

COMPUTER VISION-BASED EARLY FIRE DETECTION USING OPEN CV AND MACHINE LEARNING

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering degree in
Computer Science and Engineering

By

PETA MANISH KUMAR REDDY(Reg. No.38110404)

POTU LAKSHMI PRANEETH REVANTH KUMAR(Reg. No.38110414)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY) Accredited with Grade "A" by
NAAC JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI –
600 119

MARCH - 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **PETA MANISH KUMAR REDDY(Reg. No.38110404)** and **POTU LAKSHMI PRANEETH REVANTH KUMAR(Reg. No.38110414)** who carried out the project entitled **“COMPUTER VISION-BASED EARLY FIRE DETECTION USING OPEN CV AND MACHINE LEARNING ”** under my supervision from **August 2021** to **March 2022**.

Internal Guide

Dr.S.Raja Shree M.E., P.h.D.,

Head of the Department

Dr. S. Vigneswari, M.E., Ph.D.,

Dr. L. Lakshmanan, M.E., Ph.D.,

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

DECLARATION

**I PETA MANISH KUMAR REDDY and POTU LAKSHMI PRANEETH
REVANTH KUMAR** hereby declare that the Project Report entitled “
**COMPUTER VISION-BASED EARLY FIRE DETECTION USING OPEN
CV AND MACHINE LEARNING** ” is done by me under the guidance of
Dr.S.Raja Shree M.E.,P.h.D., Department of Computer Science and
Engineering at Sathyabama Institute of Science and Technology is
submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.S.Raja Shree M.E.,P.h.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The project aimed to detect fire by using the image processing technology that will alert people by early detection of fire. As there are many automatic fire alarm systems already existed like the sensor method, that has some limitations and designed to sense fire with the smoke, limited areas. To reduce limitations and to optimize with new technology, the project is proposed. The project is implemented by using pycharm IDE and to connect the webcam as hardware. Webcam is taken as an input source, which captures the video feed from the surrounding and feeds into the system for analysis. The entire code is written in pure python language using the open CV library for image processing. The theoretical parts emphasize more in computer vision, machine learning, image processing, color model, and the working algorithm of the project to detect the fire. The project gives a better understanding of object detection with the computer and the use of these technologies in different forms and uses.

Table of contents

Chapter No.	Title	Page No.
	ABSTRACT	V
	LIST OF FIGURES	VII
1.	INTRODUCTION	9
	1.1 Overview	9
	1.2 Scope and Objective	10
	1.3 Domain Overview	10
	1.3.1 Machine Learning Model	11
	1.3.2 Data Pre-processing	12
	1.3.3 Supervised Learning	14
	1.3.4 Regression	16
	1.3.5 Unsupervised Learning	17
	1.3.6 Training and testing the model of data	26
	1.3.7 Validation Set	19
2.	LITERATURE SURVEY	22
	2.1 Analysis of literature	22
	2.2 Literary review	23
3.	SYSTEM ANALYSIS	26
	3.1 Existing System	26
	3.2 Problem Statement	26
	3.3 Proposed System	26
	3.4 System Design	28

	3.5 Haar Like Features	29
	3.6 Working Principle	32
	3.7 System Requirements	33
4.	RESULTS	34
5.	CONCLUSION AND FUTURE WORK	35
	5.1 Conclusion	35
	5.2 Limitation and Future Work	35
6.	REFERENCE PAPERS	36
7.	APPENDIX	38
	A. Sample Code	38
	B. Plagarism Report	43

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
1.1	Supervised Architecture	14
1.2	Unsupervised Architecture	14
1.3	RGB Colours	15
1.4	Prediction Of IQ	16
1.5	Original Clustered Data	17
1.6	Clustered Data	17
1.7	Classifications Of Machine Learning	18
1.8	Training Set	18
1.9	Validation Set	19
1.10	Confusion Matrix	20
3.1	Block Diagram	27
3.2	Flow Chart	28
3.3	Haar Features	30
4.1	Result ScreenShot	34

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Computer Vision based fire detection using image processing has the potential to be useful in conditions in which conventional methods cannot be adopted. The fire detection algorithm uses visual characteristics of fires like brightness, color, spectral texture, spectral flicker, and edge trembling to discriminate them from other visible stimuli. There are various fire detection techniques such as infrared sensor, a thermal detector, smoke detector, flame detector, and optical smoke detector. These methods are not always reliable as they do not always detect the fire itself but detect one or more phenomena resulting from fire, such as smoke, heat, infrared, ultraviolet light radiation or gas, which could be produced in other ways and hence, produces many false alarms. By the help of computer vision and image processing techniques, it is possible to get better results than conventional systems because images can provide more reliable information.

In recent times, research on detection of flame and smoke using surveillance cameras with machine vision has gained momentum. The image processing approach involves the extraction of the smoke-plume or flame from the background by using frame difference technologies. In the case of the segmentation of fire features, color processing scores over gray-scale processing. Color processing can avoid the generation of false alarms due to variations in the lighting conditions, e.g. natural background illumination, better than gray-scale processing. Further, a video camera is a volume sensor, and potentially monitors a larger area. The traditional point sensor looks at a point in space. Since the point sensor may not be affected by smoke or flame, fire would be undetected. However, vision-based flame and smoke detection still has great technical challenges, since flame and smoke are non-rigid objects, with none of the primitive image features and variability in density, lighting, etc.

1.2 SCOPE AND OBJECTIVE

The proposed software product is used in various areas like in parking areas, road traffic congestions, agriculture, farming, there are many uses of object detection. we are using the Haar Cascade classifier, which is very popular in object detection through the image or any other video feeds. In future, this approach can be extended by for other disasters as well.

1.3 DOMAIN OVERVIEW

Steps of Machine Learning

- Step 1: Gathering Data
- Step 2: Preparing that Data
- Step 3: Choosing a Model
- Step 4: Training
- Step 5: Evaluation
- Step 6: Hyper parameter Tuning
- Step 7: Prediction

Introduction:

In this blog, we will discuss the workflow of a Machine learning project this includes all the steps required to build the proper machine learning project from scratch. We will also go over data pre-processing, data cleaning, feature exploration and feature engineering and show the impact that it has on Machine Learning Model Performance. We will also cover a couple of the pre-modelling steps that can help to improve the model performance.

Python Libraries that would be need to achieve the task:

1. Numpy

2. Pandas

3. Sci-kit Learn

4. Matplotlib

Understanding the machine learning workflow

We can define the machine learning workflow in 3 stages.

1. Gathering data
2. Data pre-processing
3. Researching the model that will be best for the type of data
4. Training and testing the model
5. Evaluation

1.3.1 MACHINE LEARNING MODEL

The machine learning model is nothing but a piece of code; an engineer or data scientist makes it smart through training with data. So, if you give garbage to the model, you will get garbage in return, i.e. the trained model will provide false or wrong prediction

Gathering Data:

The process of gathering data depends on the type of project we desire to make, if we want to make an ML project that uses real-time data, then we can build an IoT system that using different sensors data. The data set can be collected from various sources such as a file, database, sensor and many other such sources but the collected data cannot be used directly for performing the analysis process as there might be a lot of missing data, extremely large values, unorganized text data or noisy data. Therefore, to solve this problem Data Preparation is done. We can also use some free data sets which are present on the internet. Kaggle and UCI Machine learning Repository are the repositories that are used the most for making Machine learning models. Kaggle is one of the most visited websites that is used for practicing machine learning algorithms, they also host competitions in

which people can participate and get to test their knowledge of machine learning.

1.3.2 Data pre-processing:

Data pre-processing is one of the most important steps in machine learning. It is the most important step that helps in building machine learning models more accurately. In machine learning, there is an 80/20 rule. Every data scientist should spend 80% time for data pre-processing and 20% time to actually perform the analysis. Data pre-processing is a process of cleaning the raw data i.e. the data is collected in the real world and is converted to a clean data set. In other words, whenever the data is gathered from different sources it is collected in a raw format and this data isn't feasible for the analysis. Therefore, certain steps are executed to convert the data into a small clean data set, this part of the process is called as data pre-processing.

As we know that data pre-processing is a process of cleaning the raw data into clean data, so that can be used to train the model. So, we definitely need data pre-processing to achieve good results from the applied model in machine learning and deep learning projects. Most of the real-world data is messy, some of these types of data are:

- **Missing data:** Missing data can be found when it is not continuously created or due to technical issues in the application (IOT system).
- **Noisy data:** This type of data is also called outliers, this can occur due to human errors (human manually gathering the data) or some technical problem of the device at the time of collection of data.
- **Inconsistent data:** This type of data might be collected due to human errors (mistakes with the name or values) or duplication of data.

How can data pre-processing be performed?

These are some of the basic pre — processing techniques that can be used to convert raw data.

- Conversion of data: As we know that Machine Learning models can only handle numeric features, hence categorical and ordinal data must be somehow converted into numeric features.
- Ignoring the missing values: Whenever we encounter missing data in the data set then we can remove the row or column of data depending on our need. This method is known to be efficient but it shouldn't be performed if there are a lot of missing values in the dataset.
- Filling the missing values: Whenever we encounter missing data in the data set then we can fill the missing data manually, most commonly the mean, median or highest frequency value is used.
- Machine learning: If we have some missing data then we can predict what data shall be present at the empty position by using the existing data.
- Outliers detection: There are some error data that might be present in our data set that deviates drastically from other observations in a data set.
[Example: human weight = 800 Kg; due to mistyping of extra 0]

Searching the suitable model :

Our main goal is to train the best performing model possible, using the pre-processed data. The Machine learning is divided into two parts namely Supervised Learning and Unsupervised Learning. So these methods will help us with the best model.

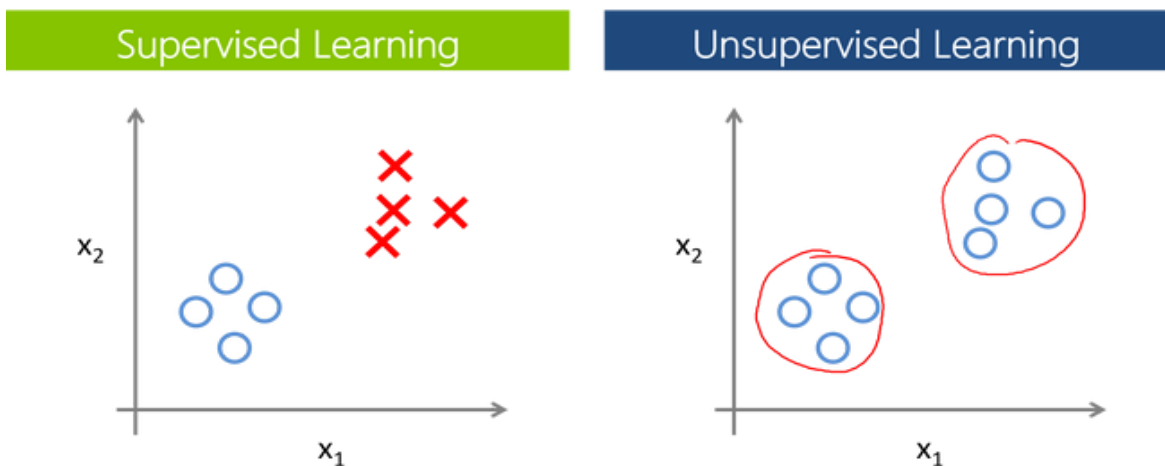


Fig 1.1 Supervised Learning

Fig 1.2 Unsupervised Learning

1.3.3 Supervised Learning:

In Supervised learning, an AI system is presented with data which is labelled, which means that each data is tagged with the correct label.

The supervised learning is categorized into 2 other categories which are “**Classification**” and “**Regression**”.

Classification:

Classification problem is when the target variable is **categorical** (i.e. the output could be classified into classes — it belongs to either Class A or B or something else).

A classification problem is when the output variable is a category, such as “red” or “blue”, “disease” or “no disease” or “spam” or “not spam”.



Fig 1.3 RGB Colours

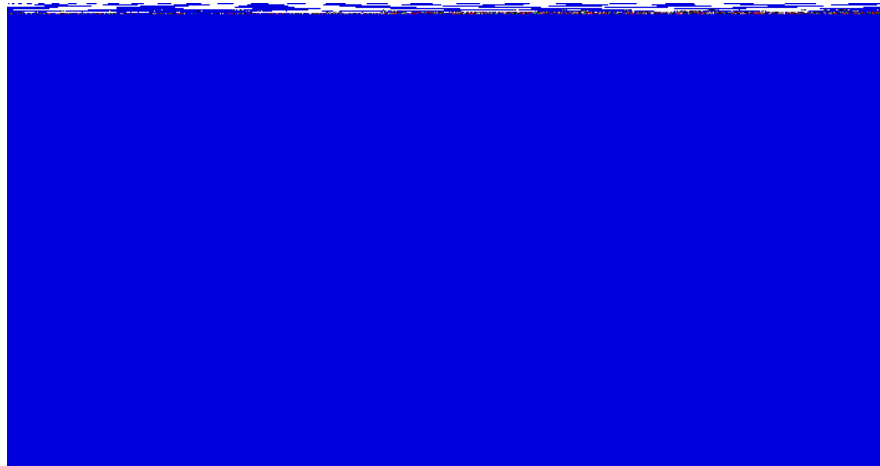
As shown in the above representation, we have 2 classes which are plotted on the graph i.e. red and blue which can be represented as ‘setosa flower’ and ‘versicolor flower’, we can image the X-axis as there ‘Sepal Width’ and the Y-axis as the ‘Sepal Length’, so we try to create the best fit line that separates both classes of flowers.

These some most used classification algorithms.

- K-Nearest Neighbor
- Naive Bayes
- Decision Trees/Random Forest
- Support Vector Machine
- Logistic Regression

1.3.4 Regression:

Regression is a technique for investigating the relationship between independent variables or features and a dependent variable or outcome. It's used as a method for predictive modelling in machine learning, in which an algorithm is used to predict continuous outcomes.



1.4 Prediction of IQ

As shown in the above representation, we can imagine that the graph's X-axis is the 'Test scores' and the Y-axis represents 'IQ'. So we try to create the best fit line in the given graph so that we can use that line to predict any approximate IQ that isn't present in the given data.

These are some most used regression algorithms.

- Linear Regression
- Support Vector Regression
- Decision Tree/Random Forest

- Gaussian Processes Regression
- Ensemble Methods

1.3.5 Unsupervised Learning:

The unsupervised learning is categorized into 2 other categories which are “**Clustering**” and “**Association**”.

Clustering:

A set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.

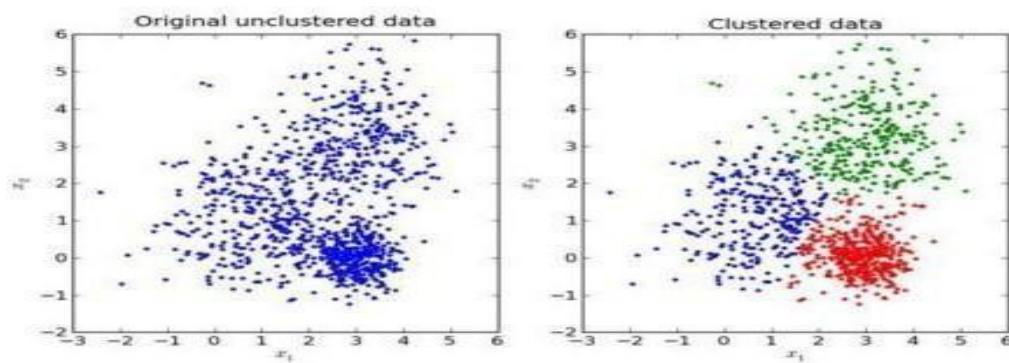


Fig 1.5 Original Clustered Data

Fig 1.6 Clustered Data

Methods used for clustering are:

- Gaussian mixtures
- K-Means Clustering
- Boosting

- Hierarchical Clustering
- K-Means Clustering
- Spectral Clustering

Overview of models under categories:

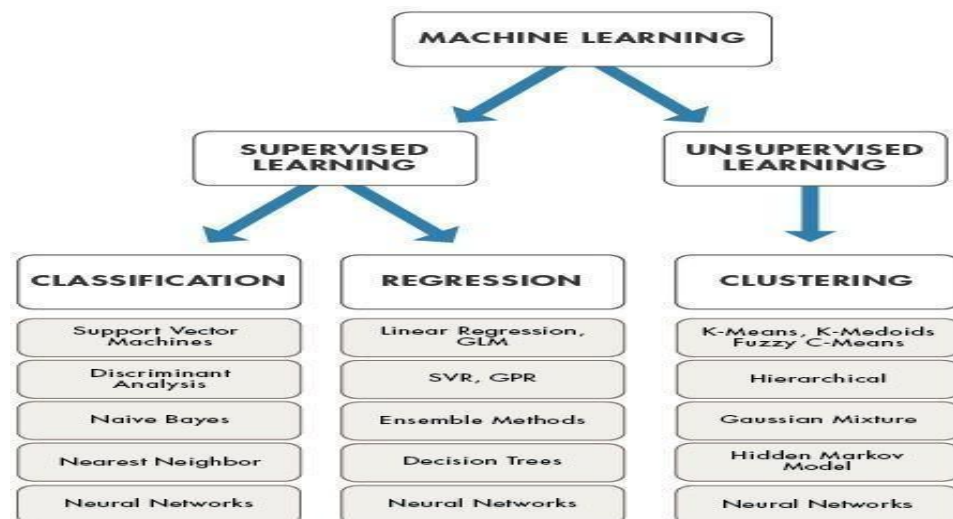


Fig 1.7 Classifications of Machine Learning

1.3.6 Training and testing the model on data:

For training a model we initially split the model into 3 three sections which are 'Training data' , 'Validation data' and 'Testing data'. You train the classifier using 'training data set', tune the parameters using 'validation set' and then test the performance of your classifier on unseen 'test data set'. An important point to note is that during training the classifier only the training and/or validation set is available. The test data set must not be used during training the classifier. The test set will only be available during testing the classifier.



Fig 1.8 Training set

Training set:

The training set is the material through which the computer learns how to process information. Machine learning uses algorithms to perform the training part. A set of data used for learning, that is to fit the parameters of the classifier.

1.3.7 Validation set:

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. A set of unseen data is used from the training data to tune the parameters of a classifier.

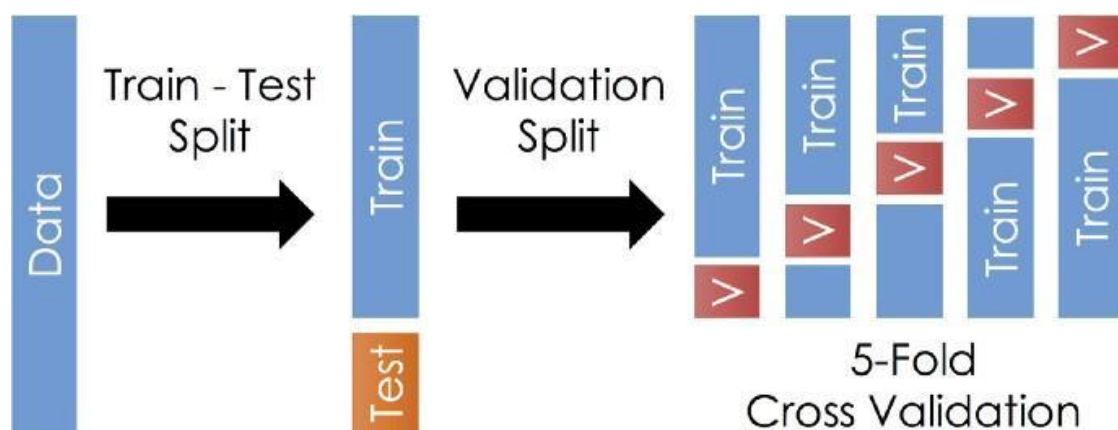


Fig 1.9 Validation Set

Once the data is divided into the 3 given segments we can start the training process.

In a data set, a training set is implemented to build up a model, while a test (or validation) set is to validate the model built. Data points in the training set are excluded from the test (validation) set. Usually, a data set is divided into a training set, a validation set (some people use 'test set' instead) in each iteration, or divided into a training set, a validation set and a test set in each iteration.

The model uses any one of the models that we had chosen in step 3/ point 3. Once the model is trained we can use the same trained model to predict using the testing data i.e. the unseen data. Once this is done we can develop a confusion matrix, this tells us how well our model is trained. A confusion matrix has 4 parameters, which are 'True positives', 'True Negatives', 'False Positives' and 'False Negative'. We prefer that we get more values in the True negatives and true positives to get a more accurate model. The size of the Confusion matrix completely depends upon the number of classes.

n=165	Predicted: NO	Predicted: YES
	Actual: NO	Actual: YES
	50	10
	5	100

1.10 Confusion Matrix

True positives : These are cases in which we predicted TRUE and our predicted output is correct.

True negatives : We predicted FALSE and our predicted output is correct.

False positives : We predicted TRUE, but the actual predicted output is

FALSE.

False negatives : We predicted FALSE, but the actual predicted output is TRUE.

We can also find out the accuracy of the model using the confusion matrix.

Accuracy = (True Positives + True Negatives) / (Total number of classes)

i.e. for the above example:

Accuracy = (100 + 50) / 165 = 0.9090 (90.9% accuracy)

Evaluation:

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future.

To improve the model we might tune the hyper-parameters of the model and try to improve the accuracy and also looking at the confusion matrix to try to increase the number of true positives and true negatives.

Conclusion:

In this blog, we have discussed the workflow a Machine learning project and gives us a basic idea of how a should the problem be tackled.

CHAPTER 2

LITERATURE SURVEY

2.1 ANALYSIS OF THE LITERATURE

Literature survey is the main advance in programming improvement measure. Prior to building up the instrument it is important to decide the time factor, economy and friends strength. When these things are fulfilled, at that point the subsequent stage is to figure out which working framework and language can be utilized for building up the device. When the developers begin assembling the apparatus the software engineers need parcel of outer help. This help can be gotten from senior developers, from book or from sites. The major part of the project development sector considers and fully survey all the required needs for developing the project. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Prior to building the framework the above thought are considered for building up the proposed framework. The significant piece of the undertaking advancement area considers and completely survey all the necessary requirements for building up the venture. For each undertaking Literature survey is the main area in programming improvement measure. Prior to building up the instruments and the related planning it is important to decide and survey the time factor, asset prerequisite, labor, economy, and friends strength. When these things are fulfilled and completely surveyed, at that point the following stage is to decide about the product details in the separate framework, for example, what kind of working framework the venture would require and what are largely the important programming are expected to continue with the subsequent stage like building up the apparatuses, and the related activities. Here we have taken the general surveys of different creators and noted down the fundamental central issues with respect to their work. In this venture literature survey assumes a prevailing part in get assets from different areas and all the connected points that are exceptionally valuable under this segment. The most awesome aspect if this is the manner in which things get all together and encourages us to suite our work according to the current information.

2.2 LITERARY REVIEWS

Towards Distributed Event Detection in Wireless Sensor Networks(IEEE2020)

Norman Dziengel et al. distributed event detection in wireless sensor networks (WSNs) is the process of observing and evaluating an event using multiple sensor nodes without the help of a base station or other means of central coordination and processing. Current approaches to event detection in WSNs transmit raw data to an external entity for evaluation or rely on simplistic pattern recognition schemes. This implies either high communication overhead or low event detection accuracy, especially for complex events. In this paper, we present our currently on-going work on a system for distributed event detection that particularly suits the specific characteristics of WSNs. Adapting traditional pattern recognition algorithms to highly embedded devices, it uses the distributed sampling of sensor nodes to optimize the accuracy of the event detection process. Four different algorithms for distributing, classifying and fusing “fingerprints” of the raw data sampled on each sensor are proposed and quantitatively evaluated in a small-scale experiment.

Detecting and monitoring time-related abnormal events using a wireless sensor network and mobile robot(IEEE, 2019)

YuanYuan Li and Lynne E. Parker et al. presented an anomaly detection system that is able to detect time-related anomalies by using a wireless sensor network and a mobile robot. The sensor network uses an unsupervised fuzzy Adaptive Resonance Theory (ART) neural network to learn and detect intruders in a previously unknown environment. Upon the detection of an intruder, a mobile robot travels to the position where the intruder is detected to investigate by using its camera. The wireless sensor network uses a hierarchical communication/learning structure, where the mobile robot is the root node of the tree. Our fuzzy ART network is based on Kulakov and Davcev’s implementation [8]. However, we enhance their work by extending the fuzzy ART neural network with a Markov model to learn a time series and detect time-related anomalies.

Finally, a mobile robot is employed to verify whether the detected anomalies were caused by intruders.

A Novel Mechanism for Fire Detection in Subway Transportation Systems Based on Wireless Sensor Networks(vol. 2019)

Z.-J. Zhang, J.-S. Fu, H.-P. Chiang, and Y.-M. Huang et al. said fire is a common and disastrous phenomenon in subway transportation systems because of closed environment and large passenger row. Traditional methods detect and forecast re incidents by fusing the data collected by wireless sensor networks and compare the fusion result with a threshold. However, this approach has a significant shortcoming. Fusion of data decreases the amount of useful information since the distribution of data is not considered. In this paper, a novel mechanism based on wireless sensor networks is proposed and we can use the mechanism to detect re in subway transportation systems. e core algorithm in this mechanism is an alternative of data fusion in wireless sensor networks. e mechanism uses the quantity information acquired from the data and also considers the distribution information of the data in order to predict the occurrence of re incidents. e simulation results show that re incidents can be detected fast and eciently.

Forest Fire Smoke Detection Based on Visual Smoke Root and Diffusion (IEEE2019)

The damage caused by forest fire to forestry resources and economy is quite serious. As one of the most important characters of early forest fire, smoke is widely used as a signal of forest fire. In this paper, we propose a novel forest fire smoke detection method based on computer vision and diffusion model. Unlike the video-based methods that usually rely on image characters extraction, we try to find the shape of smoke that is at the generation stage. To combine vision and diffusion model together, the basic concept of smoke root is proposed. In the frame processing stage, none characters of fire smoke are extracted (like texture, color, frequency information etc.), and continuous frames are only used to extract stable points in dynamic areas as the smoke root candidate points. In the diffusion model simulation stage, all smoke root candidate points information is adopted by

the model to generate the simulation smoke. Finally, the match algorithm based on

color, dynamic areas and simulation smoke is implemented to get the final results. In order to reduce the complexity of computation, we ignored the simulation process of the smoke details, such as texture and turbulence, and only retained the contour features in two-dimensional form.

Flutter Analysis Based Video Smoke Detection(IEEE2011)

Based on the flutter characteristic of smoke, a novel video smoke detection algorithm is proposed. Firstly, the motion region is detected from video frames. Then four flutter features of the motion region are extracted over a sliding time window, including the flutter direction and three types of flutter intensities. Finally, the neuro-fuzzy inference system is used to detect smoke, in which fuzzy rules and membership functions are trained according to the valid sample set. Experimental results show that the new algorithm has excellent performance on Receiver Operating Characteristic (ROC) curve which is very important for real application.

AN EARLY FIRE-DETECTION METHOD BASED ON IMAGE PROCESSING. (IEEE2004)

The paper presents an early fire-alarm raising method based on video processing. The basic idea of the proposed of fire-detection is to adopt a RGB (red, green, blue) model based chromatic and disorder measurement for extracting fire-pixels and smoke-pixels. The decision function of fire-pixels is mainly deduced by the intensity and saturation of R component. The extracted fire-pixels will be verified if it is a real fire by both dynamics of growth and disorder, and further smoke. Based on iterative checking on the growing ratio of flames, a fire-alarm is given when the alarm-raising condition is met. Experimental results show that the developed technique can achieve fully automatic surveillance of fire accident with a lower false alarm rate and thus is very attractive for the important military, social security, commercial applications, and so on, at a general cost.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The methods and classifiers used in the existing system are:

- Correlation based analysis and transformation methods
- Data Mining Approach to Predict Forest Fires using Meteorological Data
- SVM classifier
- Circle-based Approximation to Forest Fires with Distributed Wireless Sensor Networks using clustering
- Event Detection in Wireless Sensor Networks for Disaster Management

3.2 PROBLEM STATEMENT:

The main motivation for fire prediction is to provide proper resource allocation and to help in best possible way to firefighters of Fire Management team. The main factors of fire are Meteorological conditions. the climatic information is gotten from nearby sensors which are fused in the closest meteorological stations. Land with a possible high fire risk has many indicators that can be used to measure the forecast by closely evaluating the indications. Every year, fire destroys millions of hectares of land. These fires have burned vast areas and generate more carbon monoxide than total vehicle traffic. Monitoring potential danger areas and early warning of fire can greatly reduce response time, as well as the potential for damage and firefighting costs.

3.3 PROPOSED SYSTEM:

Here In Fire Detection, We Are Using The Haar Cascade Classifier, Which Is Very Popular In Object Detection Through The Image Or Any Other Video Feeds. For Detection We Use Open Cv Which Is A Library Of Python Bindings Designed To Solve Computer Vision Problems. We Use Adaboost Algorithm Which Is A Learning Algorithm Used To Train The Classifier And Selects The Best Subset Of

Features. Cascade Classifier Is Used For The Accuracy Of Identification. It Is Composed of Several Stages Consisting Of A Strong Classifier.

ADVANTAGES:



BLOCK DIAGRAM:

Fig 3.1 Block Diagram

FLOW DIAGRAM:

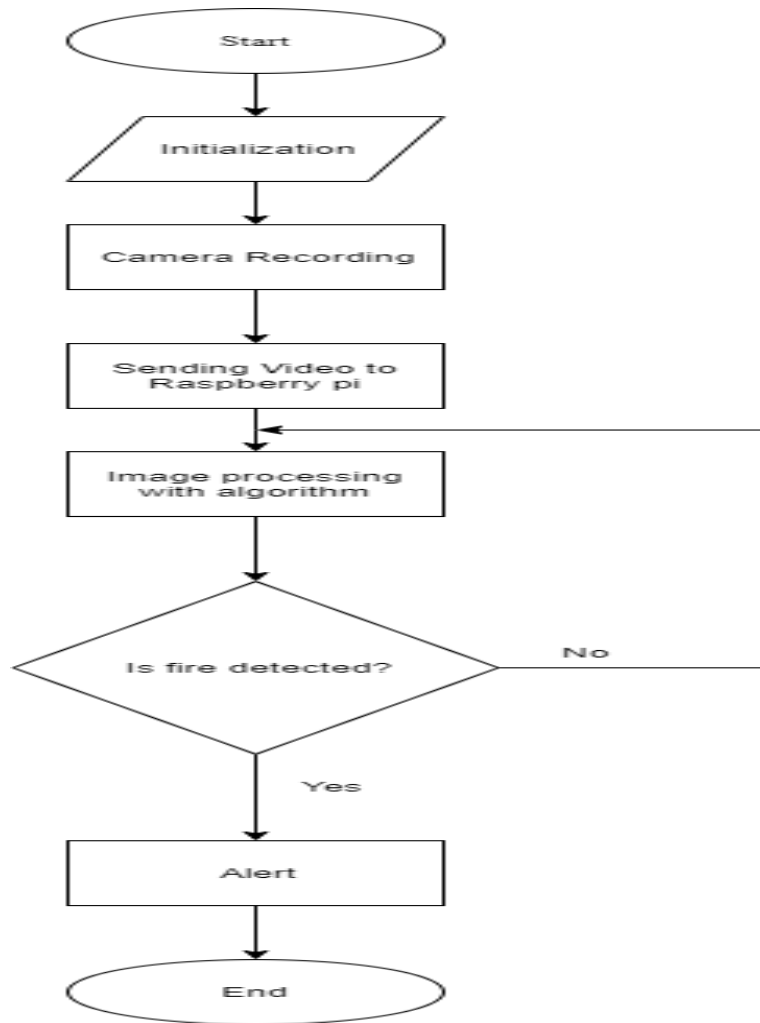


Fig 3.2 Flow diagram

3.4 SYSTEM

DESIGN Modules:

- Detection
- Haar-like Features
- Integral Image
- Training and Learning
- Adaptive Boosting (Ada Boost)
- Cascading

Detection:

The procedure for detecting fire classifies images based on the value of simple features from an image. While detecting the fire, first of all, image is converted into grayscale since it is easy to work with, and it has less data rather than RGB color images

The algorithm outlines a box and searches for the fire in the image,, the box is searching for Haar-like features.

Along with small steps, the box detects the features like edges, brightness level from the image of fire, and then data collected by boxes are put together, which helps to determine where the fire is located

3.5 Haar-like Features:

Haar features are similar to convolutional kernels, which are used to detect the presence of those features in the given image.

The grayscale has some features some of the regions are black, some regions are white, and some region is slightly darker or vice-versa from each other which helps the machine to understand an image.

He Haar features like edge features, line features, center-surround features rectangle is moved around the image.

The color white and black is decided according to the region. Which compares the brightness and extracts features by assigning some values as it goes and inspects through that region in an image.

- The construction of the haar is like that of a confirmed egg and is utilized to decide if these properties are displayed in a given picture.
- Dim has a few dark regions and some have white regions, some are hazier or some are a method for aiding the vehicle feel the picture.
- He Haar has side highlights, line elements, and elements around the focal point of the square shape.

- White and dark are territorially supported. It represents light and disposes of the trademark by allocating a particular worth as it goes through the photo and controlling the region.

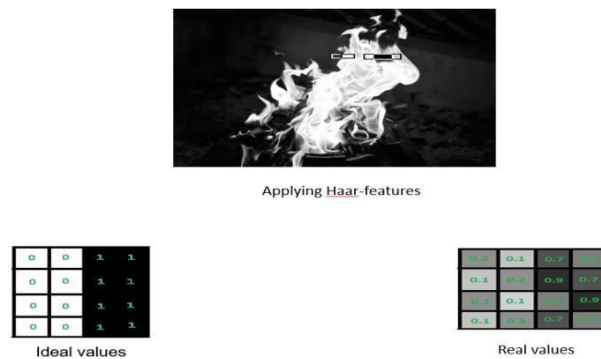


Fig 3.3 Haar Features

Formula:

$$\Delta = \text{dull} - \text{white} = 1 \cdot n \sum I(x) - 1 \cdot n \sum I(x)$$

For positive pictures, $\Delta: 1 - 0 = 1$

For the genuine picture, $\Delta: 0.67 - 0.15 = 0.52$

Integral Image:

The integral image plays an important role. It gives a boost to the calculation, which makes the approach as fast as possible because there are thousands of pixels that need to be calculated.

While applying rectangular haar features the sum of pixels in unshaded rectangles sides are subtracted from the sum of the pixel in the shaded side of rectangles.

Even for a small size images there are lots of features (over 160,000 for a 24x24 image).

Since due to large number of features the algorithm requires iterating overall features, the features must be computed efficiently. So to solve this issue integral image is introduced.

The sum of the pixel in rectangle D can be calculated with reference to four arrays. The value of the image at location 1 is the sum of pixels in a rectangle. Respectively value at position 2 is A+B, value at position 20 3 is A+C, and value at position 4 is A+B+C+D.

Training and Learning:

Extracting features from an image to detection. The machine is now trained to identify the features from the input. So it is fed with lots of information so that it can predict objects from the information.

To train the machine, It needs a lot of pictures of fire, different fireplaces, conditions of fire, also known as a positive image. Concerning that, many negative images were uploaded.

The negative image refers to pictures in which fire is not included in an image.

Negative images are uploaded so it can differentiate between the two classes, which helps to an algorithm which features are more likely to be a fire and which are not expected to be fire.

Adaptive Boosting (Ada Boost):

Adaboost algorithm is a learning algorithm used to train the classifier and selects the best subset of features. The algorithm learns from the data that is given and determine the false positive and true negatives in data.

$$F(x) = a_1f_1(x) + a_2f_2(x) + \dots$$

Here $F(x)$ is the strong classifier, and $a_1f_1(x)$, $a_2f_2(x)$ are weak classifiers where a_1 , a_2 are the weights, and f_1 , f_2 are the features.

So the strong classifier is made up of many weak classifiers since one weak classifier is not good as adding more classifiers makes the algorithm stronger and accurate, which is called an ensemble.

Cascading:

Cascade classifier is used for the accuracy of identification. It is composed of several stages consisting of a strong classifier. Those strong classifiers are passed by So all the features are grouped in several stages where each stage has a certain number of features.

The use of these several stages is used to determine whether the given input sub window has features of fire or not if there are no features of fire, then the given sub window is discarded and fails to go for other stages.

there are some stages stage 1 and stage 2. Usually, the first few stages will contain very less numbers of features.

If the window fails, it is discarded if not apply the second stage of features and continue the process. The window or stage which passes all the features of fire, then it is detected as fire.

3.6 WORKING PRINCIPLE:

The first step is the training of the classifier, as mentioned early. Training for the highly accurate classifier needs lots of time and processing power, so here we only used a little number of pictures.

After training the fire cascade classifier, the captured frame from the webcam is converted into grayscale. The reason for converting frame into grayscale is because the frame captured by webcam is in RGB color.

Since RGB images have three channels of colors, so if the image is converted into grayscale, there will be only one channel, either black or white, which is easy to process.

After the conversion, the fire classifier is used, which will help to find the features and location of images. Parameters like scale factor and min neighbor are passed. These factors are an essential factor in detecting fire. A scale factor is used for creating of scale pyramid because while training the classifier fixed size of the image is trained, so the scale factor will allow rescaling the size of an input frame to detect the fire.

Another parameter min 24 neighbor will determine the quality of an image here
for thesis min neighbor factor

3.7 SYSTEM REQUIREMENTS:

HARDWARE REQUIREMENTS:

System : Pentium i3 Processor.

Hard Disk : 500 GB.

Monitor : 15" LED

Input Devices : Keyboard, Mouse

Ram : 2 GB

SOFTWARE REQUIREMENTS:

Operating system: Windows 7.

Coding Language: python

Tool: Anaconda Navigator

Libraries: OpenCV

CHAPTER 4

Results and Discussion

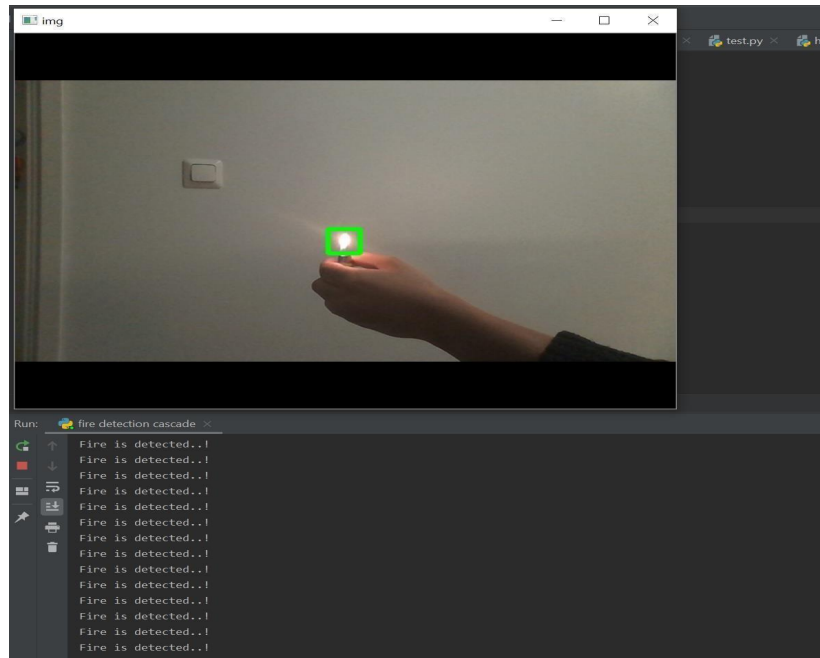


Fig 4.1 Result Screen Shot

Since the test was performed during room lighting, the estimations must be changed by the room light to be effective.

- In case of a fire, a boisterous commotion. The outcomes are likewise displayed as FIRE.
- Assuming that there is no fire, the outcome won't be NO FIRE

Chapter 5

Conclusion and Future Work

5.1 Conclusion:

The project aimed to detect fire with a different approach rather than using an existing system. As technology is getting better and better as to keep it up with the technology and to minimize the limitations also, the new system has created. By using image processing technology for detecting the fire, these limitations can be reduced because in this system camera acts like a human eye, as it detects a fire, the video is captured, and the image is processed using the software alert user. It can be used everywhere eg. Hospital, railway station, forest etc. The designed prototype successfully detects fire. gives the review analysis, designing system, and algorithm, test, and result.

Currently, we not used the systems like a smoke detector and sprinkler water discharge systems, but in future it can be included.

5.2 Limitation & Future scope:

Due to safety reasons, the effectiveness of the system cannot be tested in the full phase, so here for the prototype demo, the lighter is used as the source of fire as it has the same features and characteristics of the fire. There is some error in detection when the fire is far away from the camera because of the size of the flame. The system can be made more accurate and precise by using other machine learning algorithms like YOLO(You Only Look Once), Tensorflow, Keras algorithm, neural network like CNN, RCNN, which are more advanced than the current algorithm.

REFERENCE PAPERS:

- [1] N. Dziengel, G. Wittenburg, and J. Schiller, "Towards distributed event detection in Wireless Sensor Networks," in Adjunct Proc. of 4th IEEEfACM Intl. Conf. on Distributed Computing in Sensor Systems (DCOSSa:AZ'08), Santorini Island, Greece, 2020.
- [2] S. Jarupadung, "Distributed event detection and semantic event processing," in The 6th ACM International Conference on Distributed Event- Based Systems (DEBS 2020)(Doctoral Symposium), 2020.
- [3] Y. Li and L. E. Parker, "Detecting and monitoring time-related abnormal events using a Wireless Sensor Networks and mobile robot," in Intelligent Robots and Systems, 2019. IROS 2019. IEEEfRSJ International Conference on. IEEE, 2019.
- [4] P. Radivojac, U. Korad, K. M. Sivalingam, and Z. Obradovic, "Learning from class-imbalanced data in Wireless Sensor Networks," in Vehicular Technology Conference, 2020. VTC 2020-Fall. 2020 IEEE 58th, vol. 5, IEEE, 2020.
- [5] Z.-J. Zhang, J.-S. Fu, H.-P. Chiang, and Y.-M. Huang, "A novel mechanism for fire detection in subway transportation systems based on Wireless Sensor Networks," International Journal of Distributed Sensor Networks, vol. 2019, 2019.
- [6] Boroujeni, N. S. (2019). Monocular vision system for unmanned aerial vehicles. doi:10.22215/etd/2013-07242.
- [7] Akyurek, S., Yılmaz, M.A., and Taskiran, M. (2012)., "İnsansız Hava Araçları(Muharebe Alanında ve Terörle
- [8] Mücadelede Devrimsel Dönüşüm", technical report No: 53, Ankara, TR.
- [9] Chen, T., Wu, P., & Chiou, Y. (2014). An early fire-detection method based on image processing. 2004
- [10] International Conference on Image Processing (ICIP04), Singapore.
- [11] Gao, Y., & Cheng, P. (2019). Forest Fire Smoke Detection Based on Visual Smoke Root and Diffusion
- [12] Model.Fire Technology. doi:10.1007/s10694-019-00831-x.
- [13] Gao, Y., & Cheng, P. (2019). Forest Fire Smoke Detection Based on Visual

Smoke Root and Diffusion

[14] Model.Fire Technology. doi:10.1007/s10694-019-00831-x.

- [15]Wang, T., Liu, Y., & Xie, Z. (2011). Flutter Analysis Based Video Smoke Detection. Journal of Electronics & Information Technology,33(5), 1024-1029.
doi:10.3724/sp.j.1146.2010.00912.
- [17]Lin et. al, Smoke detection in video sequences based on dynamic texture using volume local binary patterns. (2017). KSII Transactions on Internet and Information Systems, (11).
doi:10.3837/tiis.2017.11.019.
- [19]Lin et. al, Wang, Z., Wang, Z., Zhang, H., & Guo, X. (2017). A Novel Fire Detection Approach Based on CNN-SVM Using Tensorflow. Intelligent Computing Methodologies Lecture Notes in Computer Science,682-693.
- [21]Shaqura, M., & Shamma, J. S. (2017). An Automated Quadcopter CAD based Design and Modeling Platform using Solidworks API and Smart Dynamic Assembly. Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics.
doi:10.5220/0006438601220131.
- [24] O. T. Cetinkaya, S. Sandal, E. Bostancı, M. S. Güzel, M. Osmanoğlu and N. Kanwal, "A Fuzzy Rule Based Visual Human Tracking System for Drones," 2019 4th International Conference on Computer Science and Engineering (UBMK), Samsun, Turkey, 2019, pp. 1-6, doi: 10.1109/UBMK.2019.8907104.

APPENDIX

A.SAMPLE CODING

```
import cv2
import numpy as np
import math
import pygame
pygame.init()

pygame.mixer.music.load("alarm.wav")
windowName = "OpenCV Video Player"
cv2.namedWindow(windowName)

cap = cv2.VideoCapture(0)

ret, lframe = cap.read()
LI, A, B =
cv2.split(lframe)
SI = np.zeros(np.shape(LI))
print(SI)
NOI = 0
NOc = 0
CGO = 0
x = 1
dt = 0
nop = 0
count = 0
mc = 0
while (ret):
    ret, frame = cap.read()
```

if ret:

LI, A, B = cv2.split(lframe)

```

ret, frame = cap.read()
cframe = frame
cv2.imshow("FIRE", frame)
b, g, r = cv2.split(frame)
rt = 230

gb = cv2.compare(g, b, cv2.CMP_GT)
rg = cv2.compare(r, g, cv2.CMP_GT)
rrt = cv2.compare(r, rt, cv2.CMP_GT)
rgb = cv2.bitwise_and(rg, gb)
im = cv2.bitwise_and(rgb, rrt)
# cv2.imshow("RGB",im)

t = 5
p = 1
k = cv2.getStructuringElement(cv2.MORPH_CROSS, ((t, t)))
# print(k)
dil = cv2.dilate(im, k, iterations=p)
er = cv2.erode(im, k, iterations=p)
fin = cv2.bitwise_and(er, dil)
# cv2.imshow("FIRE",frame)
# cv2.imshow("segmented fire",cv2.bitwise_and(frame,frame,mask=fin))

img_ycrCb = cv2.cvtColor(frame, cv2.COLOR_BGR2YCR_CB)
Y, Cr, Cb = cv2.split(img_ycrCb)
Ym = np.mean(Y)
Crm = np.mean(Cr)
Cbm = np.mean(Cb)

I1 = cv2.compare(Y, Ym, cv2.CMP_GT)
I2 = cv2.compare(Cb, Cbm, cv2.CMP_LT)
I3 = cv2.compare(Cr, Crm, cv2.CMP_GT)
I12 = cv2.bitwise_and(I1, I2)
I23 = cv2.bitwise_and(I2, I3)

```

```
I123 = cv2.bitwise_and(I12, I23)
```

```

# cv2.imshow("I123",I123)
cbcrdiff = cv2.absdiff(Cb, Cr)
asd = cv2.compare(cbcrdiff, 40, cv2.CMP_GT)
# cv2.imshow("asd", asd)

img_cie = cv2.cvtColor(frame, cv2.COLOR_BGR2Lab)
L, a, b = cv2.split(img_cie)
Lm = np.mean(L)
am = np.mean(a)
bm = np.mean(b)
R1 = cv2.compare(L, Lm, cv2.CMP_GT)
R2 = cv2.compare(a, am, cv2.CMP_GT)
R3 = cv2.compare(b, bm, cv2.CMP_GT)
R4 = cv2.compare(b, a, cv2.CMP_GT)
R12 = cv2.bitwise_and(R1, R2)
R34 = cv2.bitwise_and(R3, R4)
R14 = cv2.bitwise_and(R1, R4)
# cv2.imshow("R14",R14)

kl = cv2.getStructuringElement(cv2.MORPH_CROSS, ((t, t)))
e_R14 = cv2.erode(R14, kl, iterations=p)
d_R14 = cv2.dilate(R14, kl, iterations=p)
bin_cie = cv2.bitwise_and(e_R14, d_R14)
# cv2.imshow("cie_res",cv2.bitwise_and(frame,frame,mask=bin_cie))

im = cv2.bitwise_and(bin_cie, im)
# cv2.imshow("RGBCIE",im)

cframe = cv2.cvtColor(frame, cv2.COLOR_BGR2Lab)
Lc, A, B = cv2.split(cframe)

Dlfd = cv2.absdiff(Lc, LI)
# cv2.imshow("difd",Dlfd)

```

```
u_Dlfd = np.mean(Dlfd)
```

```
sd_Dlfd = np.std(Dlfd)
```

```
if u_Dlfd + sd_Dlfd >= 10:
```

```
    Tfd = u_Dlfd + sd_Dlfd
```

```
else:
```

```
    Tfd = 10
```

```
_, FD = cv2.threshold(Dlfd, Tfd, 255, cv2.THRESH_BINARY)
```

```
# cv2.imshow("difd2",FD)
```

```
# print(FD)
```

```
cntr = cv2.compare(FD, 0, cv2.CMP_EQ) / 255
```

```
SI = np.add(cntr, SI)
```

```
# print(SI)
```

```
MPM = cv2.bitwise_and(FD, im)
```

```
MPM = cv2.bitwise_and(MPM, asd)
```

```
MPM = cv2.erode(MPM, kl, iterations=p)
```

```
# cv2.imshow("MPM", MPM)
```

```
CF = cv2.bitwise_and(im, MPM)
```

```
# cv2.imshow("candi", CF)
```

```
retval, labels, stats, centroids = cv2.connectedComponentsWithStats(CF)
```

```
for i in range(retval):
```

```
    if i > 0:
```

```
        if stats[i][4] > 2:
```

```
            NOc = NOc + stats[i][4]
```

```
            # print(NOc)
```

```
        if NOc > NOI:
```

```
            CGO = CGO + 1
```

```
        NOI = NOc
```

```
        NOc = 0
```

```
        fps = 24
```

```
if x % fps == 0:
```

```

    dt = CGO * 1.0 / fps
    print(dt)
    CGO = 0

    x = 1
    x = x + 1
    # print(dt)
    lframe = cframe
    if dt > 0.3:
        count = count + 1
        if count > 10:
            mc = mc + 1
            if mc % 15 == 1:
                print("fire")
                try:
                    pygame.mixer.music.play()
                except:
                    pass
            count = 0
        if dt < 0.2:
            print("not fire")
    k = cv2.waitKey(1)
    if k == ord('q'):
        break

cv2.destroyAllWindows()
cap.release()

```

B.PLAGARISM REPORT

2 COMPUTER VISION-BASED EARLY FIRE DETECTION USING MACHINE LEARNING AND OPEN CV

Potu Lakshmi Praneeth Revanth Kumar *

Student

poturecvanth@gmail.com

Peta Manish Kumar Reddy *

Student

petaamanishkumarreddy@gmail.com

Dr.S.Raja Shree M.E.,Ph.D

Professor

rajashree.cse@sathyabama.ac.in

Abstract:

The undertaking intends to recognize fires utilizing picture handling innovation to caution individuals of early fire location. Similarly as with most fire quenchers, there are various sensors that have their own constraints, intended to be delicate to fire and smoke in restricted spaces. Projects are expected to lessen lines and stay up with new advancements. The venture involves pycharm IDE and webcam joining as apparatuses. A webcam is viewed as an info source and takes recordings from a close by source and enters them into an investigation framework. All code is written in great python language utilizing an open picture altering library. Hypothetical areas center around project-based calculations for PC direction, AI, picture handling, shading delivering, and fire identification. Projects permit us to all the more likely comprehend and understand things and PCs, and use them in an alternate way.

Keywords: OpenCV, Python, Fire Detection, ADA Boost, Haar.

I. INTRODUCTION:

Vision-based PCs with picture handling can be valuable in a manner that is regularly unusable. The start calculation utilizes light signals like light, shading, surface, wavering, and vibration to recognize them from different boosts. There are different techniques for identifying fire, for example, infrared sensors, heat meters, smoke alarms, fire indicators, and smoke alarms. This technique isn't dependably solid since it doesn't recognize the actual fire, however it distinguishes at least one wellsprings of fire, like smoke, heat, infrared, bright, and gas radiation, and consequently gives results, a great deal of commotion. With the assistance of PC vision and imaging innovation, pictures can give dependable data and accomplish preferable outcomes over traditional frameworks.

II. LITERATURE REVIEW:

[1] A Video-Based Fire Detection Using DeepLearning Models

Abstract: Fire is one of the most perilous types of life and property. In this paper, we offer a top to bottom investigation of fire-based preparing utilizing video that impersonates the course of fire. The proposed technique utilizes Convolutional Neural Network Rapid Speed (R-CNN) to portray fire (SRoFs) and non-fire parts. The attributes of the cases, which comprise of a progression of successions gathered by long haul memory (LSTM), are then summed up to decide if a fire broke out in a brief timeframe. It is feasible to consolidate transient progressive choices and get a larger part vote to settle on a ultimate conclusion over the long haul. It additionally considers the parts of fire and smoke, clarifies their momentary changes, and clarifies the idea of the fire and the last finish of the fire. Studies have shown that drawn out video-based techniques to diminish bogus and misdirecting data further develop fire location contrasted with video-based or momentary video-based strategies.

2 [2] Computer Vision-based Early Fire Detection Using Enhanced ChromaticSegmentation and Optical Flow Analysis Technique

Abstract: Late advances in imaging innovation have prompted the investigation of PC based fire recognition frameworks. This report traces a staggered fire locator, including investigating chromatic data, changing settings, and contrasting flames. Most importantly, the chromatic data-based fire pixel thickness work utilizes a great deal of sound among present day shading location frameworks to dispose of intriguing parts. The erased pixels are then affirmed by continuous energy examination. At long last, the violent speed is checked optically by the insightful calculation to affirm the presence of fire.

[3]. Machine vision-based real-time early flame and smoke detection

Abstract: This article depicts a better approach to utilize an alarm framework and to decide when smoke can be infused into an early observing framework. The programmed observing framework utilizes chronicled calculations to record the potential places of fire and smoke in the video and afterward investigate the articles, space and transient qualities of the fire and smoke in the video observation. The greater part of the correlations are made by looking at the histograms of shading and smoke histograms utilizing the HSI shading space. The most extreme thickness of the not entirely settled by the computation of how much fire and smoke contrasted with the border and the region. Spread of figures. The thickness and balance are observed by an obscure framework to give fire and smoke destinations. The transient thickening is portrayed by the arrangement of a light aspect and the partition of non-light items from light and smoke. The specific season of fire and smoke is currently needed at the most elevated level utilizing the continuously adaptive mean shift (CAMSHIFT) observing calculation. Test results performed at various occasions demonstrate that the normal technique can precisely recognize fire and smoke.

[4] Machine Learning Based Early Fire Detection System using a Low-Cost Drone

Abstract: This paper gives a better approach to prepare machines to recognize early and fruitful woodlands and flames. Accordingly, we plan to make a new and clear vision for ranger service and ranger service. To do this, form a robot. Microcontrollers in the framework are prepared and created in cutting edge research, enabling the robot to recognize smoke, the primary indication of fire. The most widely recognized issue with fire calculations is bogus detailing and carelessness. Checking the outcomes found in the photos and the extra strides of actually looking freely increment the dependability of the framework, yet will likewise reinforce the genuine outcomes. Because of the remarkable capacities of automated aeronautical vehicles, it is feasible to screen data precisely and reliably in any capacity. The working of the framework is constrained by impersonation and actual testing.

III. Existing system:

The examination depends on investigation and adjustment. Firefighting and a circulated pipeline utilizing SVM arranging and information assortment techniques to distinguish woods and woodland fires utilizing climate data. Discover what's going on in the catastrophe the executive organization

3.1 Problem statement:

- ❖ The fundamental justification behind foreseeing fires is to give the best assets and to help the fire detachment however much as could be expected.
- ❖ The air is a major fire. Climate data is accessible on the sensor close the closest station.
- ❖ There are various indications of fire hazard evaluation and estimation.
- ❖ A great many hectares of arable land every year. This fire consumes in a bigger region and delivers more monoxide than some other vehicle.
- ❖ Observing of possibly risky regions and early admonition of fire dangers incredibly diminishes salvage time, just as harm and fire doxing costs.

3.2 Proposed system:

- ❖ Here we utilize the classes of Haar Cascade to decide the fire, which is most popular for looking through photographs or different recordings.
- ❖ Menu - open cv - python
- ❖ Things like Haar
- ❖ Full Image
- ❖ Preparing and learning
- ❖ Variation to environmental change
- ❖ To mimic

5.1 Software required:

- system: Windows 10.
- Language: python
- Tool: Anaconda Navigator
- Libraries: OpenCV

VI. Results:



- Since the test was performed during room lighting, the estimations must be changed by the room light to be effective.
- In case of a fire, a boisterous commotion. The outcomes are likewise displayed as fire.
- Assuming that there is no fire, the outcome won't be FIRE.

VII. Conclusion:

The venture was pointed toward recognizing fires in an alternate manner from utilizing existing frameworks. Innovation is progressing, and new advancements and difficulties are being presented. These hindrances can be alleviated with the assistance of video handling innovation to distinguish fires, in light of the fact that in this framework, the camera goes about as a natural eye, detecting fire, catching pictures, and handling pictures through video clients. It very well may be utilized anyplace, for instance. Emergency clinics, trains, backwoods, and that's only the tip of the iceberg. The plan plainly shows the fire. Gives general examination, plan framework, calculations, tests and arrangements. Frameworks, for example, floppy plates and spinkers are not at present being used, but rather can be introduced later on.

VIII. References:

- [1] N. Dziengel, G. Wittenburg, and J. Schiller, "Towards distributed event detection in Wireless Sensor Networks," in Adjunct Proc. of 4th IEEEACM Intl. Conf. on Distributed Computing in Sensor Systems (DCOSS&AZ08), Santorini Island, Greece, 2020.
- [2] S. Jarupadung, "Distributed event detection and semantic event processing," in The 6th ACM International Conference on Distributed Event- Based Systems (DEBS 2020)(Doctoral Symposium), 2020.
- [3] Y. Li and L. E. Parker, "Detecting and monitoring time-related abnormal events using a Wireless Sensor Networks and mobile robot," in Intelligent Robots and Systems, 2019. IROS 2019. IEEE/RSJ International Conference on. IEEE, 2019.
- [4] P. Radivojac, U. Korad, K. M. Sivalingam, and Z. Obradovic, "Learning from class-imbalanced data in Wireless Sensor Networks," in Vehicular Technology Conference, 2020. VTC 2020-Fall. 2020 IEEE 58th, vol. 5, IEEE, 2020.
- [6] Z.-J. Zhang, J.-S. Fu, H.-P. Chiang, and Y.-M. Huang, "A novel mechanism for fire detection in subway transportation systems based on Wireless Sensor Networks," International Journal of Distributed Sensor Networks, vol. 2019, 2019.

4.7 Adaptive Boosting (Ada Boost)

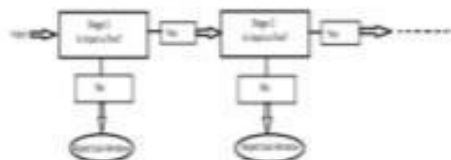
- Adaboost calculation is a calculation practice that is utilized to prepare classes and select the most awesome aspect of the fig. The calculation gains from the data gave and distinguishes the advantages and disadvantages.

$$F(x) = a_1 f_1(x) + a_2 f_2(x) + \dots$$

- Here $F(x)$ is a solid classifier, $a_1 f_1(x)$, $a_2 f_2(x)$ is a frail classifier, a_1 and a_2 are weighted, while f_1 and f_2 are trademark.
- Adding one powerless classification isn't great on the grounds that the calculation is solid and valid, so the huge classifications are comprised of numerous frail and group individuals.

4.8 Cascading

- Course records are utilized to decide precision. It comprises of a few classifications comprising of a solid classifier. These strong powers move. Therefore, every one of the attributes are gathered into many parts, and every classification has its own one-of-a-kind number.
- Utilizing a bit by bit number can be utilized to decide whether there are indications of fire in the sub-window, and in the event that there is no indication of fire, the sub-window can't be taken out and no further advances can be taken.
- There are stages 1 and stage 2. Ordinarily the primary stages have not very many highlights.
- Assuming that the window falls flat, erase assuming the second step the capacity isn't utilized, then, at that point, proceed with the cycle. A window or stepping stool conveys every one of the objects of fire, and afterward tells the fire.



4.9 Working principle:

- The initial step is to prepare the classes, as referenced previously. Top notch preparing requires a ton of time and work to process, so here we are utilizing few pictures.
- Subsequent to preparing the fire course, the edge eliminated from the web camera will become dark. The justification for changing the dim casing is on the grounds that the edge caught by the webcam is RGB in shading.
- Since RGB pictures have three shading channels, changing the picture over to blue will bring about just one channel, dark or white, and simple to process.
- After change, a fire quencher is utilized, which assists with getting the qualities and picture of the picture. Passing estimations like estimations by a little neighbor.
- These focuses are significant in getting fire. The scale is utilized to make the pyramid scale, in light of the fact that the scale is utilized to prepare the size of the picture, so the scale assists you with changing the size of the contribution to decide the fire.
- Min 24 Neighbor Other measures here decide picture quality per min Neighbor System

V. Hardware required:

System : Pentium i3 Processor .

HDD : 500 GB.

Screen : 15" LED

Input Devices : Keyboard, Mouse

Random Access Memory : 2 GB

- Dim has a few dark regions and some have white regions, some are hazier or some are a method for aiding the vehicle feel the picture.
- He Haar has side highlights, line elements, and elements around the focal point of the square shape.
- White and dark are territorially supported. It represents light and disposes of the trademark by allocating a particular worth as it goes through the photo and controlling the region.



4.4 Formula:

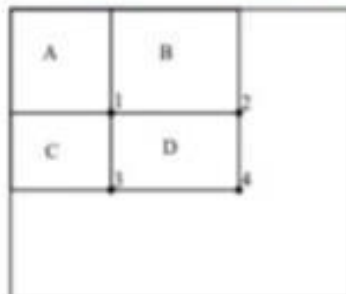
$$\Delta = \text{dall} - \text{white} = 1 \pi \sum I(x) \text{ndark} - 1 \pi I(x)$$

For positive pictures, $\Delta: 1 - 0 = 1$

For the genuine picture, $\Delta: 0.67 - 0.15 = 0.52$

4.5 Integral Image

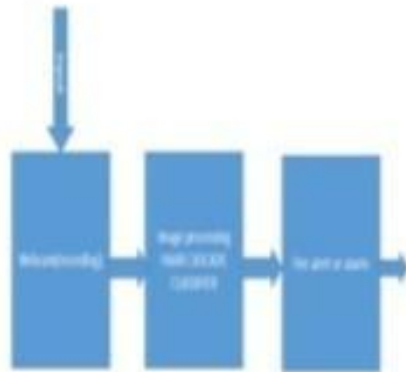
- The fundamental person assumes a key part. It gives registering power and velocities up the cycle as it can count huge number of pixels.
- Deducting the amount of the pixels of the non-rectangular sides takes away the quantity of pixels in the shadow of the square shape.
- Indeed, even little pictures have many capacities (north of 160,000 on 24x24 pictures).
- Calculations require countless things to acquire general things, so the properties should be determined accurately. Thusly, to tackle this issue, we present an essential picture.
- The amount of the pixels of a square shape D can be determined utilizing four lines.
- The worth of the picture in place 1 is the amount of the pixels in the square shape. The worth of position 2 is $A + B$, the worth of position 3 is $A + C$, and the worth of position 4 is $A + B + C + D$.



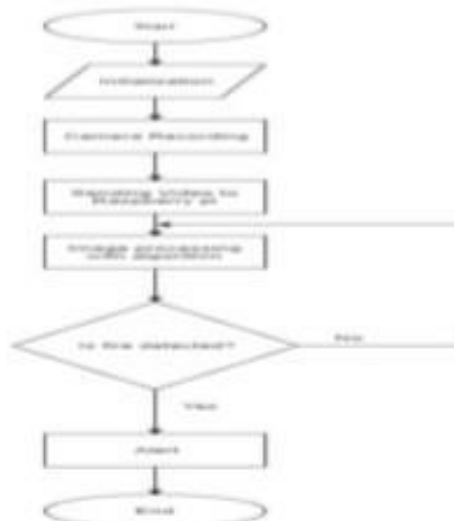
4.6 Training and Learning

- Get things good and gone to discover. The machine is currently prepared to decide the attributes of the info. Accordingly, it is taken care of a ton of data to foresee things from the source.
- Numerous firecrackers, different fires, firecrackers, thus called great pictures are utilized in vehicle preparing. Numerous awful photographs have been posted on this point.
- Awful pictures are pictures that are not ablaze.
- In planning, it is feasible to isolate the two classifications, which assists the calculation with figuring out which properties can be scorched and which can't be singed.

3.3 Block diagram:



3.4 Flow chart:



IV. Methodology and Algorithm

- Haar Cascade Sorting - Understand objects utilizing pictures or different recordings.
- AI calculation in view of AI utilizing Haar-like articles coordinated with a course classifier.
- Make an exemplary course and take recordings from the web. The video was cut into pieces.
- A decent picture is one that consumes in a fire, and an awful picture is a foundation picture that portrays those photos.

4.2 Fire detection:

- The fire location strategy decides the picture from the picture as indicated by the worth of basic items. On account of fire discovery, as a matter of first importance, the picture is more beautiful on the grounds that it is simpler to work with and has less data than RGB tone.
- The calculation shows a crate and a fire search in the picture, which is a case that contains a quest for objects like Haar.
- In little advances, the side of the case gets a great deal of light from the fire picture, and afterward the data gathered in the crate is joined to assist with deciding the area of the fire.

4.3 Haar-like Features

- The construction of the haar is like that of a confirmed egg and is utilized to decide if these properties are displayed in a given picture.

RE-2022-9423-plag-report

ORIGINALITY REPORT

3%

SIMILARITY INDEX

0%

INTERNET SOURCES

1%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Metropolia Ammattikorkeakoulu

Student Paper

2%

2

Cuijuan Wang. "Real-Time Analysis of Athletes' Physical Condition in Training Based on Video Monitoring Technology of Optical Flow Equation", Advances in Mathematical Physics, 2021

Publication

1%

Exclude quotes On

Exclude matches Off

Exclude bibliography On