.fit(x_train, epochs = 5, validation_data = x_train)
predictions = model.predict(x_train)
predictions = np.round(predictions)
predictions
print(len(predictions))
#import load_model from keras.model
from keras.models import load_model
#import image class  from keras
import tensorflow as tf
from tensorflow.keras.preprocessing import image
#import numpy
import numpy as np
#import cv2
import cv2
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
model.save("forest1.h5")
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
model=load_model("forest1.h5")
img=image.load_img(r"C:\Users\dhine\OneDrive\Desktop\Dataset\Dataset\train_set\with
fire\with fire (17).jpg")
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred=model.predict(x_test)
classes_x=np.argmax(pred,axis=1)
mport cv2
import numpy as np
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
```

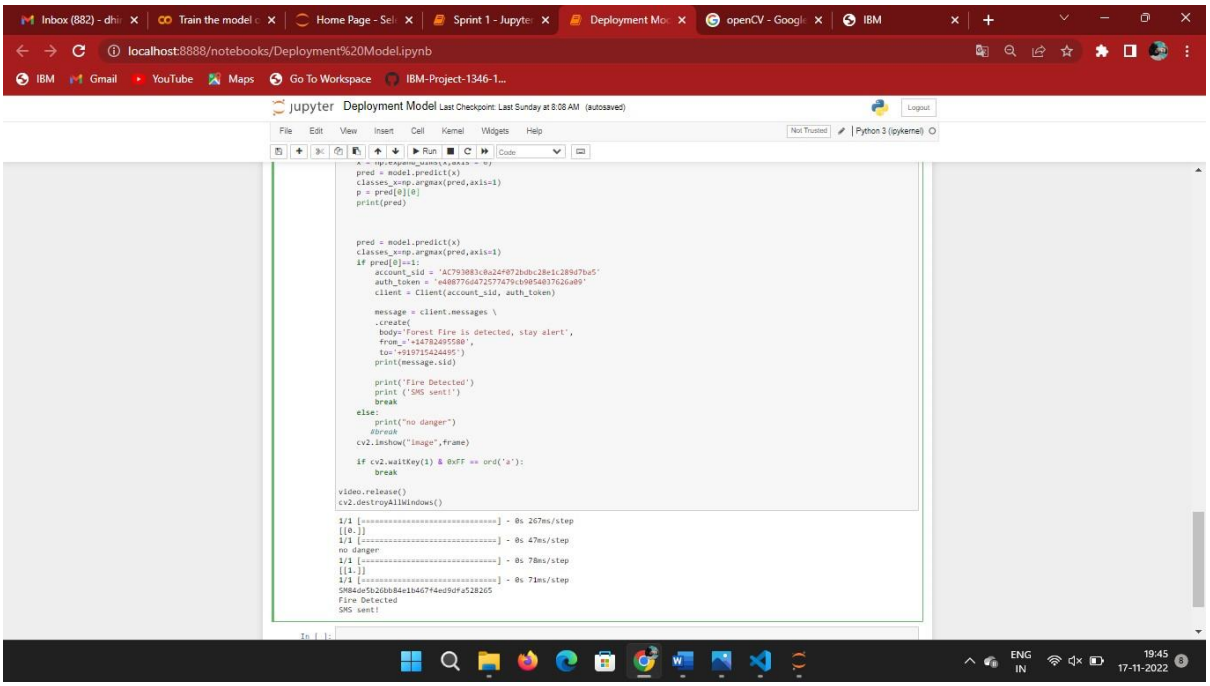
```
from twilio.rest import Client
from playsound import playsound
model = load_model(r"forest1.h5")
video=cv2.VideoCapture(0)
name=['forest','with fire']

while(1):
    success, frame = video.read()
    cv2.imwrite("image.jpg",frame)
    img = image.load_img("image.jpg",target_size = (128,128))
    x  = image.img_to_array(img)
    x = np.expand_dims(x,axis = 0)
    pred = model.predict(x)
    classes_x=np.argmax(pred,axis=1)
    p = pred[0][0]
    print(pred)
    pred = model.predict(x)
    classes_x=np.argmax(pred,axis=1)
    if pred[0]==1:
        account_sid = 'AC793083c0a24f072bdbc28e1c289d7ba5'
        auth_token = 'e408776d472577479cb9054037626a09'
        client = Client(account_sid, auth_token)

        message = client.messages \
        .create(
         body='Forest Fire is detected, stay alert',
         from_='+14782495580',
         to='+919715424495')
        print(message.sid)

        print('Fire Detected')
        print ('SMS sent!')
        break
    else:
        print("no danger")
       #break
    cv2.imshow("image",frame)

    if cv2.waitKey(1) & 0xFF == ord('a'):
        break
video.release()
cv2.destroyAllWindows()
```

Solution for sprint

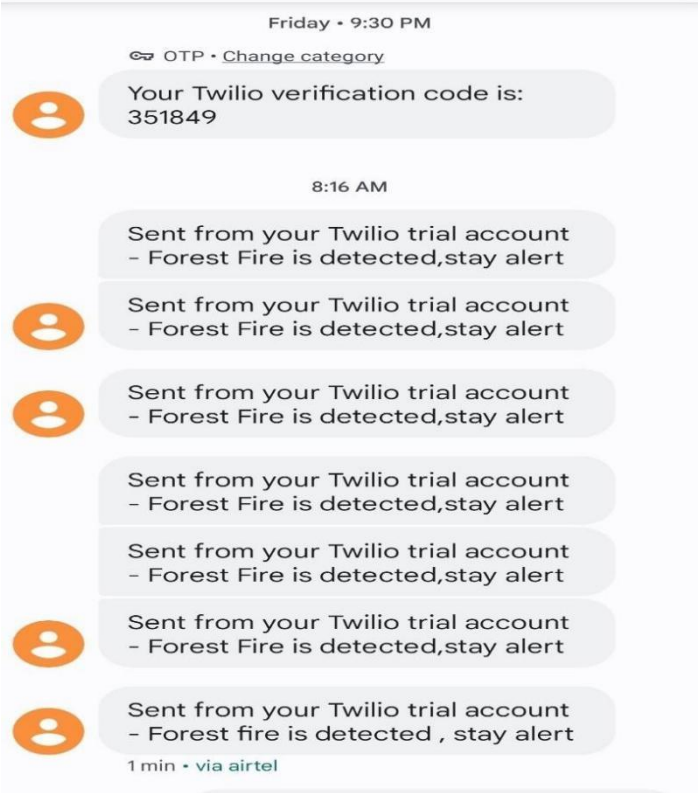
Output



7.2.6 No Danger



7.2.7 Fire Detected



7.2.8 Sending Alert Message

8. TESTING

8.1 TEST CASES

A test case is a document, which has a set of test data, preconditions, expected results and postconditions, developed for a particular test scenario in order to verify compliance against a specific requirement. Test Case acts as the starting point for the test execution, and after applying a set of input values

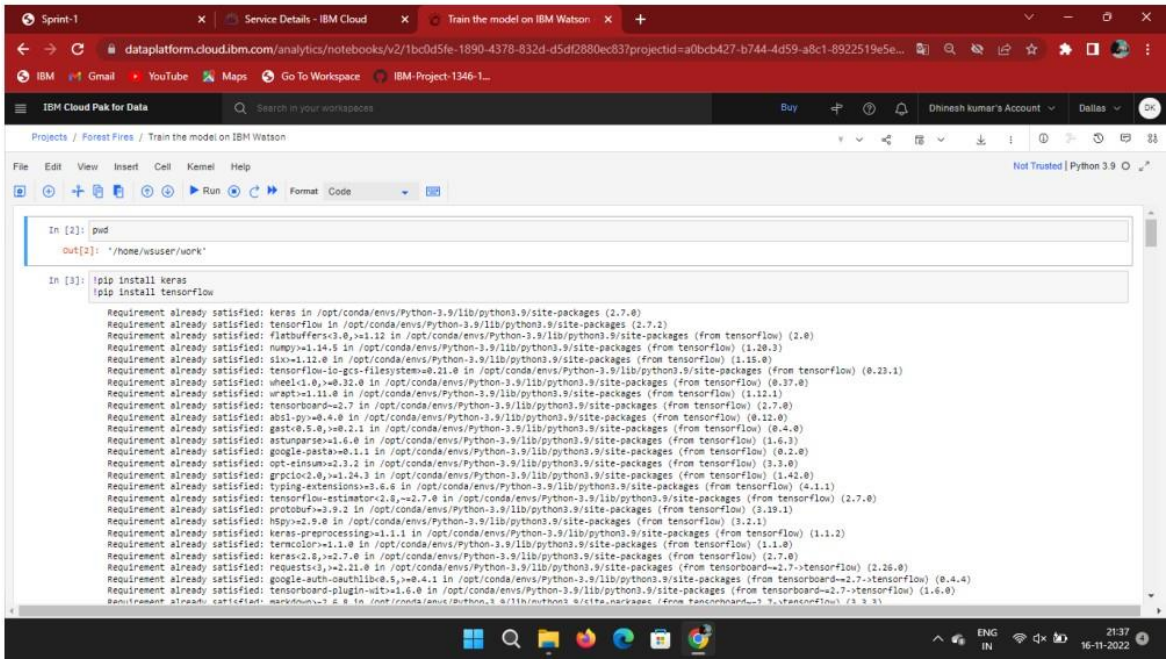
CODING:

```
from io import BytesIO
import zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_3.read()),'r')
file_paths=unzip.namelist()
for path in file_paths:
    unzip.extract(path)

x_test=test_datagen.flow_from_directory(r'/home/wsuser/work/archive/Dataset/Dataset/test_set',
                                        target_size=(128,128),
                                        batch_size=32,
                                        class_mode='binary')

x_train=train_datagen.flow_from_directory(r'/home/wsuser/work/archive/Dataset/Dataset/train_set',
                                          target_size=(128,128),
                                          batch_size=32,
                                          class_mode='binary')

x_train.class_indices

#import model building libraries

#To define Linear initialisation import Sequential
from keras.models import Sequential
#To add layers import Dense
from keras.layers import Dense
#To create Convolution kernel import Convolution2D
from keras.layers import Convolution2D
#import Maxpooling layer
from keras.layers import MaxPooling2D
```

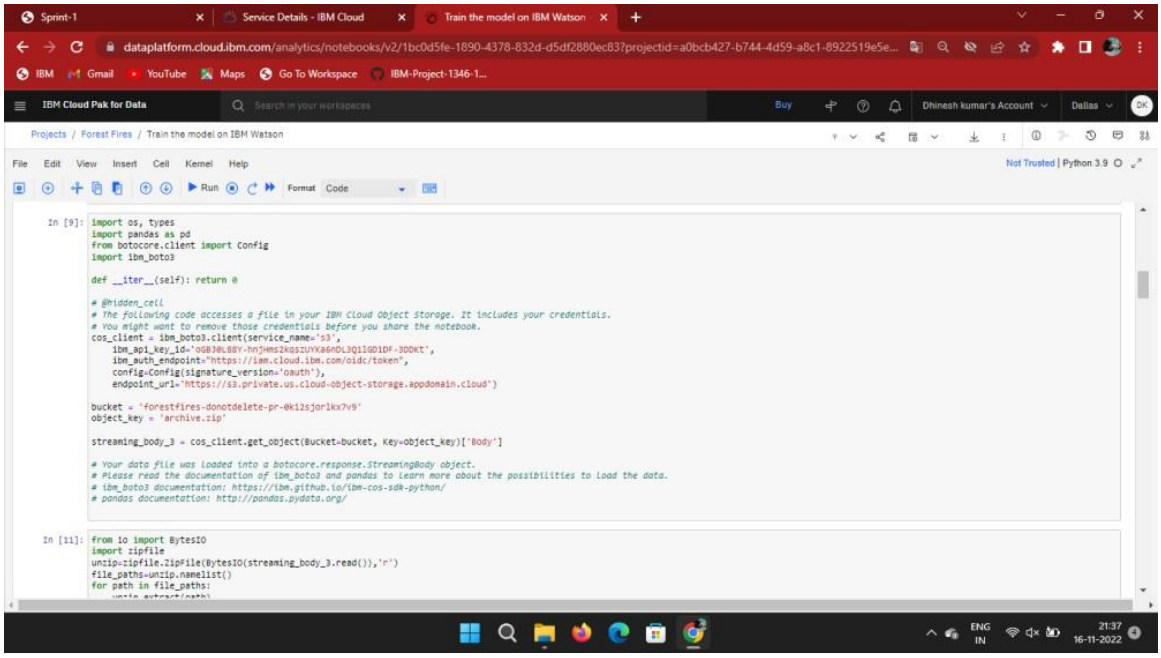
```
#import flatten layer
from keras.layers import Flatten
import warnings
warnings.filterwarnings('ignore')

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

predictions = model.predict(x_train)
predictions = np.round(predictions)

print(len(predictions))

#import load_model from keras.model
from keras.models import load_model
#import image class from keras
import tensorflow as tf
from tensorflow.keras.preprocessing import image
#import numpy
import numpy as np
#import cv2
import cv2

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

model.save("forest1.h5")

!tar -zcvf forest-fires-model_new.tgz forest1.h5

ls -1

!pip install watson-machine-learning-client --upgrade

# Replace the credentials that you got from Watson Machine Learning service
from ibm_watson_machine_learning import APIClient
wml_credentials={
                "url":"https://us-south.ml.cloud.ibm.com",
                "apikey":"fEGuZH13TjZlcGrOR4AN-M-Leni8lcL0uTWaxcbgmTbd"
                }
client=APIClient(wml_credentials)

client=APIClient(wml_credentials)

client.spaces.list()

def guid_from_space_name(client, space_name):
    space = client.spaces.get_details()
    #print(space)
```

```
    return(next(item for item in space['resources'] if item['entity']['name']
 == space_name)['metadata']['id'])

space_uid = guid_from_space_name(client, 'Forest_fires')
print("Space UID = "+ space_uid)

client.software_specifications.list(100)

software_spec_uid=client.software_specifications.get_uid_by_name("tensorflow_
rt22.1-py3.9")
software_spec_uid

model_details = client.repository.store_model(model="forest-fires-
model_new.tgz",meta_props={
    client.repository.ModelMetaNames.NAME:"forest-fires-model",
    client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid})
model_id=client.repository.get_model_uid(model_details)

model_id

client.repository.download(model_id,'my_model.tar.gz')
```

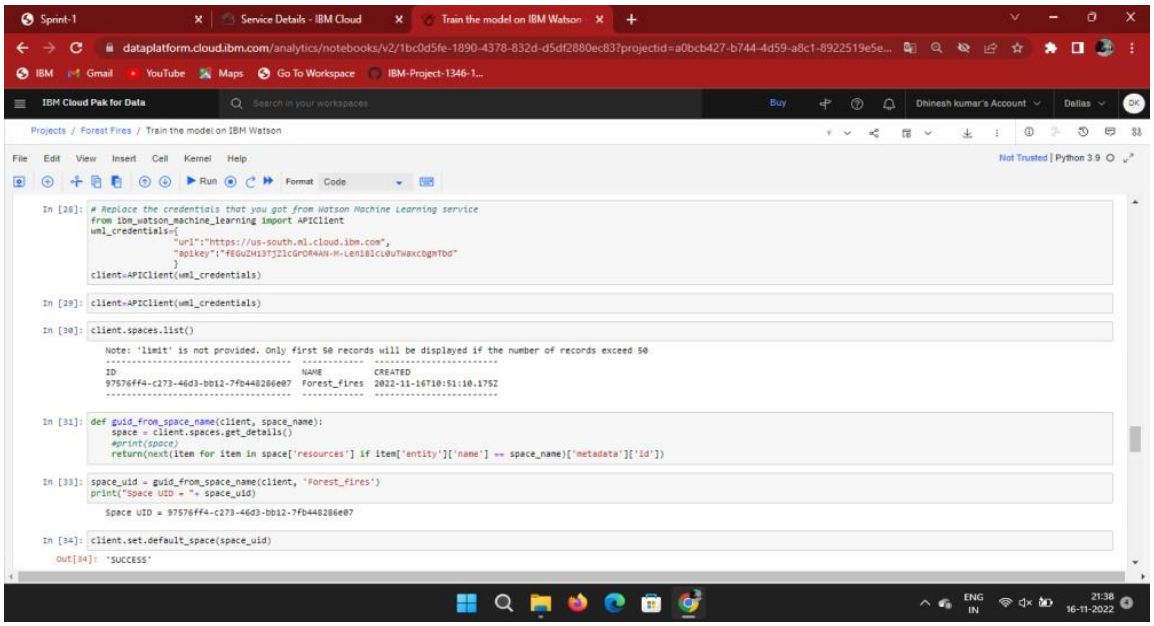
JUPYTER NOTEBOOK(SCREEN SHOT):



8.1.1 Jupiter Notebook



8.1.2 Inserting Streaming Object

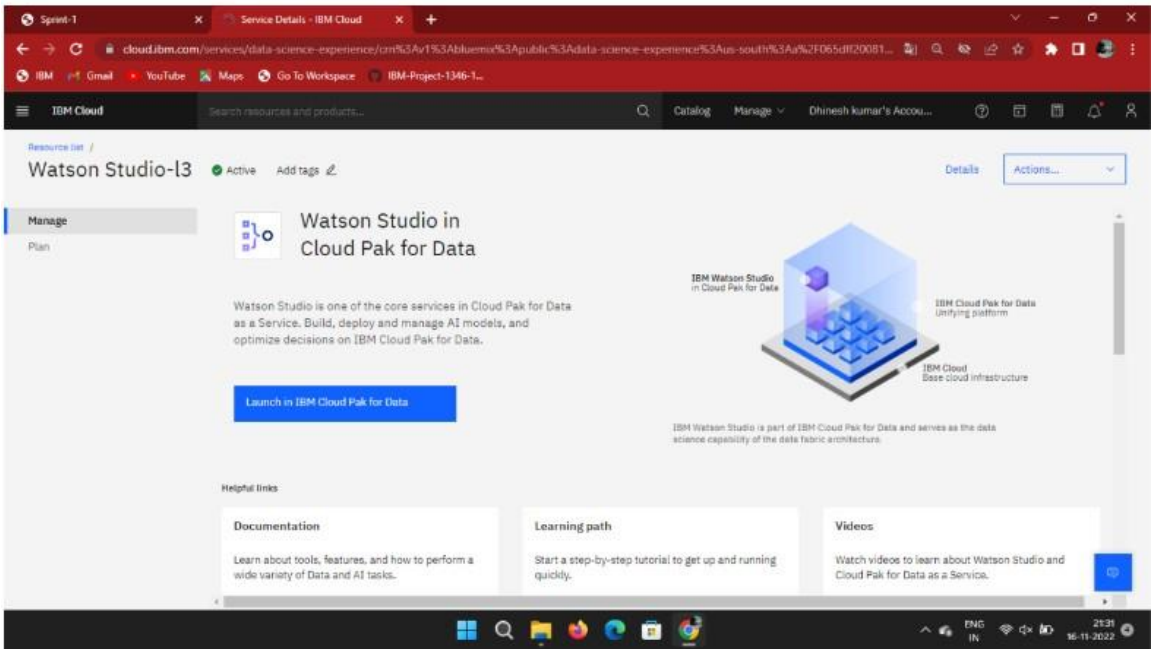


8.1.3 Saving the Model

8.2 USER ACCEPTANCE TESTING:

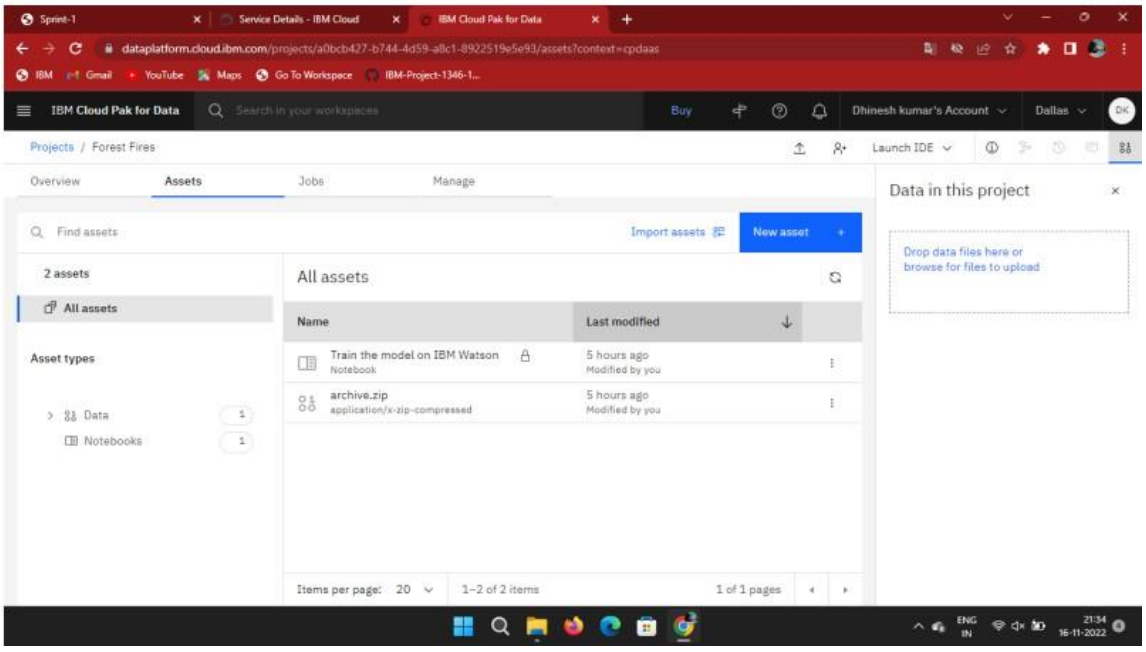
Acceptance testing is a quality assurance (QA) process that determines to what degree an application meets end users' approval. Depending on the organization, acceptance testing might take the form of beta testing, application testing, field testing or end-user testing. Executing the model testing and deploying in the model on Watson Studio creating aa new API for IBM cloud. After getting the API key used to connect with Watson Studio. Executing and store the model in cloud object storage.

GO ON WATSON SERVICES(SCREEN SHOT):



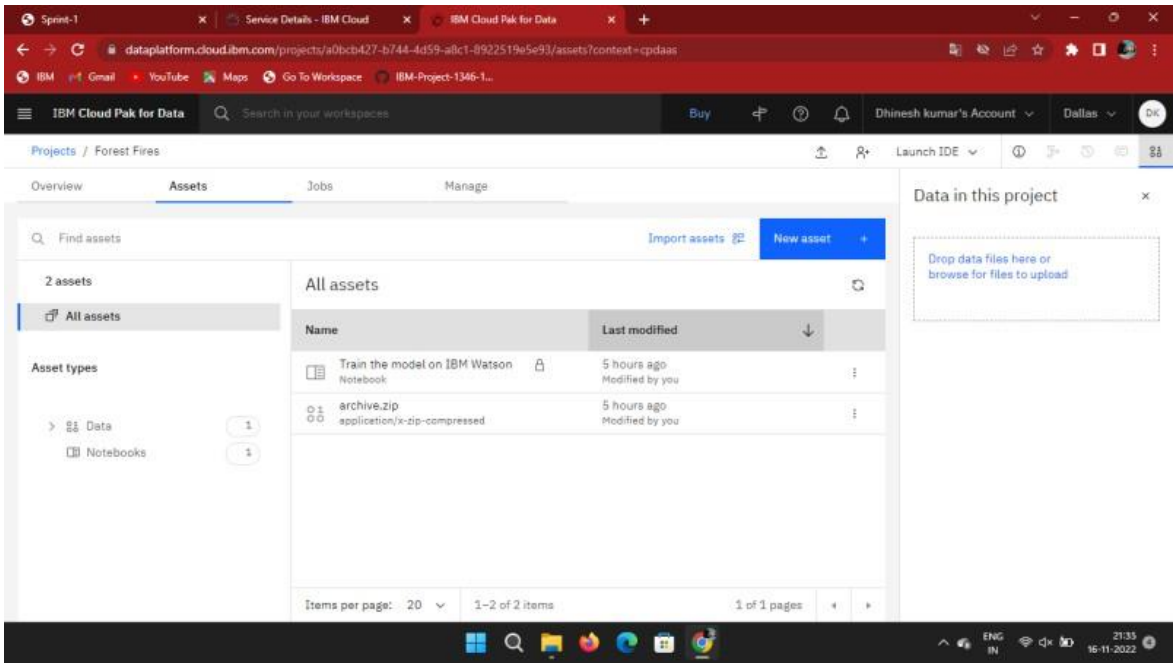
8.2.1 Watson Services

GO ON ASSEST(SCREEN SHOT):



8.2.2 Watson Studio Asset

GO ON IBM_PROJECT(SCREEN SHOT):

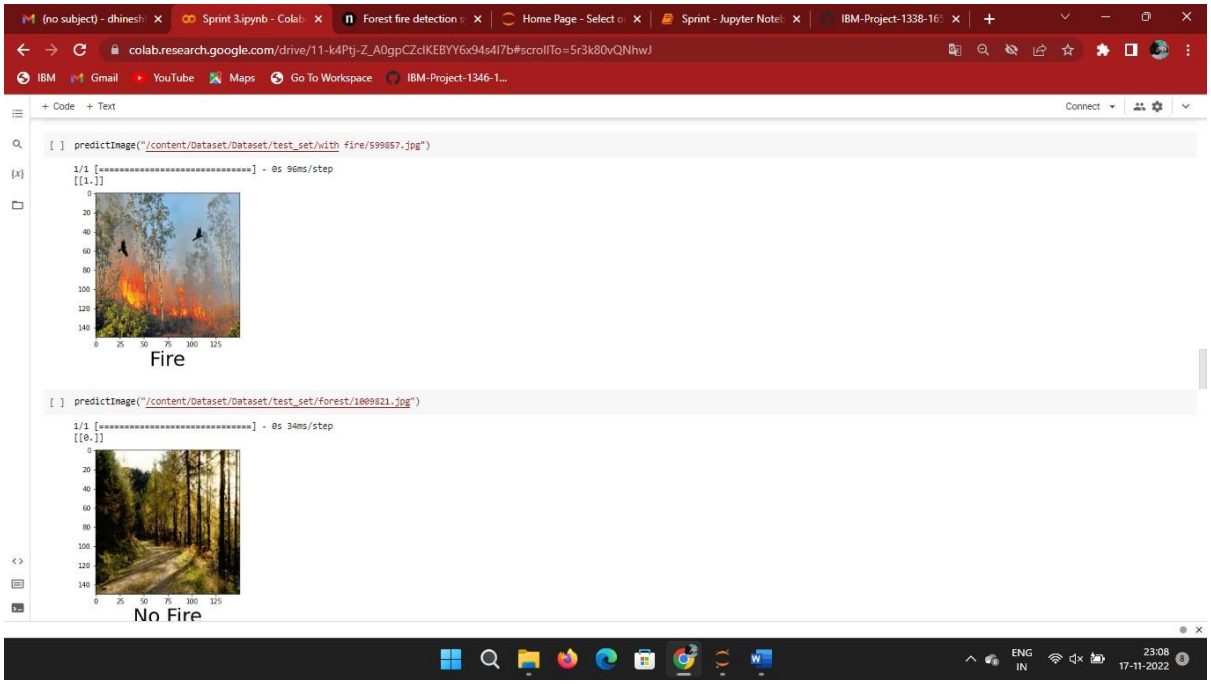


8.2.3 IBM Project

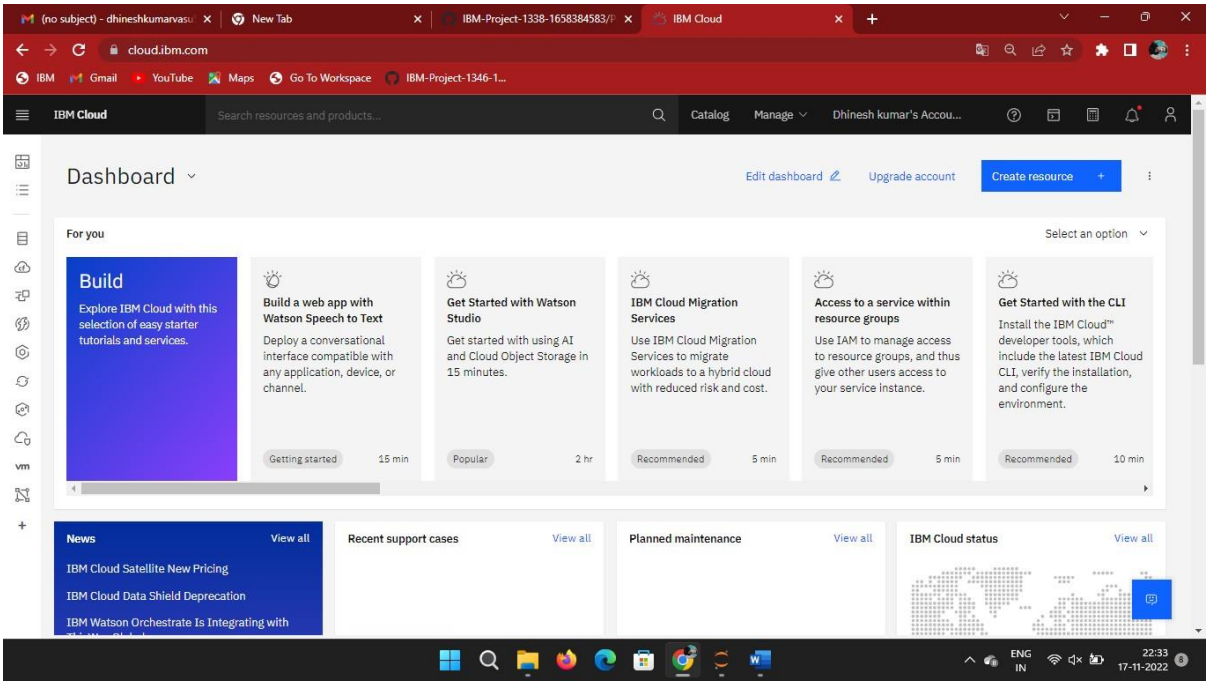
9. RESULT

This model is first trained on a desktop and later loaded into UAV-CPU for testing. It is trained over 100 images for 50,000 steps each. The comparison table of existing works and the proposed model.The model is able to detect fire in ale environmental conditions such as rainy, sunny, snow, and so on, and the training data of the model contains all three examples in equal proportion to avoid variance and bias. The other parameters such as the wind speed and temperature of the regions are analyzed, and fire is detected in such cases. Therefore, in order to improve accuracy in the high smoke zones, we used IR sensors; by utilizing the optical flow, fire is easily detected. The experimental results obtained demonstrate that the model is capable of detecting forest fires and flame regions, 3D modeling of the affected, area and forest fire tracking with satisfactory results, while the problem of using satellite imagery and low-level performance is significantly reduced as well.

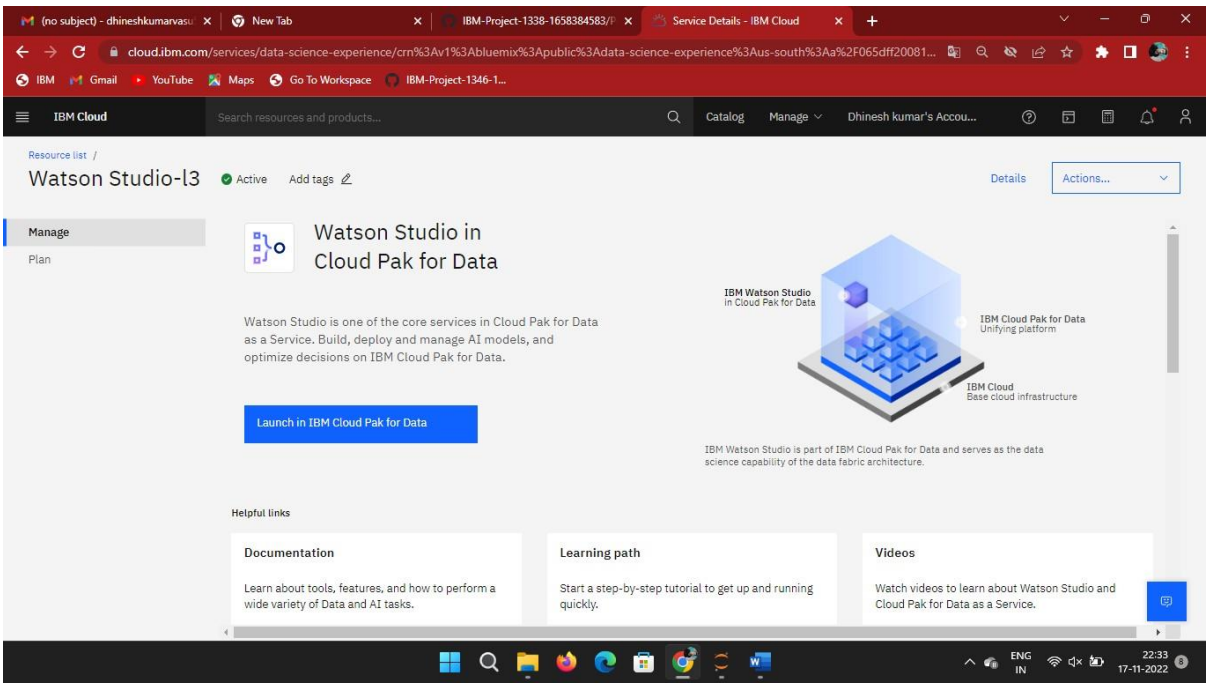
Prediction of result



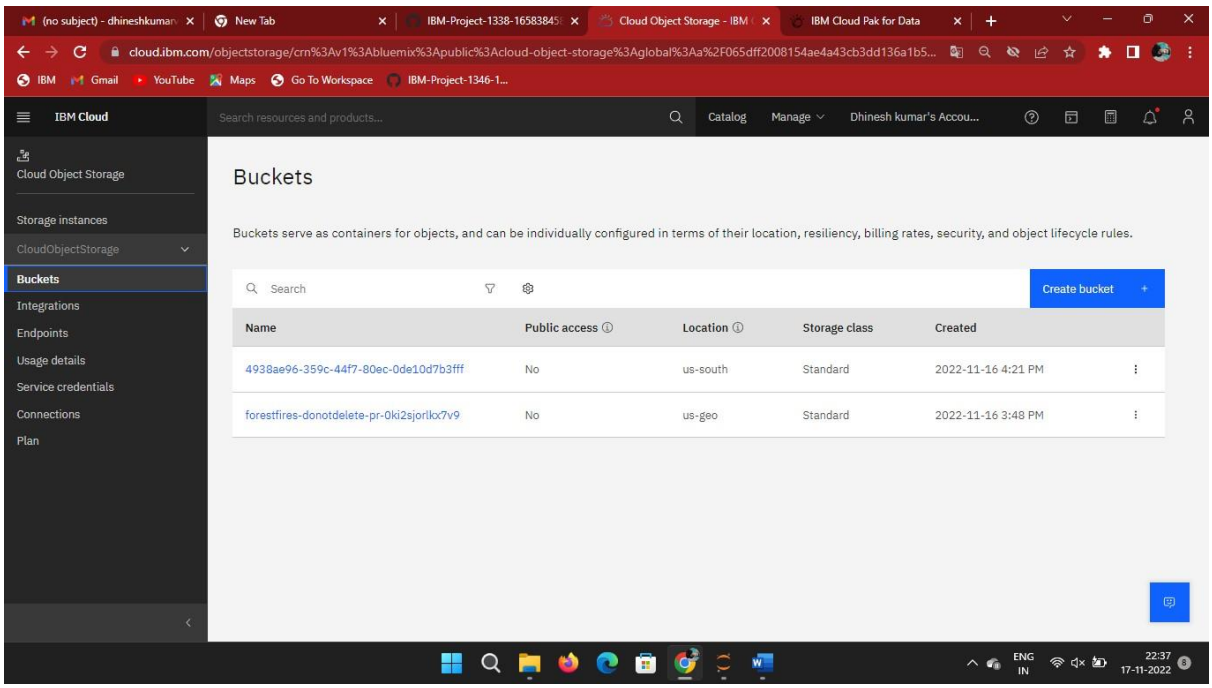
9.1 Prediction of result



9.2 Creating the IBM account and Login into the Dashboard



9.3 Creating the Watson Studio for Testing the code



9.4 Storing the Model on cloud object storage space

9.1 PERFORMANCE METRICS

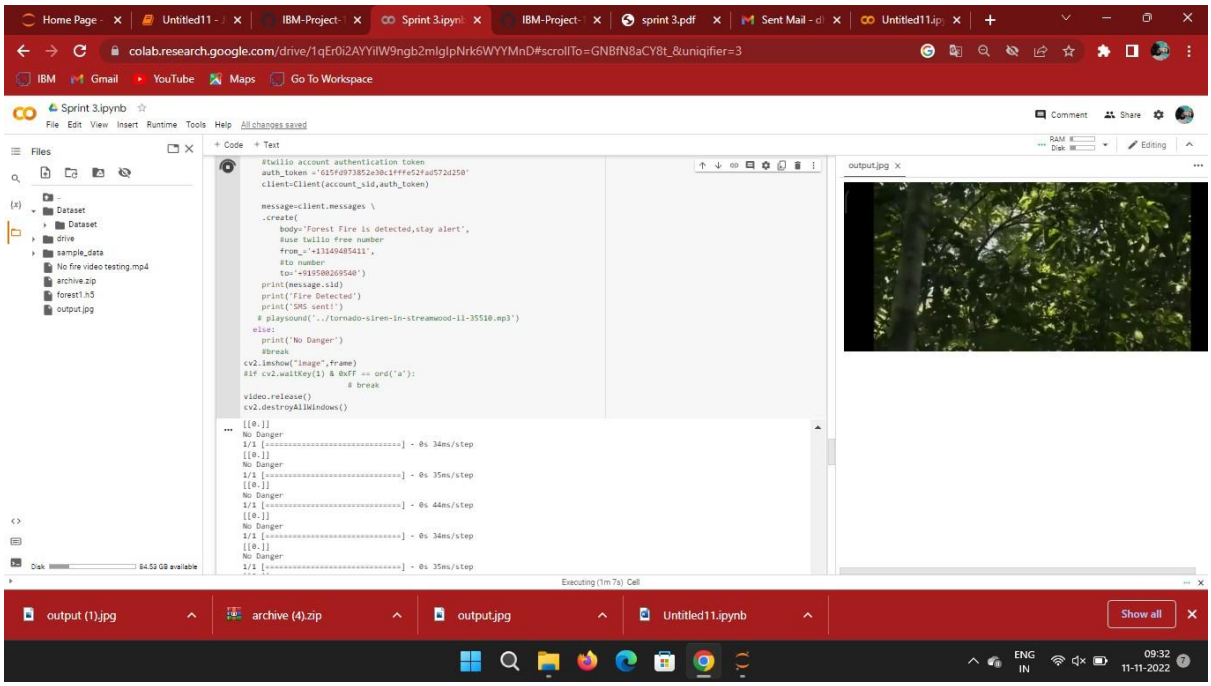
Performance metrics are integral to an organization's success. It's important that organizations select their chief performance metrics and focus on these areas because these metrics help guide and gauge an organization’s success. Key success factors are only useful if they are acknowledged and tracked. Business measurements must also be carefully managed to make sure that they give right answers, and that the right questions are being asked.

The following performance measurement necessities are the same whether you’re measuring business, service, process, or laboratory variables. Together, they constitute a measurement plan.

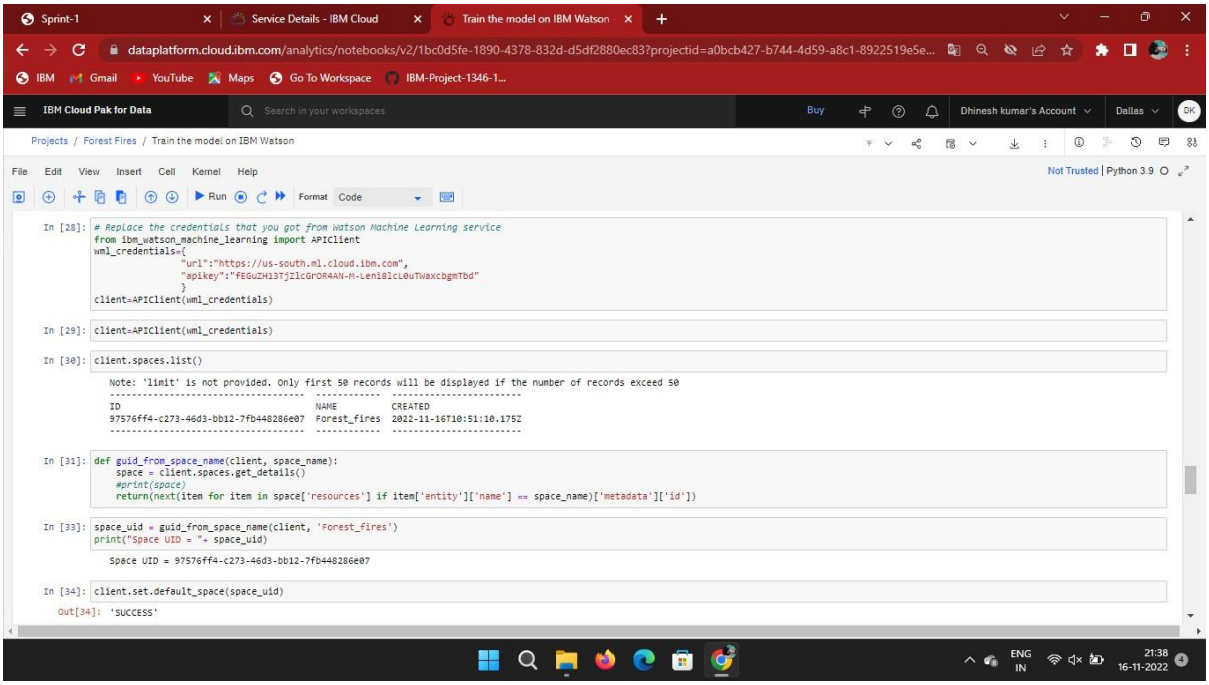
- **Definition of purpose:** Why is a measurement being made? What process or variable is being measured? For what will the resulting data be used?
- **Statement of the required measurement performance indicators (accuracy, precision, resolution):** These may be determined by organizational policy, adherence to a

published standard or an analysis of the requirements based on use, ability to measure, or more.

- **The unit or variable being measured** and a statement as to why measuring that particular variable supports the purpose of the measurement.
- **An operational definition:** A detailed, yet easily understood, description of the measurement process.
    - Example: An operational definition for the measurement of a sales-fulfilment cycle time might be, “The time interval to be measured begins when the sales department places a validated order form in the sales order out box, and ends when the completed, boxed order is delivered to the loading dock for pickup.”
- **An analysis plan:** A typical example is a monthly report that makes comparisons to the previous month, year over year, and year to date. The different time frames provide greater context and allow the data to be presented graphically.



9.1.1 Detecting the Forest Fires



9.1.2 Successfully download the model

10. ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- It can also gauge temperature, humidity, and air pressure to create a climate map of the forest. This map provides the means to assess the risk of fire.
- The proposed system detects the forest fire at a faster rate compared to existing system. It has enhanced data collection feature. The major aspect is that it reduces false alarm and also has accuracy due to various sensors present. It minimizes the human effort as it works automatically. This is very affordable due to which can be easily accessed. The main objective of our project is to receive an alert message through an app to the respective user.
- The fast response to fires**.**
- Built fire datasets will be a valuable mean to assist firefighters and other emergency responders facing efficiently wildfires**.**
- It has enhanced data collection feature. The major aspect is that it reduces false alarm and also has accuracy due to various sensors present.
- The location of fire is sensed using this method not just the radiation,
- The captured images can be analyzed and it can be used for future purposes and storage
- It can be used for outdoor places which covers large area

DISADVANTAGES:

- The electrical interference diminishes the effectiveness of radio receiver. The main drawback is that it has less coverage range areas.
- 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.
- Although there are many drawbacks, such as inefficiency, power consumption, latency, accuracy and implementation costs.
- The transfer of images is a heavy load for wireless sensor networks in relation to their limited resources.

11.CONCLUSION

In this project we have briefly presented methods for early forest fire detection, including part of their characteristics and main components. We have also analysed some of the benefits.

The latest UAVs are equipped with both visible and infrared cameras, improving the detection accuracy and allowing night operation; however, they are affected by weather conditions and, in many cases, their flight time is limited

In conclusion, jungles and forests on the Earth are vanishing dramatically due to man-caused fires and various other reasons, life is, hence, becoming more critical for all creatures.

12.FUTURE SCOPE

Evolution emerges in the processing, computation, and algorithms.This strives many researchers to pay attention in many domains where they work in the processing of surveillance video streams so that abnormal or unusual actions could be detected.The usage of UAVs is recommended in the detection of forest fire due to the high mobility and ensures the coverage areas at various altitudes and locations at a low cost.

As our future works, focus to meet practical detection and meet the necessity of early detection including the generation of the mixed reality model of the forest fire area that gives more information, and prevention analysis will be made easy.

The 3D modeling techniques presented in this paper can also be extended to various natural disaster prediction models.However, this model is sensitive to the forest with dense fogs and clouds. This is because smoke appears as the same as fog, and the model may misclassify the fog as smoke.

Hence, an efficient and scalable UAV is used for detection.The accuracy of the detection rate achieved through this model is 91%. The proposed model outperforms the other existing techniques in terms of detecting in the early stage.

13.APPENDIX

13.1 SOURCE CODE

```
import cv2
import numpy as np
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
from twilio.rest import Client
from playsound import playsound
model = load_model(r"forest1.h5")
video=cv2.VideoCapture(0)
name=['forest','with fire']

while(1):
    success, frame = video.read()
    cv2.imwrite("image.jpg",frame)
    img = image.load_img("image.jpg",target_size = (128,128))
    x  = image.img_to_array(img)
    x = np.expand_dims(x,axis = 0)
    pred = model.predict(x)
    classes_x=np.argmax(pred,axis=1)
    p = pred[0][0]
    print(pred)

    pred = model.predict(x)
    classes_x=np.argmax(pred,axis=1)
    if pred[0]==1:
        account_sid = 'AC793083c0a24f072bdbc28e1c289d7ba5'
        auth_token = 'e408776d472577479cb9054037626a09'
        client = Client(account_sid, auth_token)
```

```
            message = client.messages \
            .create(
             body='Forest Fire is detected, stay alert',
             from_='+14782495580',
             to='+919715424495')
            print(message.sid)

            print('Fire Detected')
            print ('SMS sent!')
            break
        else:
            print("no danger")
           #break
        cv2.imshow("image",frame)

        if cv2.waitKey(1) & 0xFF == ord('a'):
            break

    video.release()
    cv2.destroyAllWindows()
```

13.2 GIT HUB & PROJECT DEMO LINK

DEMO VIDEO LINK:

https://drive.google.com/file/d/1UjpyElXaR8NmcPy_kmWmOz-4XG92nwlZ/view?usp=sharing

GIT HUB LINK:

https://github.com/IBM-EPBL/IBM-Project-1338-1658384583