FIRE DETECTION SYSTEM USING IMAGE PROCESSING

Submitted in partial fulfillment of the requirements
of the degree of

Bachelor in Engineering

By

SARKAWAS NAGMA MEHABOOBSAB 12ET07
AZAD KULSUM ALAMGEER 12ET02
MOKASHI MUZNA NOOR AHMED 12ET03
ANSARI MOHD SALIM BADRUDDIN 12ET13

Supervisor (s):
Asst. Prof. ALTAF BALSING

Department of Electronics and Telecommunication Engineering
Anjuman-I-Islam's Kalsekar Technical Campus

2015-2016

Project Report Approval for B.E

This project report entitled ***Fire Detection using Image processing*** by ***Ansari Mohd Salim Badruddin, Sarkawas Nagma Mehaboobsab, Mokashi Muzna Noor Ahmed and Azad kulsum Alamgeer*** is approved for the degree of *Bachelor in Engineering*.

Examiners:

1.________________________________.

2.________________________________.

Supervisor(s):

1.___________________________________

Asst. Prof. ALTAF BALSING

H.O.D(EXTC):

____

Asst. Prof. MUJIB A. TAMBOLI

Date:

Place:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

1.______________________________
SARKAWAS NAGMA MEHABOOBSAB
12ET07

2.______________________________
AZAD KULSUM ALAMGEER
12ET02

3.______________________________
MOKASHI MUZNA NOOR AHMED
12ET03

4.______________________________
ANSARI MOHD SALIM BADRUDDIN
12ET13

Date:

Place:

Acknowledgement

Before we get into thick of things we would like to add a few heartfelt words for the people who are a part of our team as they have given unending contribution right from the start of construction of the report.

Apart from the team we are highly grateful to **Dr. ABDUL RAZZAK HONNUTAGI**, Director and **Asst**.**Prof. MUJIB TAMBOLI**, Head of EXTC Department for providing the facilities and conductive environment. We would like to extend our sincere thanks to our project in charge **Asst. Prof. ALTAF BALSING** for providing us the required technical guidance in the process of preparing this report.

We acknowledge with deep sense of gratitude towards the encouragement in the form of substantial assistance provided by **Asst. Prof. SIRAJ PATHAN** and every member of our team and our management.

Abstract

Fire detection is the main objective of this project besides surveillance. The aim of the project is to early detection of fire apart from preventive measures to reduce the losses due to hazardous fire. The project mainly is based on image processing and arduino serial communication. In this project, at the user end, the fire images will be feeded in the form of video frames.

These images will be further processed by using the software, MATLAB. The proposed system uses RGB and YCbCr color space. The advantage of using YCbCr color space is that it can separate the luminance from the chrominance more effectively than RGB color space. along with this smoke, motion, area detection is also performed using its color characteristics. The proposed system consist of hardware such as arduino, DHT 11 to monitor the Humidity and Temperature.

There is a camera for the surveillance. This camera will give a real-time video output to the user on the laptop or computer via a small GUI-graphic user interface which is to be built in MATLAB. Thus the fire will be detected using this model.

This project can also be served for security and surveillance applications.

List of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 1

Introduction

Fires represent a constant threat to ecological systems, infrastructure and human lives. Past has witnessed multiple instances of fires. With the faster and faster urbanization process, more and more high-rise buildings appear around us. This also can make the frequency of fire increase and bring great losses to people's lives and property. In areas where fire would pose an unreasonable threat to property, human life or important biological communities, efforts should be made to reduce dangers of fire. As the damage caused by fires is so tremendous that the early fire detection is becoming more and more important. Recently, some fire detectors have been used in many places, they used the smoke, temperature and photosensitive characteristics to detect fires. But they are too worse to meet the needs in a large space, harsh environment or the outdoor environment etc.

Traditional fire protection methods use mechanical devices or humans to monitor the surroundings. The most frequently used fire smoke detection techniques are usually based on particle sampling, temperature sampling, and air transparency testing. An alarm is not raised unless the particles reach the sensors and activate them. Some of the methods are mentioned below:

A. Fire Watch Tower

In watch towers human are made to observe the location throughout. If any fire occurs he reports it. However, accurate human observation may be limited by operator fatigue, time of day, time of year, and geographic location.

B. Wireless Sensor Networks

In a wireless sensor-based fire detection system, coverage of large areas in forest is impractical due to the requirement of regular distribution of sensors in close proximity and also battery charge is a big challenge.

C. Satellite and Aerial

Monitoring Satellites based system can monitor a large area, but the resolution of satellite imagery is low. A fire is detected when it has grown quite a lot, so real time detection cannot be provided. Moreover, these systems are very expensive. Weather condition (e.g. clouds) will seriously decrease the accuracy of satellite-based forest fire detection as the limitations led by the long scanning period and low resolution of satellites.

The motivation for an image processing based approach is due to rapid growth of the electronics. Fire detection systems are one of the most important components in surveillance systems used to monitor buildings and environment as part of an early warning mechanism that reports preferably the start of fire. Currently, almost all fire detection systems use built-in sensors that primarily depend on the reliability and the positional distribution of the sensors. The sensors should be distributed densely for a high precision fire detector system. In a sensor-based fire detection system, coverage of large areas in outdoor applications is impractical due to the requirement of regular distribution of sensors in close proximity. Due to the rapid developments in digital camera technology and video processing techniques, there is a big trend to replace conventional fire detection techniques with computer vision-based systems. In our project different characteristic parameters of fire i.e Color, smoke, Area and motion using image processing in MATLAB are analyzed. Along with this monitoring of temperature and Humidity of fire is done for more precision.

In order to create a color model for fire and smoke, we have analyzed the images which consist of fire. YCbCr color space is chosen intentionally because of its ability to separate illumination information from chrominance more effectively than the other color spaces. The rules defined for RGB color space in order to detect possible fire-pixel candidates can be transformed into YCbCr color space and analysis can be performed.

Color alone is not enough to identify fire. There are many things that share the same color as things that are not fire, such as a desert, sun, red leaves and other objects. The key to distinguishing between the fire and the fire colored objects is the nature of their motion.

Motion detection is used to detect any occurrence of movement in a video. It is done by analyzing difference in images of video frames. There are three main parts in moving pixel detection: frame/background subtraction, background registration, and moving pixel detection Similar to the fire detection. We are also modeling smoke pixels. The smoke pixels do not show chrominance characteristics like fire pixels. At the beginning, when the temperature of the smoke is low, it is expected that the smoke will show color from the range of white-bluish to white. Toward the start of the fire, the smoke's temperature increases and it gets color from the range of black-grayish to black. Area detection method is used to detect dispersion of fire pixel area in the sequential frames.

DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. As the fire increases humidity decreases and temperature increases, for this threshold is set. So after analyzing all above parameters fire will be detected and it will give pop-up on the screen of a fire image as well as buzzer as an indication for person present at the location to take a quick action to suppress fire and to avoid losses of human lives and their valuable properties.

Chapter 2

Review of literature

Chapter 2

Review Of Literature

1. T. Celik and Hasan Demirel et al. further enhance system that uses a statistical color model with Fuzzy logic for fire pixel classification. The proposed system develop two models; one based on luminance and second based on chrominance. Fuzzy logic uses the YCbCr color space for the separation of luminance from chrominance instead of using color spaces such as RGB. Existing historic rules are replaced with the Fuzzy logic to make the classification more robust and effective. This model achieves up to 99.00% correct fire detection rate with a 9.50% false alarm rate.

2. R. Gonzalez-Gonzalez et al. proposed a method to detect fire by smoke detection based on wavelet. In this smoke detection method, image processing on video signals is proposed. The SWT transform is used for the area detection of ROI's. This method comprises of three steps.
In the first step, preprocessing is performed and the image is resized and transformed to grayscale image. Finally indexed the image using indexation. The second step involves high frequencies of an image is eliminated using SWT and reconstruct the image by inverse SWT. In order to group the intensity colors that are closed to each other is the main purpose of image indexation.

Histogram analysis is used to determine the indexation levels. After that compare the image with a non-smoke frame and selecting those pixels that are change from one scene to another. The final stage consists of smoke verification algorithm in order to determine whether ROI is increasing its area and to reduce the generation of false alarm. These three steps are combined together to form the final result.

3. Hidenori Maruta et al. proposed another method for smoke detection based on support vector machine. In this approach robust and novel smoke detection method is proposed using support vector machine. Firstly preprocessing is performed by extracting moving objects of images. The preprocessing consists of five steps: image subtraction and accumulation, image binarization, morphological operation, extraction of Feret's regions and creation of the image mask. Image subtraction is used to extract regions of moving object. In order to eliminate noise like regions binarization and morphological operations are used. The position and approximated shape of the object is obtained by identifying Feret's diameter is called feret's region.

After preprocessing, perform texture analysis and extract the texture features. These texture features become the component of feature vector. Feature vector is applied as the input vector and support vector machine is used to classify smoke or not. Smoke detection method involves three steps: analyzing texture features, discrimination of Feret's region using support vector machine and time accumulation.

In order to extract feature vectors of the image, texture analysis is performed in this method. Support vector machine is used to classify smoke or non- smoke from the extracted image. The main advantage of this method is that more accurate extraction of smoke areas in image can be obtained using SVM.

4. Y. Habiboglu et al. proposed another method that uses covariance descriptors for fire detection. In this method, color, spatial and domain information are combined by using covariance descriptors for each spatio-temporal block. The blocks are generated by dividing the flame colored region into 3D regions. This method used a covariance matrix for the detection of flames. Background subtraction method is not used in this approach.

To detect fire, divides the video into spatio temporal blocks and covariance features are computed from these blocks. Using an SVM classifier, the flame colored region are classified by using the spatial and temporal characteristics. These classified flame colored regions are tested using video data that contain flames and flame colored objects. For the classification of pixel colors chromatic color model [4] is used and analyzed fire colored pixels. Object detection and texture classification [5] are performed by applied covariance descriptors [5].

In order to define spatio-temporal blocks temporal derivatives are calculated along with spatial parameters. Spatial temporal blocks can be defined using covariance matrices. Then compute the covariance values of the pixel property vectors in spatio-temporal blocks.
For classification, support vector machine is trained. According to the number positively classified video blocks and their positions, confidence value is determined for fire detection. This method is computationally efficient. Covariance approach is well suited for detection of flames.

Drawback of this method is that, it is well suited when the fire is clearly visible. If the fire is far away from the camera and covered with dense smoke, this method performs poorly.

5. Mehdi Torabnezhad et al. proposed another method that used image fusion technique to detect smoke. In this method, combine visual and thermal information to improve the rate of fire detection. The invisibility of smoke in LWIR image can distinguish smoke from smoke like objects. Infrared images do not detect smoke in the images but can detect smoke like object. By combining visible and IR images smoke can be distinguished. Based on characteristics of visual and thermal smoke images a potential smoke mask is created. In-order to reduce false alarms, PSM is further analyzed by disorder measurements and energy calculations. For the detection of short range smoke visible and IR image fusion algorithm is used. Scope of this paper is to detect the smoke as an indicator of fire. Here visible and infrared images are combined together to distinguish smoke from smoke like objects. Earlier approach that uses sensor or visible images only gives false alarm.

Visible images capture both smoke and smoke like objects. Infrared images do not capture smoke. Integrating these images give correct information about smoke. Objective of this paper is to save people, forest from the fire. By this method generation of false alarm can be reduced to a great extent. The proposed algorithm consists of two phases. In the first phase combine visual and thermal information of the smoke and potential smoke mask is generated. PSM is again analyzed to differentiate true and false alarms. This method is very efficient and detects smoke successfully. Improves the fire detection rate and reduces the generation of false alarm. The drawback of this method is Correct and punctual detection of fire is not possible and comparison is required to identify smoke.

Chapter 3

Proposed System & Description

Chapter 3

Purposed System and Description

3.1. Proposed System:

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. 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. Flame properties for fire detection have been used as shown in Fig 3.1.

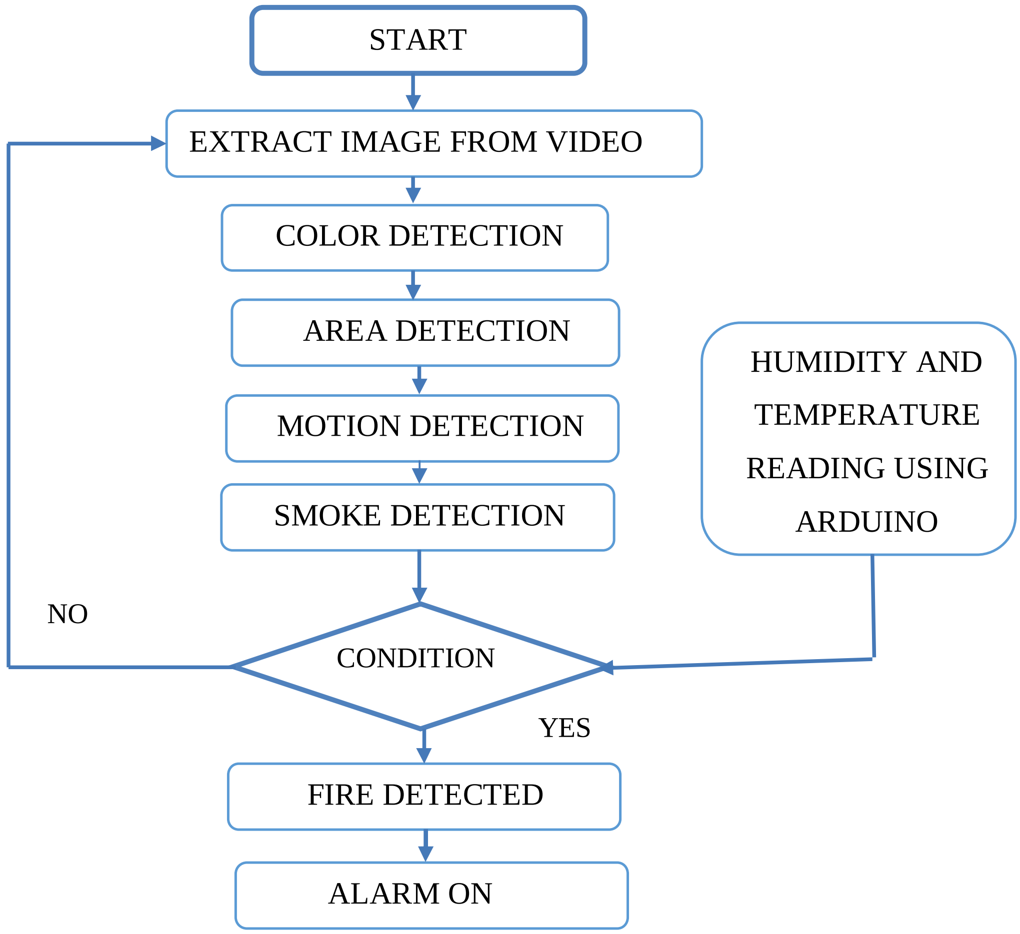


Fig. 3.1. Flow Chart

3.2. Description

In our proposed Fire Detection System, we detect the fire based on the various parameters and condition as shown in above Fig. 3.1. Flowchart. Firstly, our system extracts images of the environment on a real-time basis, in every 2 seconds. These images then go through the detection techniques of : Area detection, Color detection, Motion detection and Smoke detection. Also, continuous monitoring of environmental Humidity and Temperature is done alongside using DHT11 sensor and Arduino.

For detection purpose two consecutive frames are considered at a time, to make corresponding comparison and analysis. The captured images first go through Area detection, where the area under fire is detected in by converting RGB into HSV color space. After area detection we go further for Color detection. In Color detection the RGB components of the captured image are separated and also it is converted from RGB to YCbCr color space. Then based on various comparisons in RGB and YCbCr color space and also, using thresholds which we have decided by experimental evaluations, color detection is done. Then we go for motion detection, where we convert the 2 frames from RGB into gray and after comparison we check for the mean motion threshold, which is decided after experimental evaluation and motion detection is done. For Smoke detection, we keep the extracted images in RGB color space and based on the decided smoke threshold and evaluated mean threshold the frames are processed and smoke detection is done.

On the start of monitoring, after going through the mentioned detection techniques, the condition is checked to give final discretion that fire is detected or not, depending on which the alert alarm is turned ON.

The condition to be checked is:

If (Humidity<=Humidity_Th)||(Temperature>=Temperature_Th) || Fire_Area_Flag1&& (Fire_Color_Flag1 || Fire_Motion_Flag1 || Fire_Smoke_Flag1)

Specified condition allows us to use different combinations of detection methods flexibly. So that, we can implement the system according to the specific requirements of use.

For example:

1. For highly sensitive area, we can apply the OR gate (||) operator. So that the system will prompt for fire, if any of the method will detect the occurrence.
2. For general purpose, we can apply the combination of any two methods. So that the system will prompt for fire, if at least two methods will detect the fire.
3. For Less sensitive area, we can apply the AND gate (&&) operator. So that the system will prompt for fire, only if all methods will detect the fire.

If the mentioned condition is satisfied then Alert alarm is turned ON, wherein a message is popped showing the Cam on which the fire is detected and buzzer is started and if it is not satisfied then the system keeps on extracting images and applying the above mention procedure until we stop the monitoring.

Chapter 4

Hardware & Software Components

Chapter 4

Hardware and software components

4.1 Software Requirements

MATLAB Software

MATLAB® is the high-level language and interactive environment used by millions of engineers and scientists worldwide. It lets you explore and visualize ideas and collaborate across disciplines including signal and image processing, communications, control systems, and computational finance.

MATLAB can be used in projects such as modeling energy consumption to build smart power grids, developing control algorithms for hypersonic vehicles, analyzing weather data to visualize the track and intensity of hurricanes, and running millions of simulations to pinpoint optimal dosing for antibiotics.

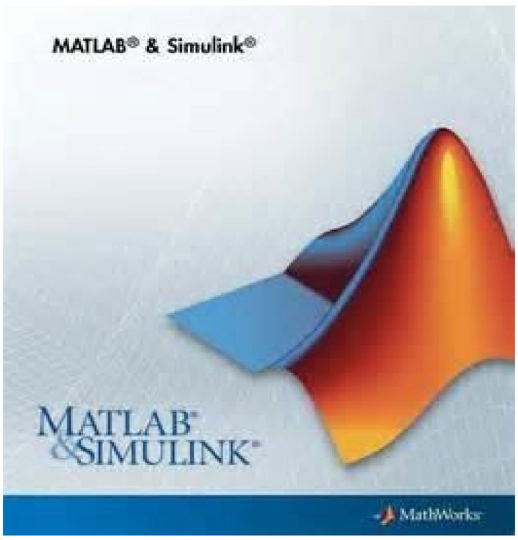


Fig 4.1.1: MATLAB LOGO

Key Features:

1. High-level language for numerical computation, visualization, and application development
2. Interactive environment for iterative exploration, design, and problem solving
3. Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration, and solving ordinary differential equations
4. Built-in graphics for visualizing data and tools for creating custom plots
5. Development tools for improving code quality and maintainability and maximizing performance
6. Tools for building applications with custom graphical interfaces
7. Functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET, and Microsoft® Excel.®

Functions:

1. Numeric Computation

MATLAB provides a range of numerical computation methods for analyzing data, developing algorithms, and creating models. The MATLAB language includes mathematical functions that support common engineering and science operations. Core math functions use processor optimized libraries to provide fast execution of vector and matrix calculations.

Available methods include:

1. Interpolation and regression
2. Differentiation and integration
3. Linear systems of equations
4. Fourier analysis
5. Eigen values and singular values

6. Ordinary differential equations (ODEs)
7. Sparse matrices

2. Data Analysis and Visualization.

MATLAB provides tools to acquire, analyze, and visualize data, enabling you to gain insight into your data in a fraction of the time it would take using spreadsheets or traditional programming languages. You can also document and share your results through plots and reports or as published MATLAB code.

3. Acquiring Data.

MATLAB lets you access data from files, other applications, databases, and external devices. You can read data from popular file formats such as Microsoft Excel; text or binary files; image, sound, and video files; and scientific files such as netCDF and HDF. File I/O functions let you work with data files in any format. Using MATLAB with add-on products, you can acquire data from hardware devices, such as your computer's serial port or sound card, as well as stream live, measured data directly into MATLAB for analysis and visualization. You can also communicate with instruments such as oscilloscopes, function generators, and signal analyzers.

4. Analyzing Data.

MATLAB lets you manage, filter, and preprocess your data. You can perform exploratory data analysis to uncover trends, test assumptions, and build descriptive models. MATLAB provides functions for filtering and smoothing, interpolation, convolution, and fast Fourier transforms (FFTs). Add-on products provide capabilities for curve and surface fitting, multivariate statistics, spectral analysis, image analysis, system identification, and other analysis tasks.

5. Visualizing Data.

MATLAB provides built-in 2-D and 3-D plotting functions, as well as volume visualization functions. You can use these functions to visualize and understand data and communicate results. Plots can be customized either interactively or programmatically. The MATLAB plot gallery provides examples of many ways to display data graphically in MATLAB. For each example, you can view and download source code to use in your MATLAB application.

6. Documenting and Sharing Results.

You can share results as plots or complete reports. MATLAB plots can be customized to meet publication specifications and saved to common graphical and data file formats. You can automatically generate a report when you execute a MATLAB program. The report contains your code, comments, and program results, including plots. Reports can be published in a variety of formats, such as HTML, PDF, Word, or LaTeX.

7. Programming and Algorithm Development.

MATLAB provides a high-level language and development tools that let you quickly develop and analyze algorithms and applications.

8. Application Development and Deployment.

MATLAB tools and add-on products provide a range of options to develop and deploy applications. You can share individual algorithms and applications with other MATLAB users or deploy them royalty-free to others who do not have MATLAB.

9. Designing Graphic User Interface.

Using GUIDE (Graphical User Interface Development Environment), you can lay out, design, and edit custom graphical user interfaces. You can include common controls such as list boxes, pull-down menus, and push buttons, as well as MATLAB plots. Graphical user interfaces can also be created programmatically using MATLAB functions.

10. Generating C Code.

You can use MATLAB Coder to generate standalone C code from MATLAB code. MATLAB Coder supports a subset of the MATLAB language typically used by design engineers for developing algorithms as components of larger systems. This code can be used for standalone execution, for integration with other software applications, or as part of an embedded application.

Development Tools.

MATLAB includes a variety of tools for efficient algorithm development, including:

1. **Command Window** - Lets you interactively enter data, execute commands and programs, and display results
2. **MATLAB Editor** - Provides editing and debugging features, such as setting breakpoints and stepping through individual lines of code.
3. **Code Analyzer** - Automatically checks code for problems and recommends modifications to maximize performance and maintainability
4. **MATLAB Profiler** - Measures performance of MATLAB programs and identifies areas of code to modify for improvement.

Syntax.

The MATLAB application is built around the MATLAB language, and most use of MATLAB involves typing MATLAB code into the Command Window (as an interactive mathematical shell), or executing text files containing MATLAB code, including scripts and/or functions.

Variables.

Variables are defined using the assignment operator, =. MATLAB is a weekly typed programming language because types are implicitly converted. It is an inferred typed language because variables can be assigned without declaring their type, except if they are to be treated as symbolic objects, and that their type can change. Values can come from constants, from computation involving values of other variables, or from the output of a function.

Matrices.

Matrices can be defined by separating the elements of a row with blank space or comma and using a semicolon to terminate each row. The list of elements should be surrounded by square brackets: []. Parentheses: () are used to access elements and sub arrays (they are also used to denote a function argument list).

Structures.

MATLAB has structure data types. Since all variables in MATLAB are arrays, a more adequate name is "structure array", where each element of the array has the same field names. In addition, MATLAB supports dynamic field names (field look-ups by name, field manipulations, etc.). Unfortunately, MATLAB JIT does not support MATLAB structures; therefore just a simple bundling of various variables into a structure will come at a cost.

GUI Programming

MATLAB supports developing applications with graphical user interface features. MATLAB includes GUIDE (GUI development environment) for graphically designing GUIs. It also has tightly integrated graph-plotting features.

Applications.

1. Data Exploration ,Acquisition ,Analyzing &Visualization
2. Engg drawing and Scientific graphics
3. Analyzing of algorithmic designing and development
4. Mathematical functions and Computational functions
5. Simulating problems prototyping and modeling
6. Application development programming using GUI building environment.

4.2. Hardware Requirement

(a)Arduino IC

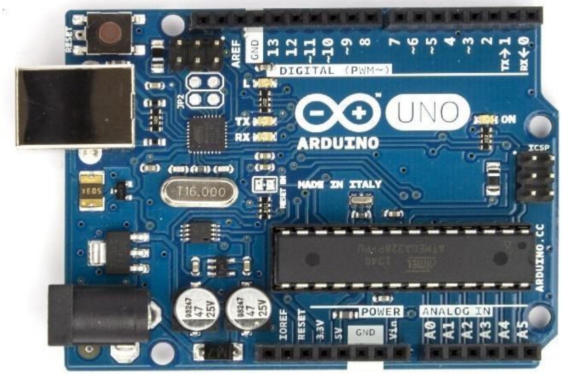


Fig. 4.2.1. Arduino IC

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

It has the following features:

1. 1.0 pin out: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with the Arduino Due that operates with 3.3V. The second one is a not connected pin that is reserved for future purposes.
2. Stronger RESET circuit.
3. Atmega 328P.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Table 4.1. Arduino specifications

Microcontroller	**ATmega328**
Operating Voltage	5V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by boot loader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16Mhz

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC to DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts.

If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

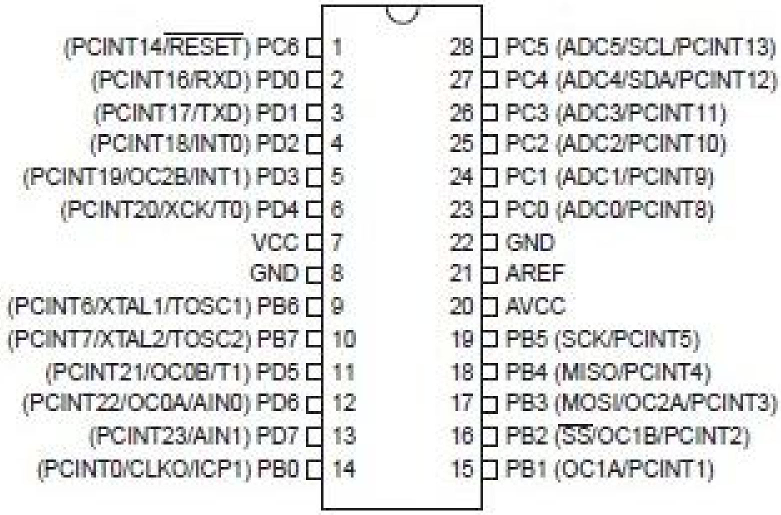


Fig 4.2.2. Pin diagram of Arduino IC.

Pin Descriptions:

1. **VCC:** Digital supply voltage.
2. **GND:** Ground.
3. **Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2:**

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit. Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier. If the Internal Calibrated RC Oscillator is used as chip clock source, PB7…6 is used as TOSC2…1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

4. **Port C (PC5:0):**

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5…0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tristated when a reset condition becomes active, even if the clock is not running.

5. PC6/RESET:

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C. If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given .Shorter pulses are not guaranteed to generate a Reset.

6. Port D (PD7:0):

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tristated when a reset condition becomes active, even if the clock is not running.

8. AVCC:

AVCC is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter.

9. AREF: AREF is the analog reference pin for the A/D Converter.

10. ADC7: 6 (TQFP and QFN/MLF Package Only):

In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

(B) DHT 11 Humidity & Temperature Sensor

1. Introduction

This DF Robot DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.

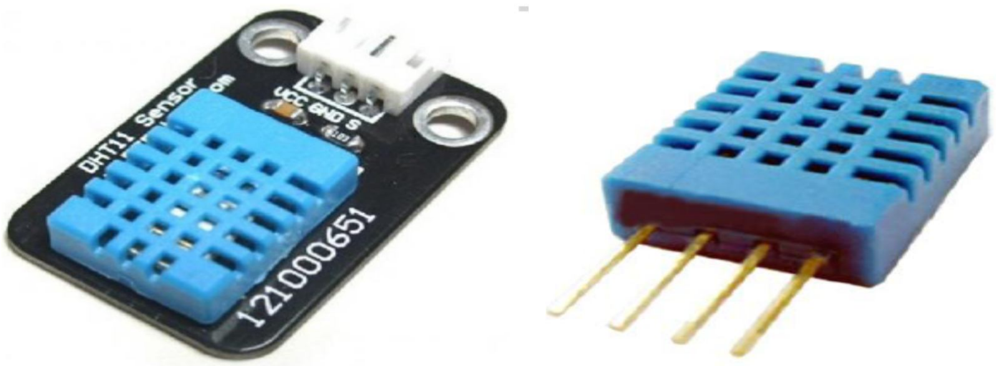


Fig.4.3.1. Humidity and Temperature sensor

Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor’s internal signal detecting process. The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package. It is convenient to connect and special packages can be provided according to users’ request.

2. Technical Specifications:

Overview:

Item	Measurement Range	Humidity accuracy	Temperature accuracy	Resolution	Package
DHT11	20-90% RH 0-50 ºC	±5%RH	±2ºC	1	4 Pin Single

Table 4.2: Overview of DHT11

Detailed Specifications:

Parameters	**Conditions**	**Minimum**	**Typical**	**Maximum**
Humidity				
Resolution		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			±1%RH	
Accuracy	25ºC		±4%RH	
	0-50ºC			±5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0ºC	30%RH		90%RH
	25ºC	20%RH		90%RH
	50ºC	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25 ºC, 1m/s Air	6 S	10 S	15 S
Hysteresis			±1%RH	
Long term Stability	Typical		±1%RH/year	
Temperature				
Resolution		1ºC	1ºC	1ºC
		8 Bit	8 Bit	8 Bit
Repeatability			±1ºC	
Accuracy		±1ºC		±2ºC
Measurement Range		0ºC		50ºC
Response Time (Seconds)	1/e (63%)	6 S		30 S

Table 4.3. Detail specification of DHT11

Chapter 5

Methodology

Chapter 5

Methodology

This algorithm is based in the fact that visual color images of fire have high absolute values in the red component of the RGB coordinates. This property permits simple threshold-based criteria on the red component of the color images to segment fire images in natural scenarios. However, not only fire gives high values in the red component. Another characteristic of fire is the ratio between the red component and the blue and green components. An image is loaded into color detection system. Color detection system applies the specific property of RGB pixels and give the output result as an image with a selected area of color detection. Rule based color model approach has been followed due to its simplicity and effectiveness. For that, color space RGB and YCbCr is chosen. For classification of a pixel to be fire we have identified seven rules. If a pixel satisfies these seven rules, we say that pixel belong to fire class.

5.1. Color Detection:

Classification of fire pixel

This section covers the detail of the proposed fire pixel classification algorithm. Figure shows the flow chart of the proposed algorithm. Rule based color model approach has been followed due to its simplicity and effectiveness. For that, color space RGB and YCbCr is chosen. For classification of a pixel to be fire we have identified seven rules. If a pixel satisfies these seven rules, we say that pixel belong to fire class.

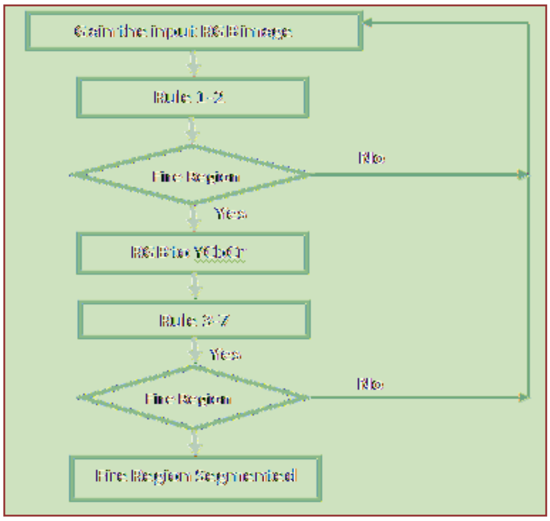


Fig. 5.1.1. Flow chart of color

A digital colored image has three planes: Red, Green and Blue (R, G, and B). The combination of RGB color planes gives ability to devices to represent a color in digital environment. Each color plane is quantized into discrete levels. Generally 256 (8 bits per color plane) quantization levels are used for each plane, for instance white is represented by (R, G, B) = (255, 255, 255) and black is represented by (R, G, B) = (0, 0, 0). A color image consists of pixels, where each pixel is represented by spatial location in rectangular grid (x, y), and a color vector (R(x, y), G(x, y), B(x, y)) corresponding to spatial location (x, y).

Rule 1

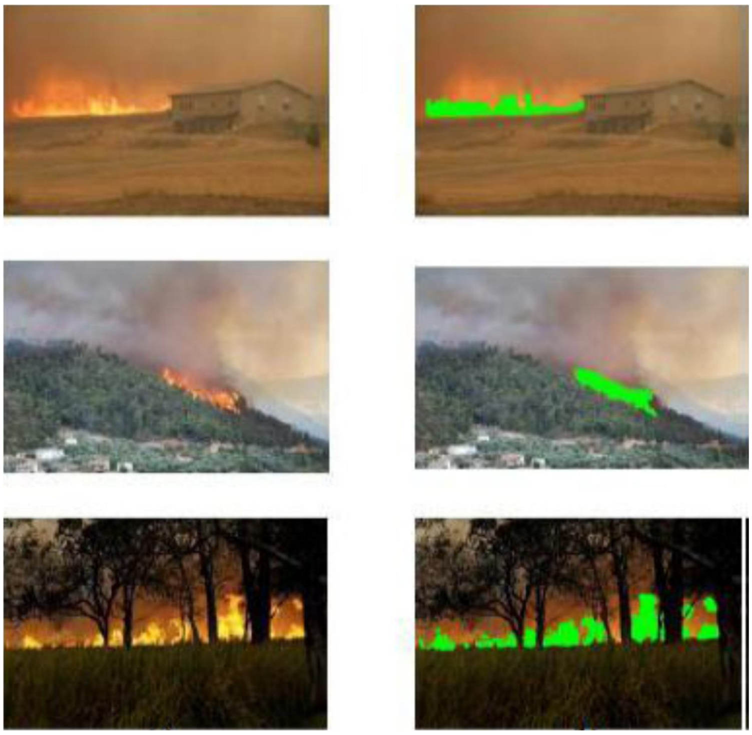
It can be noticed from Figure 5.1.2 that for the fire regions, R channel has higher intensity values than the G channel, and G channel has higher intensity values than the B channel.

(a) (b) (c) (d)

Fig.5.1.2: Original RGB image in column (a), and R, G, and B channels in column (b)-(c)-(d), respectively.

In order to explain this idea better, we picked sample images from Figure 5.1.3 (a), and segmented its fire pixels as shown in Figure 5.1.3 (b) with green color. Then we calculate mean values of R, G, and B planes in the segmented fire regions of the original images. The results are given in Table I for the images given in Figure 5.1.3. It is clear that, on the average, the fire pixels show the characteristics that their R intensity value is greater than G value and G intensity value is greater than the B. So, for a pixel at spatial location (x, y) to be fire pixel the below rule must be satisfied.

$$R_1(x,y) = \begin{cases} 1, & \text{if } R(x,y) > G(x,y) > B(x,y) \\ 0, & \text{otherwise} \end{cases} \quad \text{...........(1)}$$



(a) (b)

Fig.5.1.3: Original RGB images are given in column (a), and corresponding fire region manually segmented in column (b)

Row	Mean R	Mean G	Mean B
1	252.5	210.3	83.1
2	174.4	121.1	89.4
3	250.1	208.6	40.8

Table 5.1: Mean value of RGB for images given in Fig.5.1.3

Rule II

From the histogram analysis of the fire location which is manually segmented as, shown in Figure 5.1.4. We have identified some threshold values for the pixel to be fire. We have threshold values for all the three planes i.e., R, G, and B plane. We have tested this threshold values for hundreds of images from our data set.

Fig.5.1.4: (a) Original RGB image given left and (b) Corresponding fire regions, manually labeled with green color are shown in right side.

Based on this observation of a large number of test images, the following rule can be established for detecting a fire pixel at the spatial location (x, y):

$$\boldsymbol{R_2(x,y)} = \begin{cases} \mathbf{1}, & \boldsymbol{if\ (R(x,y) > 190) \cap (G(x,y) > 100) \cap (B(x,y) < 140} \\ \mathbf{0}, & \boldsymbol{otherwise} \end{cases} \quad \text{------- (2)}$$

When the image is converted from RGB to YCbCr color space, intensity and chrominance is easily discriminated. This help to model the fire region easily in YCbCr color space.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2568 & 0.5041 & 0.0979 \\ -0.1482 & -0.2910 & 0.4392 \\ 0.4392 & -0.3678 & -0.0714 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad \text{---------(a)}$$

Where Y is luminance, Cb and Cr are Chrominance Blue and Chrominance Red components, respectively.

Given a RGB-represented image, it is converted into YCbCr represented color image using the standard RGB to YCbCr.

The mean values of the three components Y, Cb, and Cr, denoted by Y_{mean}, Cb_{mean} and Cr_{mean} respectively are computed as follows:

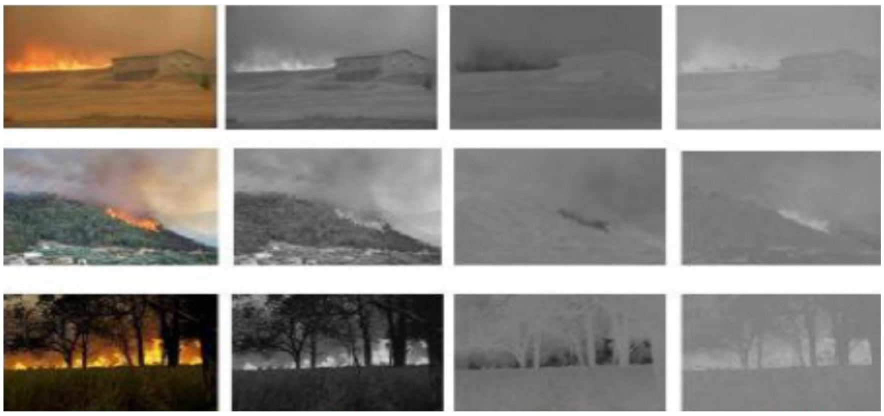
$$Y_{mean}(x,y) = \frac{1}{M \times N}\sum_{x=1}^{M}\sum_{y=1}^{N} Y(x,y)$$

$$Cb_{mean}(x,y) = \frac{1}{M \times N}\sum_{x=1}^{M}\sum_{y=1}^{N} Cb(x,y)$$

$$Cr_{mean}(x,y) = \frac{1}{M \times N}\sum_{x=1}^{M}\sum_{y=1}^{N} Cr(x,y)$$

---------(b)

Where, (x,y) denotes the spatial location of pixels, M × N is the total number of pixels in the given image.



(a) (b) (c) (d)

Fig.5.1.5: Original RGB image in column (a), and Y, Cb, and Cr channels in column (b)-(d), respectively.

Row index in Fig. 3	Mean Y	Mean Cb	Mean Cr
1	130	106	153
2	189.3	48.1	158.2
3	195	65.8	155.6

Table 5.2: Mean value of YCbCr for images given in Fig.5. 1.5

Rule 3 and 4

From the Table II we have developed the following two rules. The rules were observed for a large amount of test images from our data base.

The two rules for detecting a fire pixel at spatial location (x,y):

$$R_3(x,y) = \begin{cases} 1, & \text{if } Y(x,y) \geq Cb(x,y) \\ 0, & \text{otherwise} \end{cases}$$

………..(3)

$$R_4(x,y) = \begin{cases} 1, & \text{if } Cr(x,y) \geq Cb(x,y) \\ 0, & \text{otherwise} \end{cases}$$

………..(4)

Rule 5

The flame region is generally the brightest region in the observed scene, the mean values of the three channels, in the overall image Y_{mean}, Cb_{mean} and Cr_{mean} contain valuable information. From the Figure 7 it can be observed that for the flame region the value of the Y component is bigger than the mean Y component of the overall image while the value of Cb component is in general smaller than the mean Cb value of the overall image. Furthermore, the Cr component of the flame region is bigger than the mean Cr component.

These observations which are verified over countless experiments with images containing fire regions are formulated as the following rule:

$$R_5(x,y) = \begin{cases} 1, & \text{if } \big(Y(x,y) \geq Y_{mean}(x,Y)\big) \cap \big(Cb(x,y) \leq Cb_{mean}(x,y)\big) \cap \big(Cr(x,y) \geq Cr_{mean}(x,y)\big) \\ 0, & \text{otherwise} \end{cases}$$

……….. (5)

Where, R5(x,y) indicates that any pixel that satisfies the condition given in eq.(5) is considered to be fire pixel.

Rule 6

It can easily be observed from Figure 5(c) and Figure 5(d) that there is a significant difference between the Cb and Cr components of the fire pixels. For the fire pixel the Cb component is predominantly “black” (lower intensity) while the Cr component, on the other hand, is predominantly “white” (higher intensity).

This fact can be translated into another rule as follows:

$$R_6(x, y) = \begin{cases} 1, & if\ |Cb(x,y) - Cr(x,y)| \geq Th \\ 0, & otherwise \end{cases}$$

…………(6)

Where, the value of Threshold is accurately determined according to a Receiver Operating Characteristics (ROC) Curve. The ROC curve is obtained by experimenting different values of Th (ranging from 1 to 100) over 100 color images.

First, the fire-pixel regions are manually segmented from each color image. The rules 1 through 5 (at a chosen value of Th) are then applied to the manually segmented fire regions. The true positive is defined as the decision when an image contains a fire, and false positive is defined as the decision when an image contains no fire but classified as having fire. The ROC curve consists of 100 data points corresponding to different Th values, the corresponding true positive (i.e., correct-detection) and false positive (i.e., false-alarm) rates are computed and recorded. The correct detection is defined as the same decision when an image indeed contains a fire incident, while the false alarm is defined as incorrect decision when an image contains no fire but misdetected as having fire. For good fire detection system it should not miss any fire alarm, with the established ROC curve, the value of Th is picked such that the systems true positive rate is high enough.

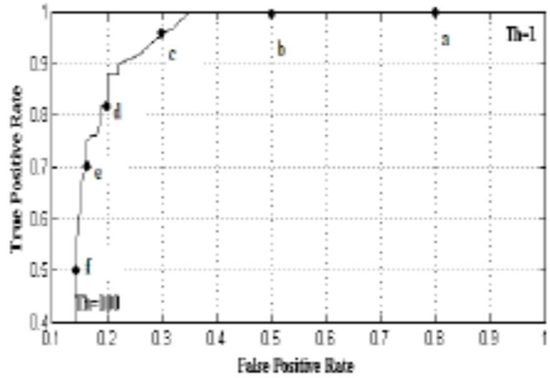
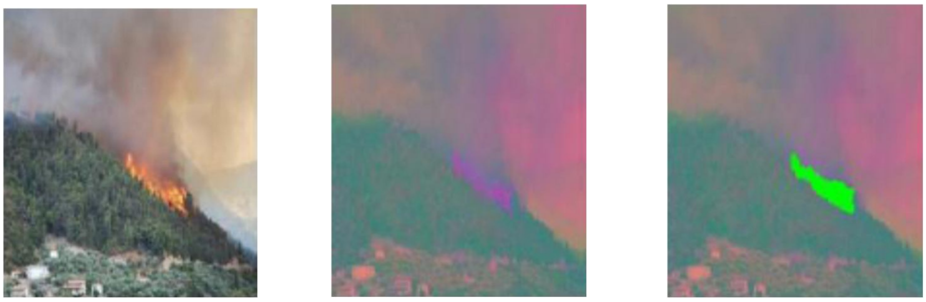


Fig.5.1.6: The receiver operating characteristics (ROC) curve for determining the desired value of Th to be used in eq. 6.

From Figure 5.1.6 it is clear that, high true positive rate means high false positive rate. Using this tradeoff, in our experiments the value of Th is picked such that the detection rate is over 95% and false alarm rate is less than 30% (point c) which corresponds to Th = 70.

Rule 7

From the histogram analysis of the fire location which is manually segmented as, shown in Figure 5.1.7(c). We have identified some threshold values for the pixel to be fire. We have threshold values for Cb and Cr planes, we are not considering Y plane because it is luminance component and it depends on illumination conditions. We have tested this threshold values for hundreds of images from our data set.



(a) (b) (c)

Fig.5.1.7: (a) Original RGB image [19], (b) RGB to YCbCr converted image, and (c) corresponding fire region manually labeled with green color on YCbCr image.

Based on this observation of a large number of test images, the following rule can be established for detecting a fire pixel at the spatial location (x,y):

$$R_7(x,y) = \begin{cases} 1, & (Cb(x,y) \leq 120) \cap (Cr(x,y) \geq 150) \\ 0, & otherwise \end{cases} \qquad \text{............(7)}$$

A pixel is classified to fire class if all the Rules I-VII is satisfied by that pixel.

5.2. Area Detection

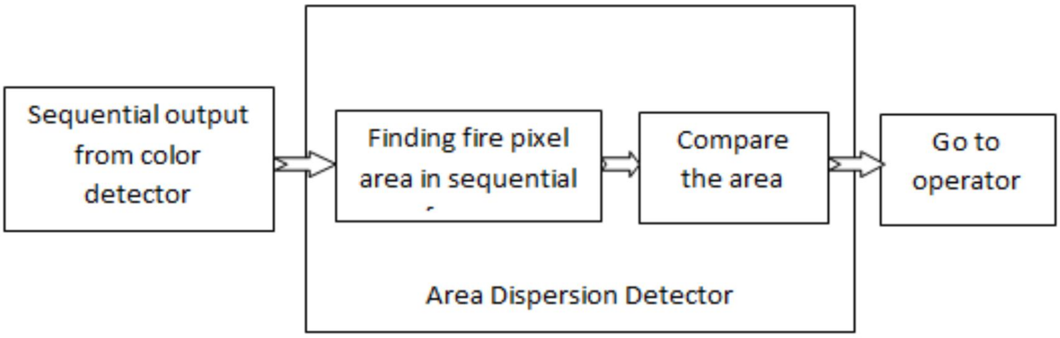
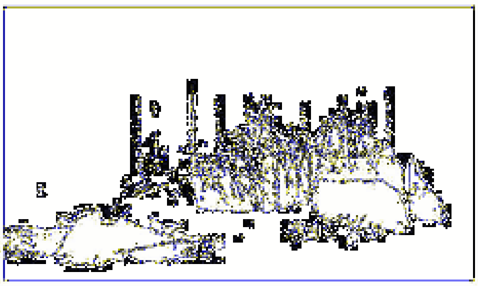


Fig. 5.2.1. Block Diagram for Area Detection

Area detection method is used to detect dispersion of fire pixel area in the sequential frames. In this method, we took two sequential images which comes out from color detector then we check dispersion in minimum and maximum coordinate of X and Y axis, acquired from color detector.

In this method we are comparing fire pixel area of two sequential frames as on the basis of minimum value of x & y and maximum value of x & y. In case of fire, if any extreme value of x and y axis will increase for next frame i.e. frame 2, then there is area dispersion takes place and system will provide output to the operator. After that operator will perform operation on the basis of logic combination selected by the system.

Fig.5.2.2 Image before Area detection



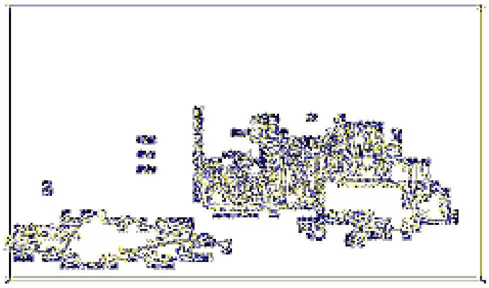


Fig. 5.2.3 Images after Area detection

5.3 Motion Detection

Motion detection is used to detect any occurrences of movement in a sample video. Block diagram of motion detection system is as in Figure 5. Using MATLAB/Simulink, a motion detector model is built based on this block diagram.

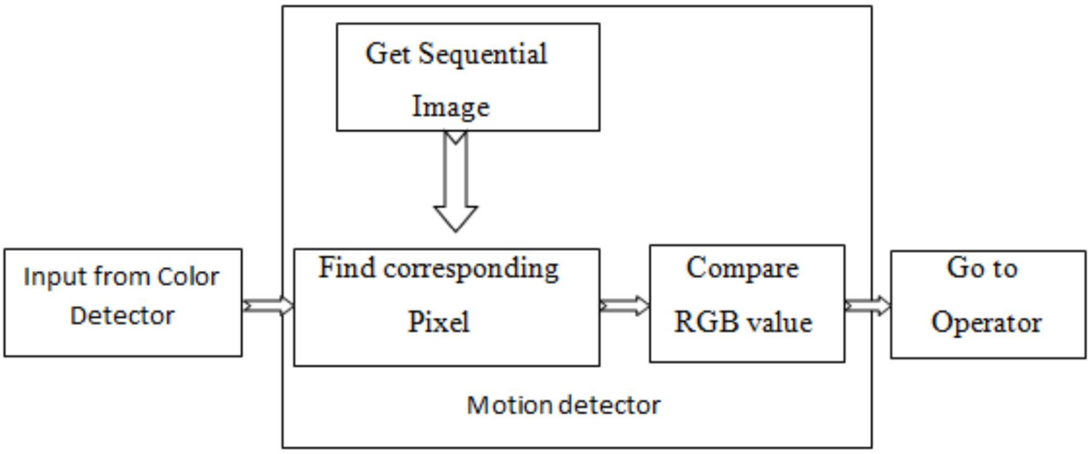


Fig.5.3.1 Block diagram of Motion Detection

We took two sequential images from video frames. After applying basic two methods edge detection and color detection we get probable area of fire pixel then we compare the RGB value to of frame1 to the frame 2 for corresponding pixel and if pixel value differs then motion detector will show motion and will give resultant output to the operator

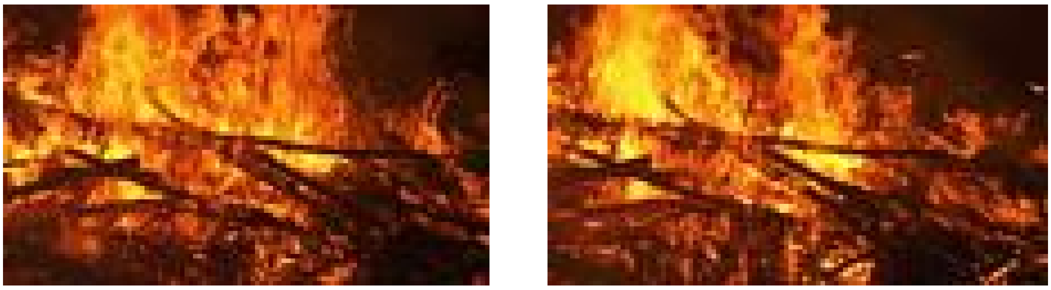


Fig. 5.3.2. Images of frame1 and frame2

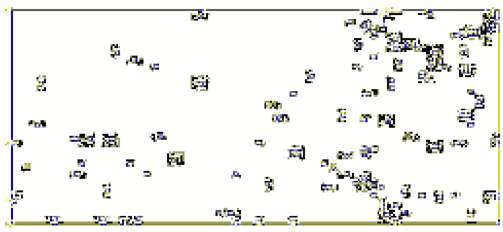


Fig. 5.3.3 Image after Motion detection

In Fig. 5.3.2 frame 1 and frame 2 are sequential images and after mapping the corresponding pixels in both of the frames, motion detector compares R, G, and B value of corresponding pixels and give the resultant output to the combination of operator.

5.4. Smoke Detection

The smoke pixels do not show chrominance characteristics like fire pixels. At the beginning, when the temperature of the smoke is low, it is expected that the smoke will show color from the range of white-bluish to white. Toward the start of the fire, the smoke's temperature increases and it gets color from the range of black-grayish to black. As can be seen from the Figure 5.4.1, most smoke samples have a grayish color. So we can formulate the smoke pixels as follows,

$$|R(x, y) - B(x, y)| \leq Th$$
$$|G(x, y) - B(x, y)| \leq Th$$
$$|R(x, y) - G(x, y)| \leq Th \qquad \text{----- (1)}$$

Where *Th* is a global threshold ranging from 15 to 25. The equation (1) states that, the smoke pixels should have similar intensities in their RGB color channels. Figure 5 shows the smoke-pixel segmentation using the equation defined in (1). Since the smoke information will be used for early fire detection system, the smoke samples should be detected when the smoke has low temperature. This is the case, where the smoke samples have color ranging from white-bluish to white, which means that the saturation of the color should be as low as possible. Using this idea, the rule defined in (2) is used where *HSV* color spaces is used.

$$S(x, y) \leq 0.1 \qquad \text{----- (2)}$$

As can be seen from the Fig. 5.4.2. That output is noisy, but the motion property of the smoke can be used to remove such noisy parts. It can be easily observed from the first row of the Figure, the sky is detected as smoke, because its property of grayish color. But, if we embed the motion detection part, the sky will be removed because of its constant color over some duration.

Fig. 5.4.1. Image before smoke detection

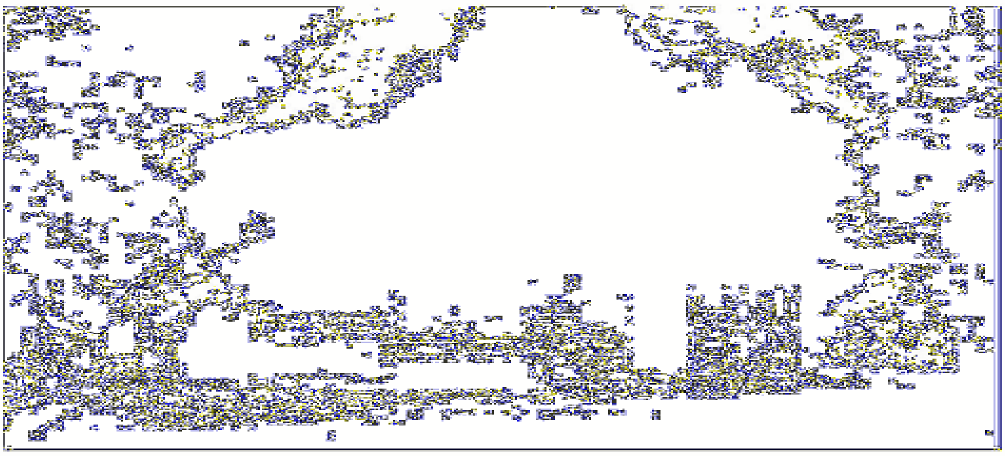


Fig. 5.4.2. Image after smoke detection

Chapter 6
Results and Discussion

Chapter 6

Results and Discussion

For practical results, we have experimented the proposed system for fire detection. The experimentation was performed to check for efficiency of the proposed system to detect actual fire and faulty detection. In the experimentation we have taken 20 trails of fire before the webcam. Among these 20 trials our system showed 2 faulty detections and 18 correct detections were made. Based on these we have made the following table 6.1

No. of trails	Correct Detection	Faulty Detection	Efficiency
20	18	2	90%

Table 6.1. Experimental values of correct detection, faulty detection and efficiency

For the comparison purposes, two sets of images are collected from Internet. One set is composed of images that consist of fire. Fire set consists of 332 images. The images in fire set show diversity in fire-color, and environmental Illuminations. The other set does not contain any fire but contains fire-like colored regions such as sun and other reddish objects. Two types of comparisons are carried out; one is for the evaluation of the correct fire detection rate and the other is for the false alarm rate. The following criterion is used for declaring a fire region: if the model achieves to detect at least 10 pixels of a fire region in a given image, then it is assumed as a correct detection, where images are in the size of 320x240. For the false alarm rate the same detection criterion is used with the non-fire image set. In Table 6.2, we have tabulated fire detection results with false alarm rates. It is clear from Table 6.2 that the new method shows better performance with respect to the technique defined, because it eliminates the colors which are similar to fire-color but it is not a fire.

Model	Detection Rate (%)	False Alarm Rate (%)
RGB, Chen et al. [3]	93.90	66.42
RGB, Celik et al. [1]	78.50	28.21
rgb, Celik et al. [2]	97.00	9.50
YCbCr, Celik et al. [10]	99.00	9.50
Proposed	99.00	4.5

Table 6.2. Performance comparisons of the models with respect detection rates, and false alarm rates

6.1 Program outputs:

Here, we have shown the actual snapshots of our program output for each evaluating parameter and for all conditions together.

1. Area detection

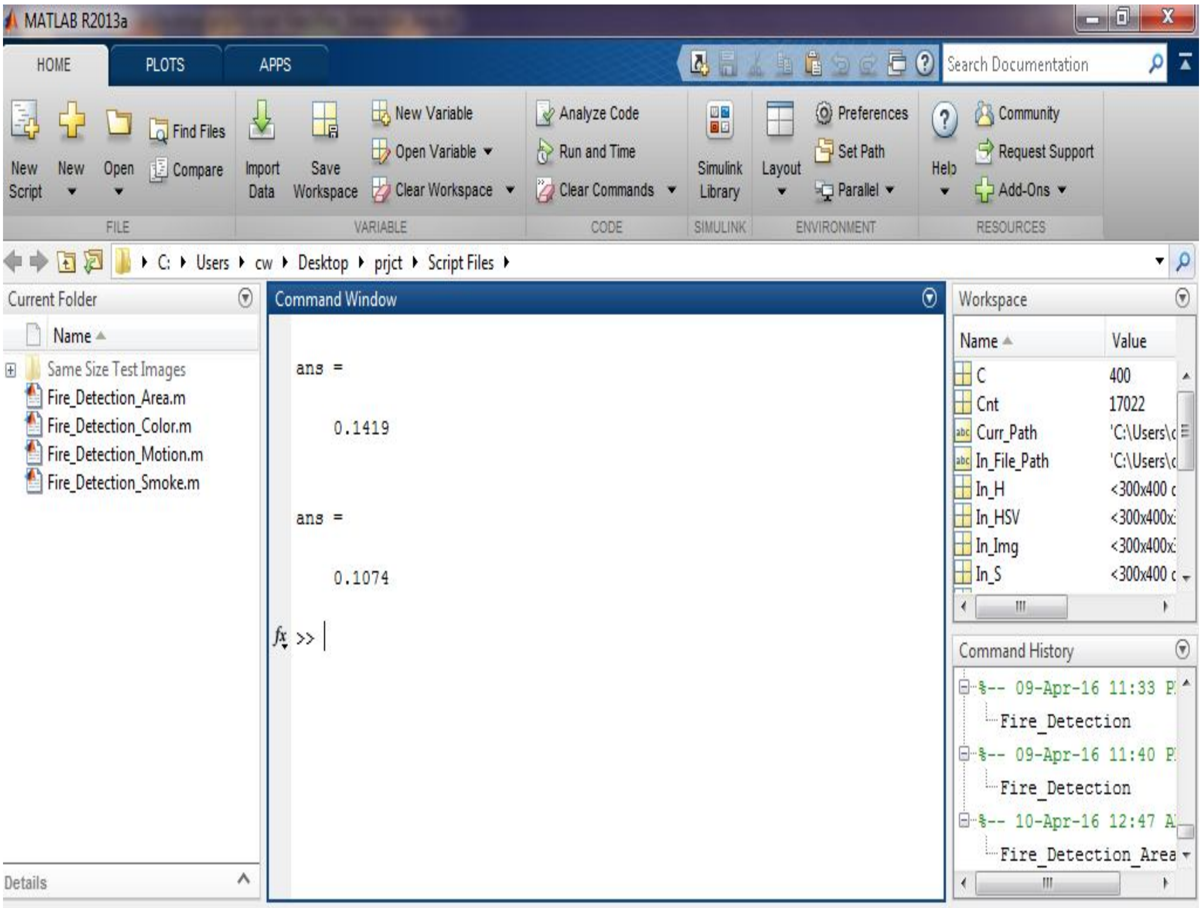


Fig.6.1.1 Mean value and standard deviation values

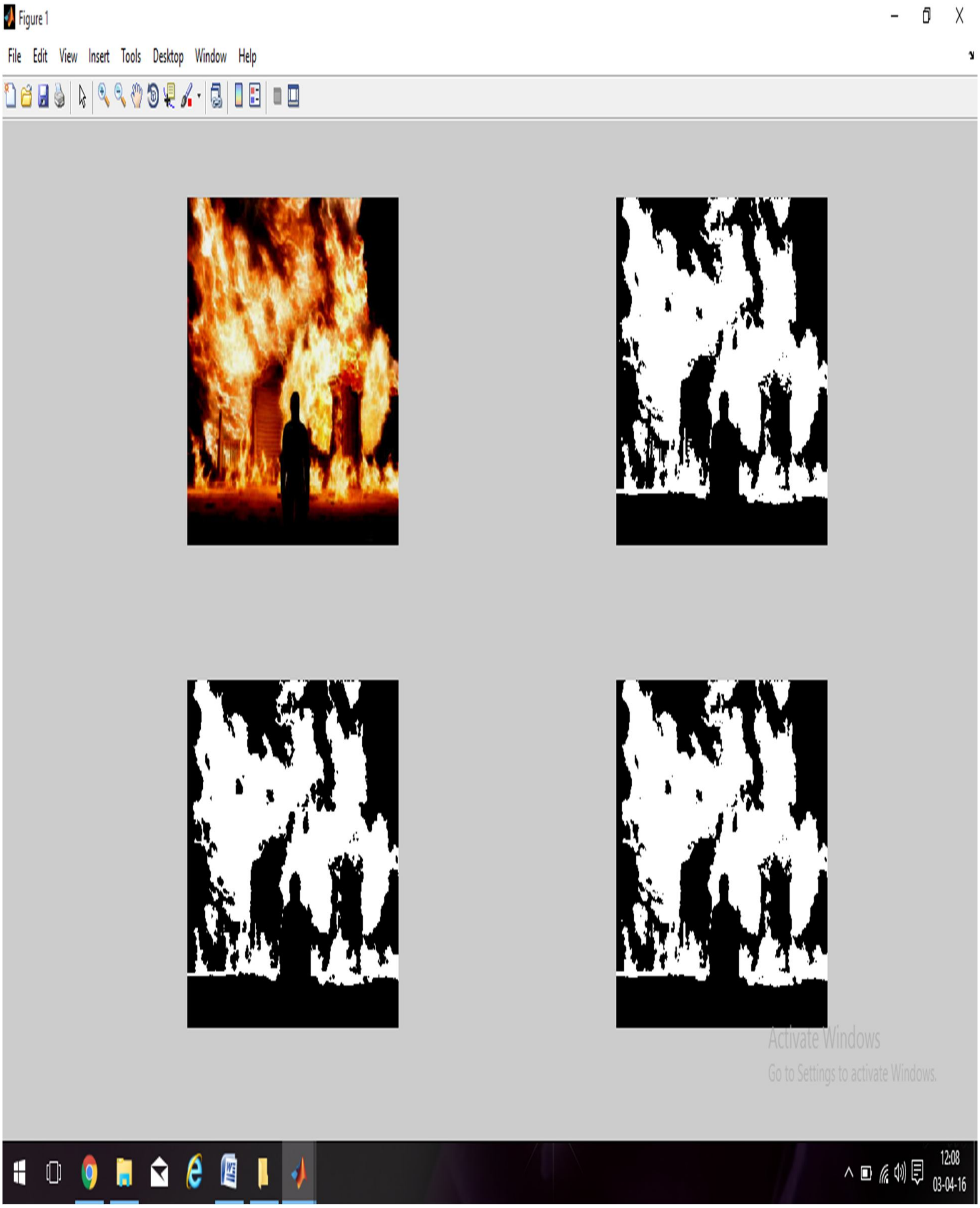


Fig. 6.1.2. Original image with eroded and dilated image

2. Motion detection:

Fig. 6.1.3. Motion detection

3. Smoke detection:

Fig. 6.1.4. smoke detection

4. Color detection:

Fig. 6.1.5. Image before color detection

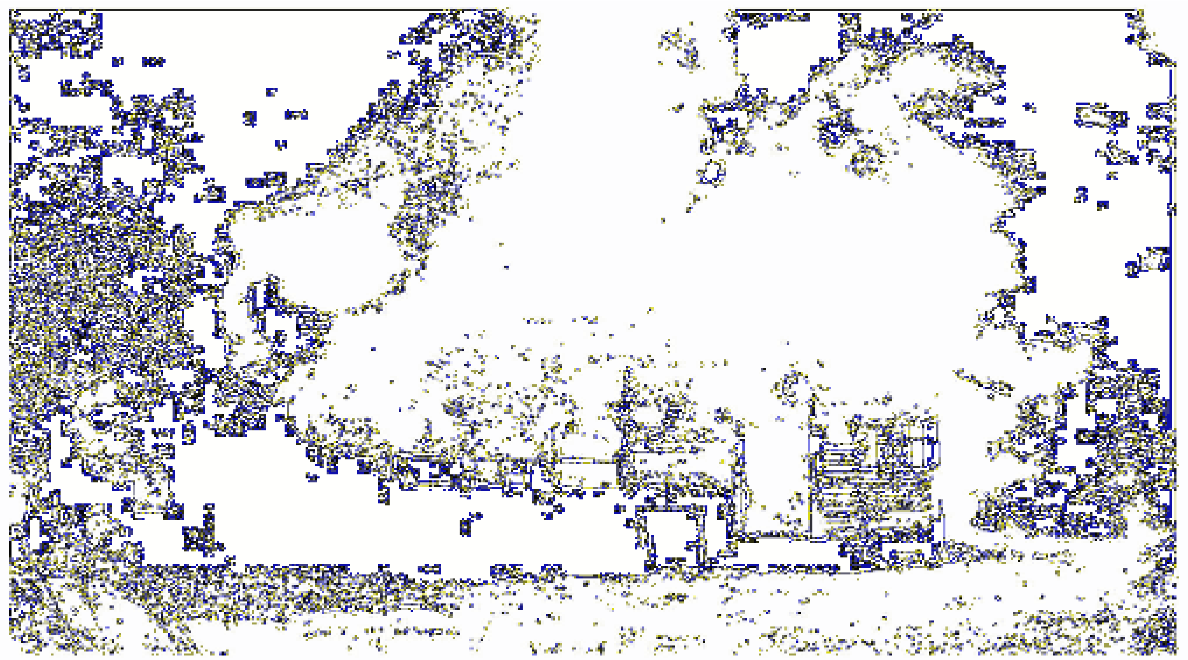


Fig. 6.1.6 (a) Image after color detection

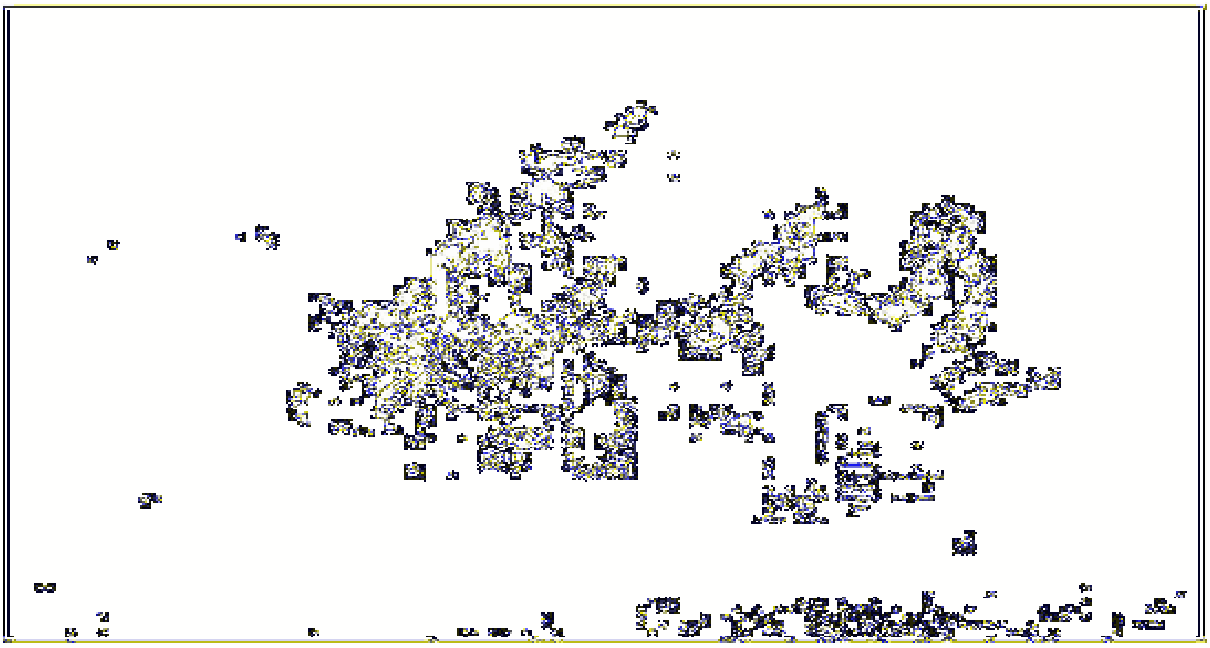


Fig. 6.1.6 (b) Image after color detection

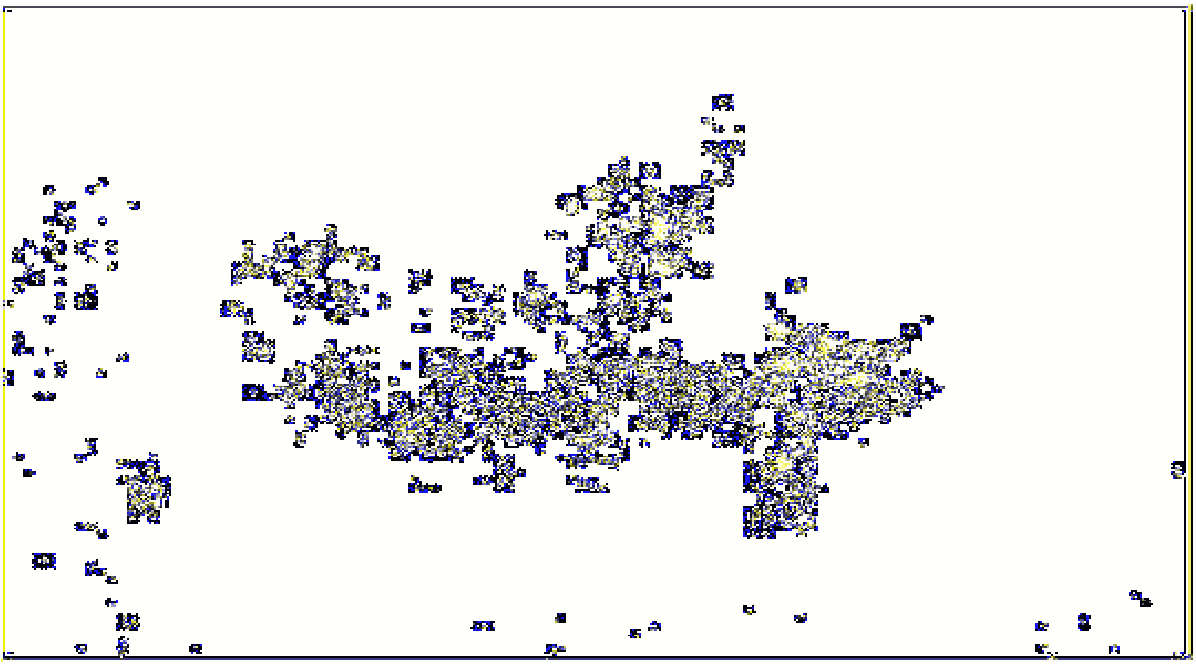


Fig. 6.1.6 (c) Image after color detection

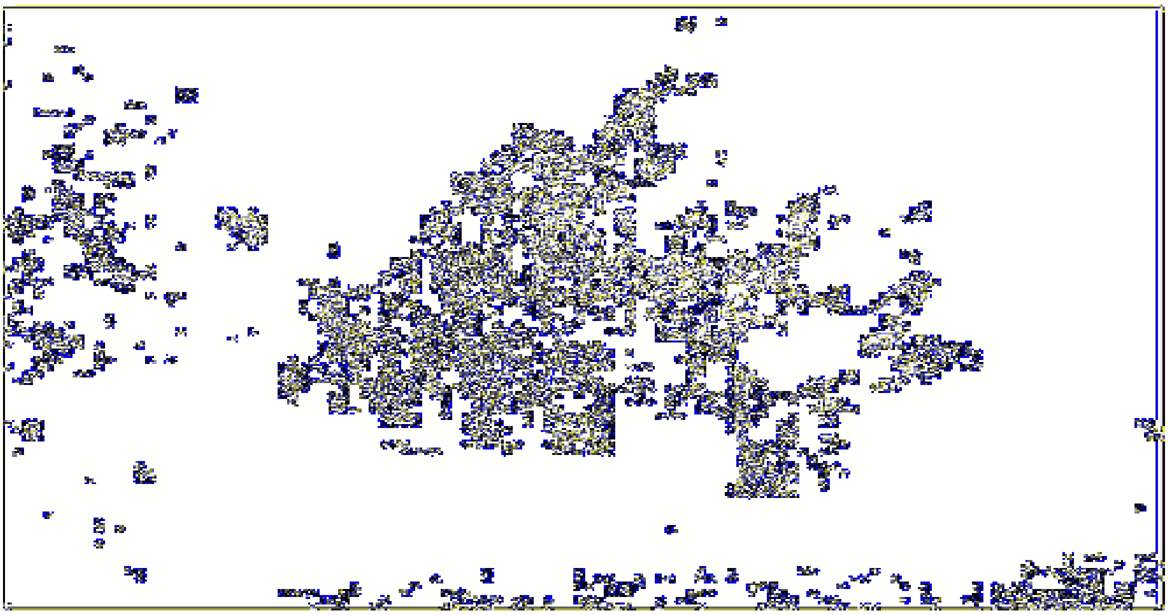


Fig. 6.1.6 (d) Image after color detection

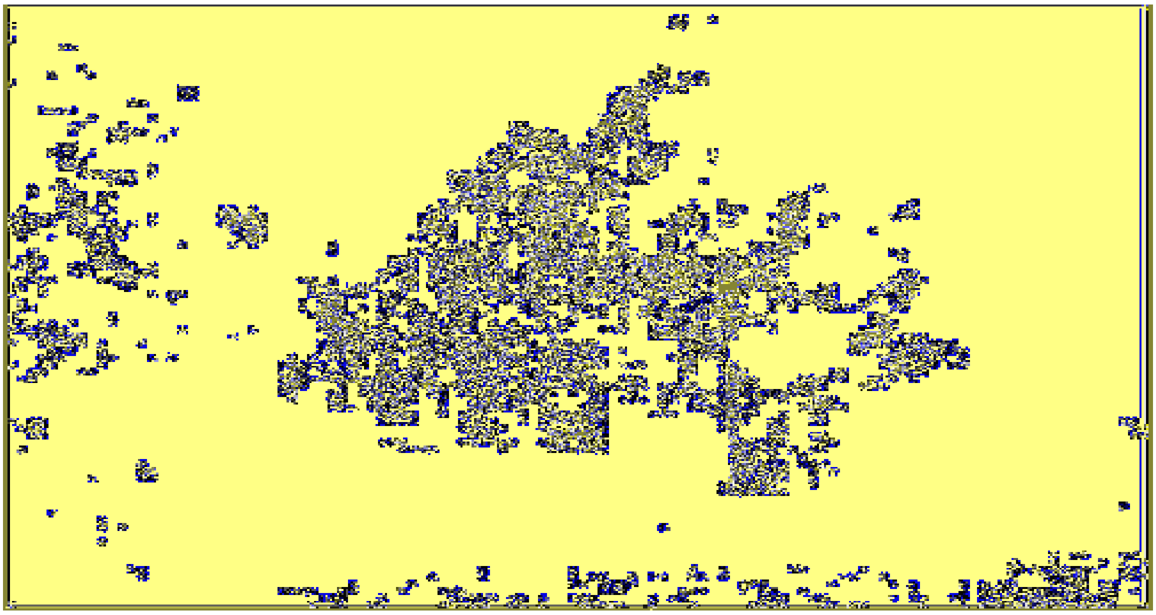


Fig. 6.1.6 (e) Image after color detection

6.2. Steps of fire detection

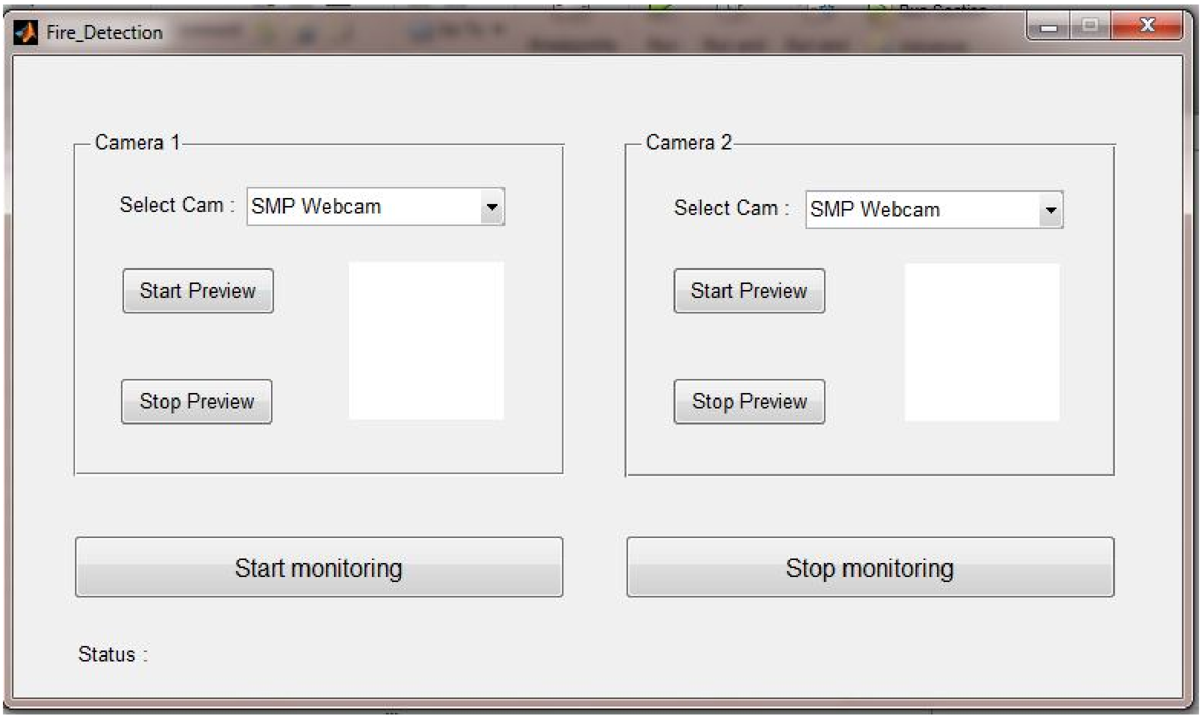


Fig.6.2.1 UI after selecting Run option

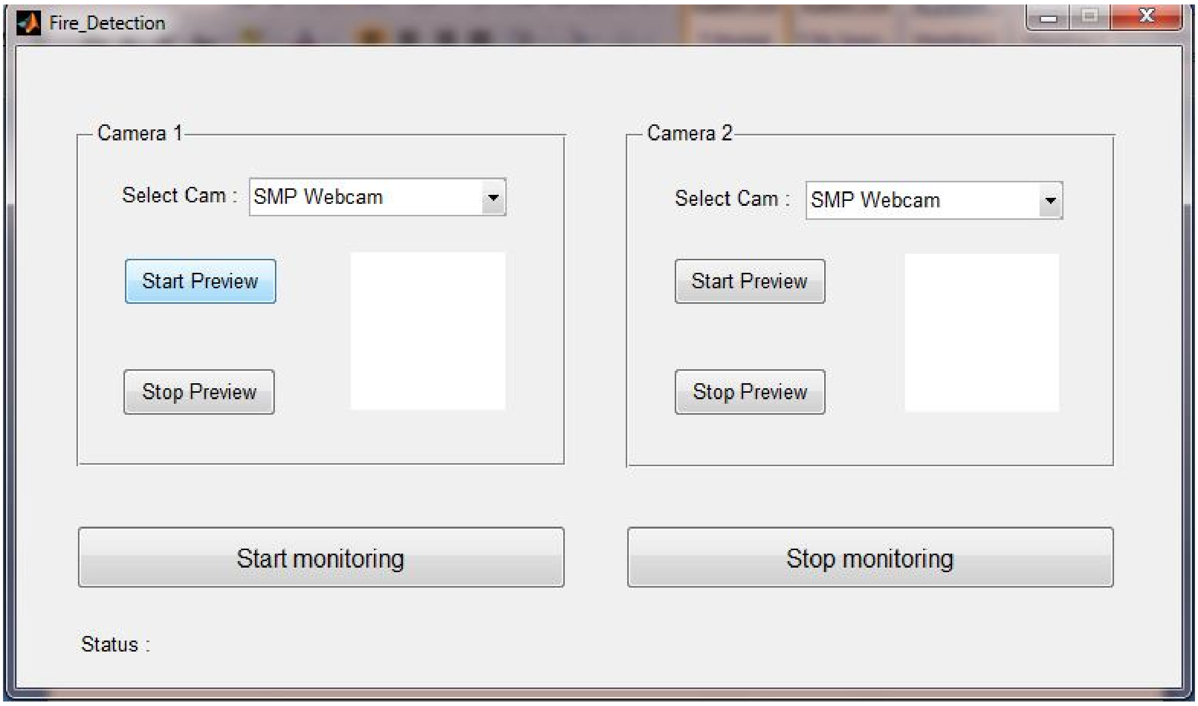


Fig. 6.2.2 Start of Preview

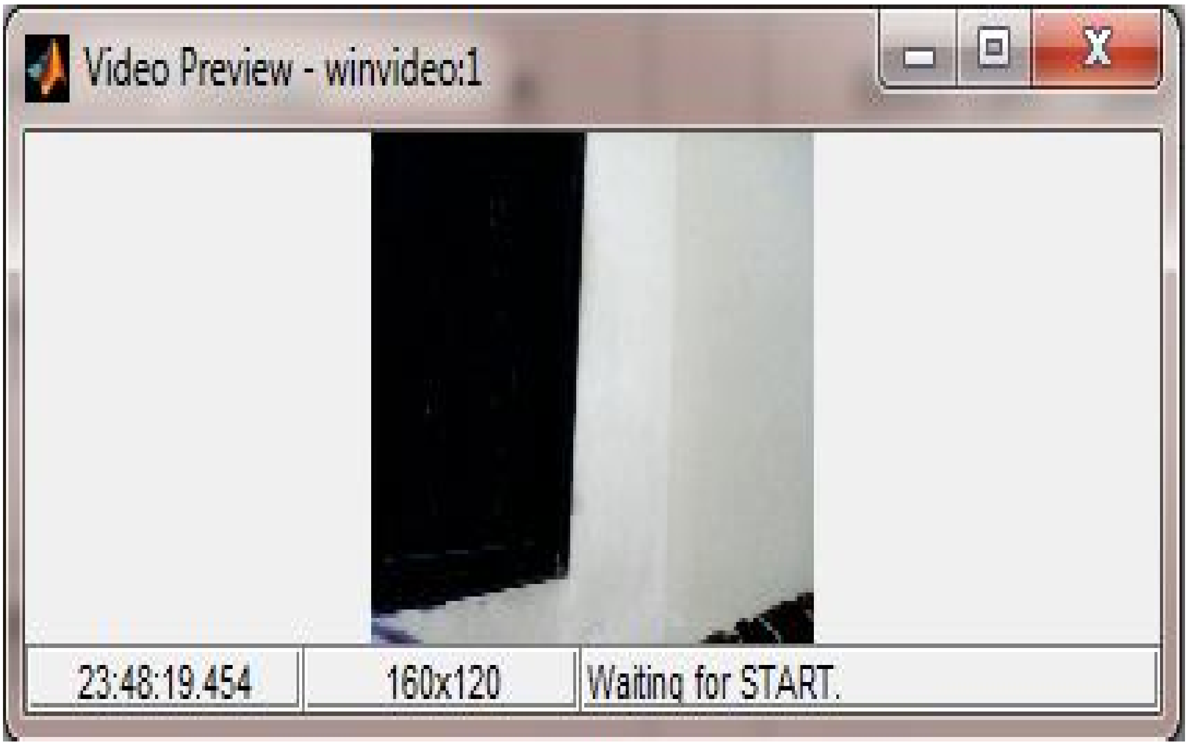


Fig. 6.2.3 Video preview

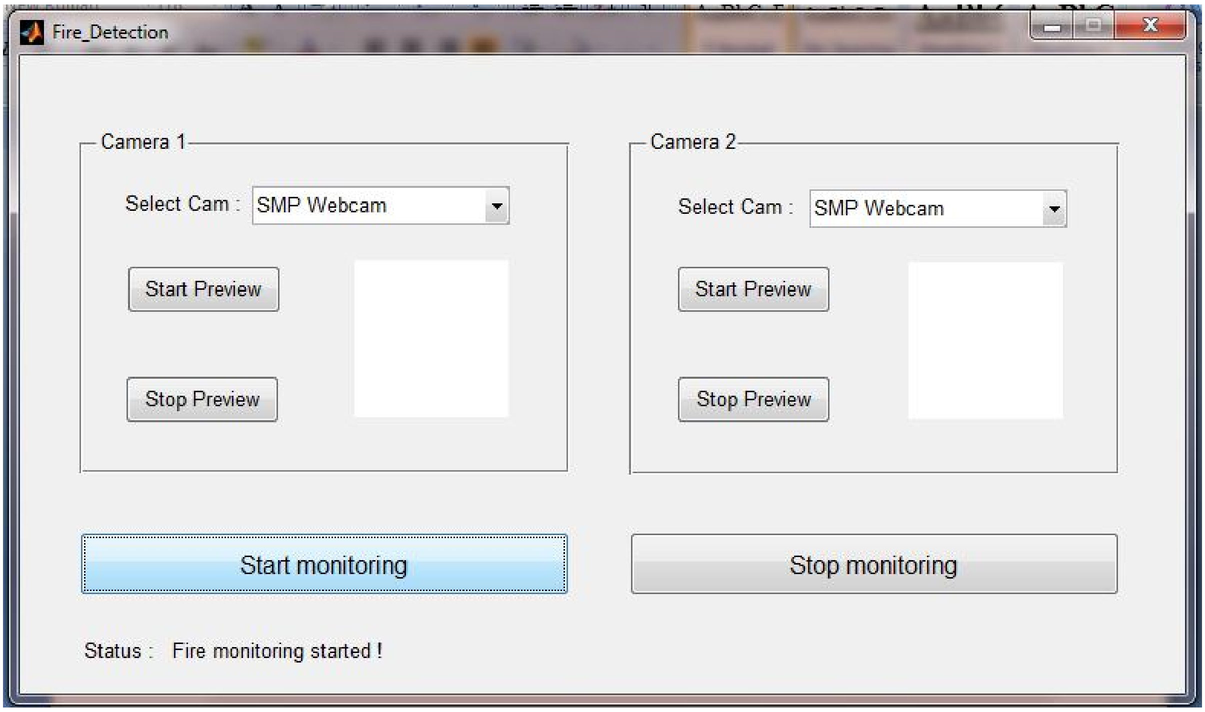


Fig. 6.2.4 Start of monitoring

Fig. 6.2.5 Capturing of fire frames

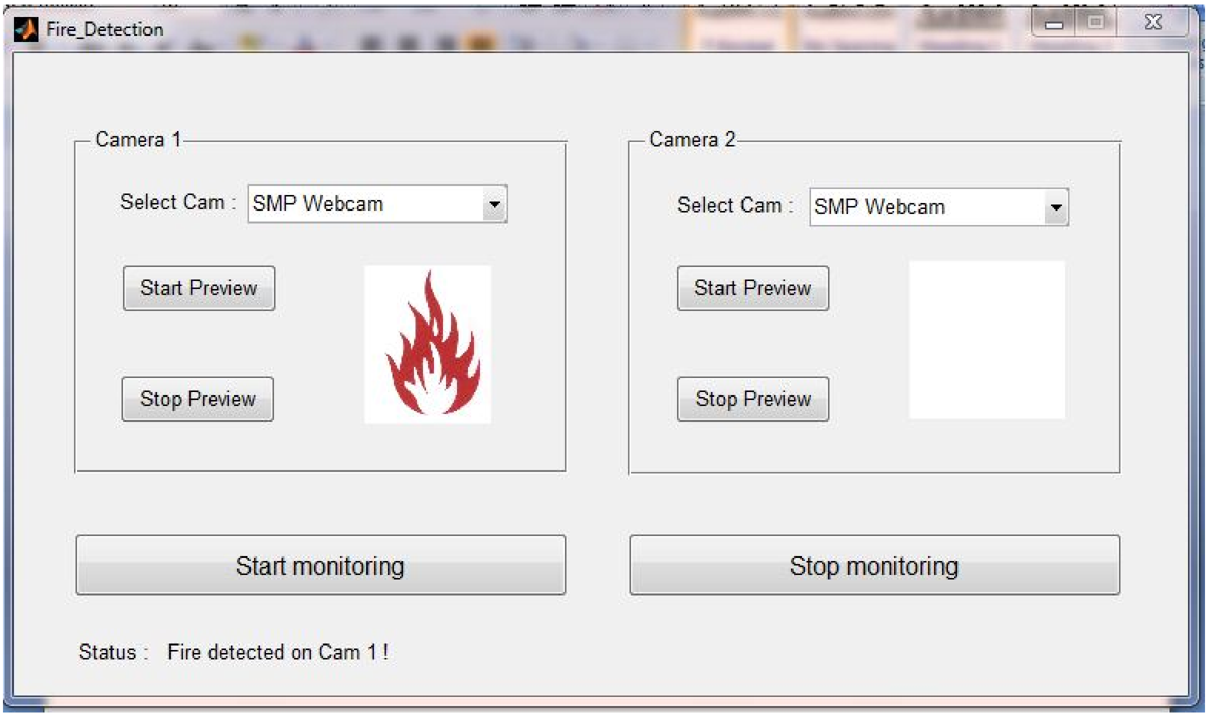


Fig. 6.2.6 Detection of Fire on Cam1

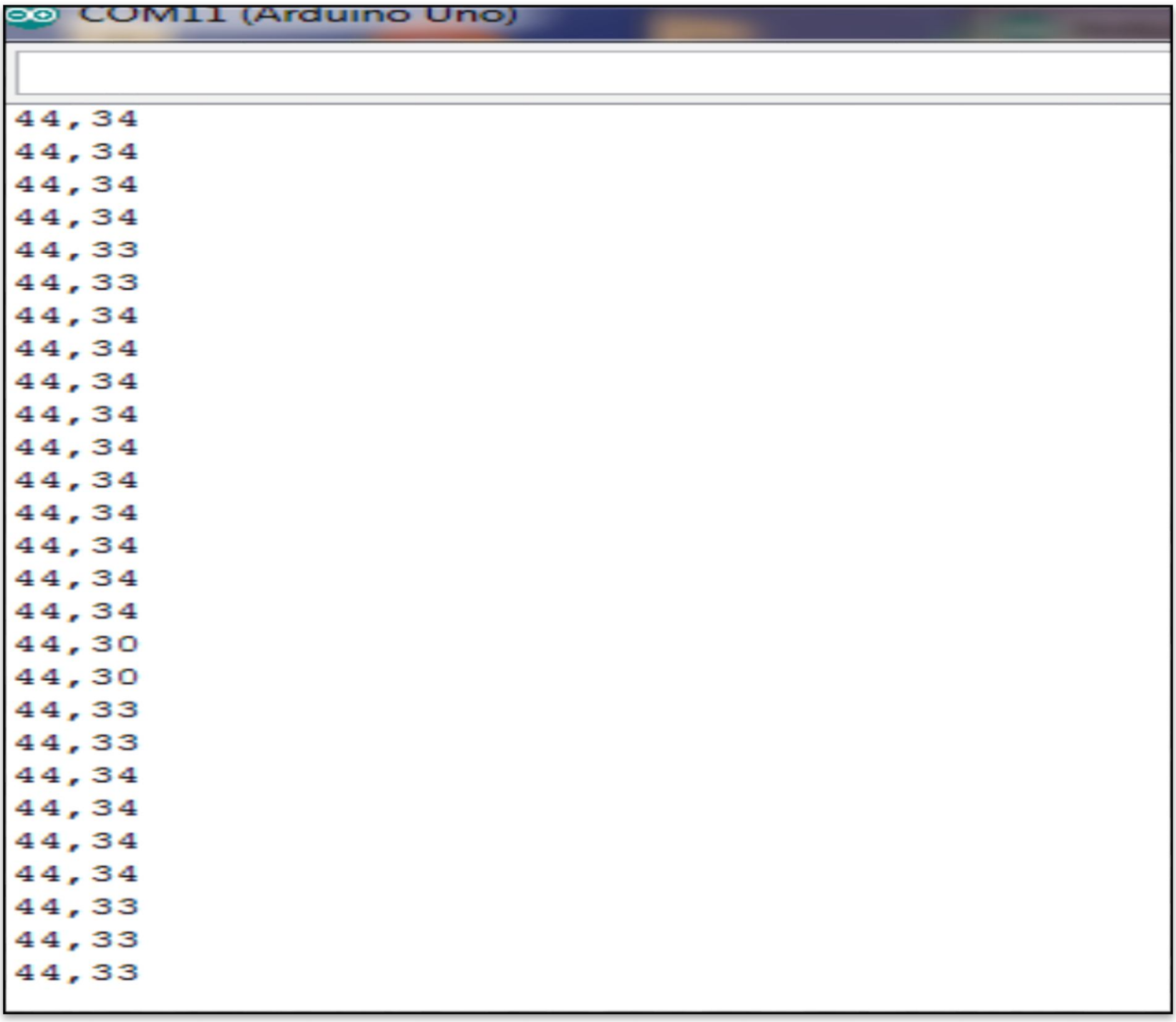


Fig. 6.2.7 Monitoring of Humidity and Temperature using Arduino

Chapter 7

Conclusion & Future scope

Chapter 7

Conclusion & Future scope

7.1. Conclusion

This project, Fire Detection System has been developed using Image Processing and Matlab software. This system has the ability to apply image processing techniques to detect fire. This system can be used to monitor fire and has achieved 90% accuracy for single webcam. The system works on real time, as it extracts frames in every 2 seconds, it provides continuous monitoring. This system has high efficiency as it has incorporated techniques of Area detection, Color detection, Motion detection, and Smoke detection as well as Humidity and Temperature detection. For better performance outcomes use of RGB, HSV and YCbCr color space is made in the detection techniques, as per their suitability, efficiency and properties. The different parameters like threshold value, blind-spots will be handled properly in our future research. Thus application of proposed fire detection system gives us a better system performance in term of less false alarm and thus a higher system performance is achieved.

7.2. Future Scope

For further accuracy use of Neural Networks for decision making can be made and GSM module can also be implemented for sending SMS to nearby fire station in case of severe fire. Water sprinklers can also be incorporated. By research and analysis, the efficiency of the proposed Fire detection system can be increased. The margin of false alarms can be reduced even further by developing algorithms to eliminate the detection of red colored cloth as fire. By proper analysis, suitable location height and length for camera installment can be decided, in order to remove blind-spot areas.

References

1. Mengxin li, Weijing xu, ke xu, Jingjing fan, Dingding hou(2013): Review of fire detection technologies based on video image, Vol. 49, pp. 701-705
2. Vipin V(2012): Image Processing Based Forest Fire Detection, vol. 2, pp. 89-92
3. DHT11 humidity and temperature sensor pdf //http:www.droboticsonline.com
4. Turgay Çelik, Hüseyin Özkaramanlı, and Hasan Demirel, Fire and Smoke Detection without sensors: Image Processing Based Approach, *15th European* Signal Processing Conference (EUSIPCO 2007), Poznan, Poland, September 3-7, 2007,
5. Hemangi Tawade , R.D. Patane (2015): Optimized Flame Detection using Image processing based Techniques, Volume 4, pp. 21
6. Optimized Flame Detection using Image processing based Techniques, Gaurav Yadav et al / Indian Journal of Computer Science and Engineering (IJCSE), Vol. 3 No. 2 Apr-May 2012.
7. Matlab codes://http: www.mathwork.edu.com
8. Abitha T.E., Paul P Mathai, Reducing False Alarm in Vision Based Fire Detection with NB Classifier In EADF, International Journal of Scientific and Research Publications , Volume 3, Issue 3, August 2013
9. Surya T.S., Suchitra M.S., Survey on Different Smoke Detection Technique using Image Processing, International Journal of Research in Computer and Communication Technology,2014