BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE 460 (January 2022)
Optoelectronics Laboratory

Final Project Report

Section: G1 Group: 01

Smoke Detection Using Webcam Based Spectrometer

Course Instructors:

Dr. Muhammad Anisuzzaman Talukder, Professor
Tanushri Medha Kundu, Part-Time Lecturer

Signature of Instructor: ______________________________

Academic Honesty Statement:

IMPORTANT! Please carefully read and sign the Academic Honesty Statement, below. Type the student ID and name, and put your signature. ***You will not receive credit for this project experiment unless this statement is signed in the presence of your lab instructor.***

"In signing this statement, We hereby certify that the work on this project is our own and that we have not copied the work of any other students (past or present), and cited all relevant sources while completing this project. We understand that if we fail to honor this agreement, We will each receive a score of ZERO for this project and be subject to failure of this course."

Signature: ____________ **Full Name: Md. Jarjis Mondal** **Student ID: 1806068**	**Signature:** ____________ **Full Name: Jonaidul Islam Sikder** **Student ID: 1906041**
Signature: ____________ **Full Name: Mehedi Hasan** **Student ID: 1906075**	**Signature:** ____________ **Full Name: Md. Faiyaz Abid** **Student ID: 1906079**

Table of Contents

1 Abstract

Smoke detection is an integral part of the overall fire hazard safety mechanism of any infrastructure. Defence against potential fire hazard must be triggered as early as possible to minimize damage and loss of lives. A smoke detection aims at achieving this objective. In our project we have implemented a webcam-based smoke detection system. It detects smoke and generates warnings accordingly by analysing the optical spectral plot resulting from smoke affecting the intensity of light emitted from a light source, e.g., an LED. We obtained the plot of the optical spectrum by constructing a spectrometer using a webcam and running our python code. When the intensity of light is affected, the code searches for the threshold value required to detect smoke and if it matches, it sends a warning for different levels of smoke hazard.

2 Introduction

Smoke is a suspension of airborne particles and gases. It contains many chemical compounds generated by combustion. The naked eye detects particle sizes greater than 7 µm. [1] The suspension of the visible particles resulting from combustion is visible to our eyes in the form of smoke. The invisible products of combustion are generally referred to as fumes, consisting of invisible gases.

There are very obvious health and fire hazards related with undesirable generation of smoke. Therefore, early detection of smoke is a key part of fire hazard prevention mechanism. Most of the commercially available smoke detectors are ionization chamber type smoke detector, which mainly detect the combustion of the particles invisible to the naked eye. But they are prone to raising false alarms because of this inherent characteristic of theirs. They detect the fumes from the heating elements of appliances and machinery that occur before the generation of visible smoke, by using radioactive isotopes such as ^{241}Am. They fail to detect the early and low-heat smoldering stage of a fire.

Therefore, it is crucial to design the type of smoke detector that can detect smoke in very early stages of its generation. We have proposed and implemented a simple yet effective smoke detector based on the principles of optoelectronics. Smoke can scatter and reduce the intensity of optical radiation. We have constructed a spectrometer to analyze the spectral plot of light. Based on the amount of reduction of intensity from the plot, smoke can be detected. The reduction in intensity for smoke or any other obstacle will not be the same. We have collected data for different obstacles and different levels of smoke. We have used the data to pinpoint a threshold for smoke detection and detected the presence both heavy and light level smoke using our observed data.

3 Design

3.1 Problem Formulation

3.1.1 Identification of Scope

This project's scope includes the design, development, and installation of a smoke detection system based on a spectrometer. Our project uses optoelectronic principles to detect smoke by evaluating the drop in light intensity produced by smoke particles. It entails building a low-cost spectrometer out of widely accessible components like a webcam and a light source, as well as writing Python-based code to interpret the spectrum data and determine smoke detection limits. The system is intended to distinguish between different degrees of smoke density, ranging from light to heavy, and issue appropriate hazard alerts. By focusing on constructing a simple yet efficient detecting mechanism, this setup tackles the constraints of current smoke detectors while providing an effective yet low cost alternative.

3.1.2 Formulation of Problem

Our main challenge is to detect smoke in its earliest stage using the principles of optoelectronics. Smoke particles scatter optical radiation and cause the intensity of its intensity to decrease. However, many other particles or obstacles do the same. Therefore, it is necessary to record as many observations as possible in order to pinpoint an accurate threshold value. There should also be warnings for different levels of smoke.

3.1.3 Analysis

Our initial assumption was that smoke may have caused IR radiation because of being generated from fire. However, research and our observation corroborate the fact that smoke is not any form of electromagnetic radiation and hence it can only scatter other electromagnetic radiations and cause their intensity to alter. We have to construct a spectrometer first in order to get the optical power/intensity

VS wavelength plot for observing the optimum condition when there is no smoke. Then we have to generate smoke and observe the alteration in intensity due to smoke.

3.2 Design Method

In this project, we used a webcam as spectrometer. For any kind of spectrometer, it should detect all kinds of spectrum, even IR filter also. So, removing IR filter was necessary for the project though we worked with visible light spectrum only.

Another task is to attach a grating material to the webcam lens. The grating material will create a spectrum pattern if light is incident on it. This setup should be at an angle with the incident light to analyze the spectrum. To prevent generating wrong readings from any internal light reflection, the whole setup is enclosed inside a box which absorbs any kind of light. So, it is kept inside a box which is colored black inside.

Finally, an LED connected to a battery and potentiometer is used so that it can roughly feed constant light intensity at the webcam. If smoke pass in front of LED, it will scatter the photons and the intensity will be reduced. By analyzing it, we can detect smoke.

3.3 Simulation Model

We have used “Thermino Spectrometer” for calibration purpose. However, the whole smoke detection part was done using Python programming language. The code is given in next section.

3.4 Full Source Code

The PyCharm IDE was used for this project. We collected a Python script and developed on it for our project purpose.

In the next page, the whole code is given and an explanation is also provided.

```python
import cv2
import numpy as np
import matplotlib
print(matplotlib.get_backend())  # Check current
backend
matplotlib.use('TkAgg')  # Example: Set to TkAgg if not
already

import matplotlib.pyplot as plt

# ===Camera Setup===
camera_index = 1  # Index 0 for laptop cam, 1 for
external cam
# Create video capture object
cap = cv2.VideoCapture(camera_index)
# Set video capture parameters
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)  # Adjust
resolution if needed
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

if not cap.isOpened():
    print("Error: Camera not accessible.")
    exit()

calib = -30  # Calibration offset

# Set up plot
plt.ion()  # Turn on interactive mode
fig, ax = plt.subplots()
ax.set_xlabel('Wavelength (nm)')
ax.set_ylabel('Intensity')

# Initialize intensity array and wavelength vector
# intensity_array = []

# Placeholder for intensity data
wavelength = np.linspace(400 + calib, 800 + calib,
640)[::-1]
initial_intensity = np.zeros_like(wavelength)  # Start
with zeros

# Plot the initial data
line, = ax.plot(wavelength, initial_intensity, 'r-')  #
Red line

ax.set_xlim([400, 800])  # Adjusted to accommodate
calib offset
ax.set_ylim([0, 255 * 480 * 1.1])  # Example: Max
grayscale sum

frame_count = 0  # Debugging counter for frames

# Loop to process video frames
while True:
    ret, frame = cap.read()
    if not ret:
        print("Error: Failed to capture frame.")
        break

    frame_count += 1
    print(f"Frame {frame_count}: Captured
successfully.")  # Debug frame capture

    # Convert to grayscale
    gray_image = cv2.cvtColor(frame,
cv2.COLOR_BGR2GRAY)
    cv2.imshow("Live Video Feed (Grayscale)",
gray_image)  # Show grayscale feed
```

```python
    # Calculate intensity (sum of each column)
    intensity = np.sum(gray_image, axis=0)
    intensity_max = np.max(intensity)
    #print(f"Frame {frame_count}: Intensity max =
{intensity_max}, min = {np.min(intensity)}")  # Debug
intensity

    # Update plot data dynamically
    line.set_ydata(intensity)  # Update y-data
(intensity)
    ax.set_ylim(0, intensity_max * 1.1)  # Adjust y-
axis dynamically

    # Find the index of the wavelength closest to 451
nm
    wavelength_target = 457
    index = np.abs(wavelength -
wavelength_target).argmin()
    intensity_at_457 = intensity[index]
    print(f"Intensity at wavelength 457 nm:
{intensity_at_457}")

     #Reference light intensity 7200-7400

    # Update plot title based on intensity thresholds
    if ((intensity_at_457>=4000) and (intensity_at_457
<= 5000)):
        plt.title("Heavy Smoke Detected!")
    elif ((intensity_at_457>=6000) and
(intensity_at_457 <= 6550)):
        plt.title("Light Smoke Detected!")
    else:
        plt.title("Intensity Plot")

    plt.draw()  # Refresh the plot
    plt.pause(0.01)  # Allow time for the plot to
update

    # Break the loop if 'q' is pressed
    if cv2.waitKey(1) & 0xFF == ord('q'):
        print("Exiting loop.")
        break

# Release resources
cap.release()
cv2.destroyAllWindows()
plt.ioff()  # Turn off interactive mode
plt.show()
```

Table: Source Code for the main program

This Python script uses **OpenCV**, **Matplotlib**, and **NumPy** to capture live video frames from a camera, process them, and dynamically update a plot of intensity data across wavelengths. Here's a breakdown of its functionality:

Library Imports

- cv2: For handling video capture and image processing.
- numpy: For numerical operations like array manipulation.
- matplotlib: For plotting data (uses the TkAgg backend for better compatibility with GUI environments).

Camera Setup

Camera Index:

- camera_index = 1: Indicates which camera to use (e.g., 0 for built-in, 1 for an external camera).
- cap = cv2.VideoCapture(camera_index): Opens the video capture for the specified camera.

Frame Dimensions:

- Sets the frame resolution to **640x480 pixels** using cv2.CAP_PROP_FRAME_WIDTH and cv2.CAP_PROP_FRAME_HEIGHT.

Error Handling:

- If the camera fails to open (cap.isOpened() returns False), it exits the script.

Wavelength and Plot Initialization

Calibration Offset:

- calib = -30: Adjusts the wavelength range for calibration.

Wavelength Array:

- wavelength = np.linspace(400 + calib, 800 + calib, 640)[::-1]:
  - Creates a linearly spaced array of 640 wavelengths (one for each pixel column in the frame) in the range 400–800 nm (adjusted by calibration).

Initial Plot:

- A live plot is set up with plt.subplots():
  - X-axis = Wavelength (nm).
  - Y-axis = Intensity (sum of pixel values per column).
- The initial data is plotted as a red line ('r-').

Video Frame Processing

Frame Capture:

- The script continuously captures frames using cap.read().
- It converts each frame to grayscale with cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY).

Intensity Calculation:

- intensity = np.sum(gray_image, axis=0) computes the sum of pixel values along each column (representing intensity at a specific wavelength).

Dynamic Plot Updates:

- The plot's Y-data (line.set_ydata) and Y-axis limits (ax.set_ylim) are updated in real-time based on the calculated intensity.

Target Wavelength Analysis:

- **Target Wavelength**: 457 nm. (blue light peak wavelength)
- index = np.abs(wavelength - wavelength_target).argmin():
  - Finds the index of the wavelength closest to 457 nm.
- intensity_at_457: Retrieves the intensity at that index.

Smoke Detection:

- Based on intensity_at_457, the plot title updates to indicate the smoke level:
  - **4000–5000**: Heavy smoke detected.
  - **6000–6550**: Light smoke detected.
  - Otherwise: Default title (No smoke detection triggered)

Displaying Frames

- The grayscale frame is displayed with cv2.imshow().
- The loop continues until the user presses **'q'** (cv2.waitKey(1) & 0xFF == ord('q')).

Cleanup

- cap.release(): Releases the camera.
- cv2.destroyAllWindows(): Closes any OpenCV display windows.
- plt.ioff() and plt.show(): Finalizes and displays the plot.

4 Implementation

4.1 Description

We have used an LED as our light source. The light emitted from the LED passes through the slit made in the box. The light waves are incident on a Compact Disc (CD) used as an inexpensive diffraction grating. Through constructive and destructive interferences, the light waves create a diffraction pattern on the CD. The CD is attached to a webcam with its IR filter removed for capturing even IR waves if needed. The diffraction pattern is captured by the webcam and sent to a laptop computer for being processed. The webcam data is analyzed by the code we have written in Python which generates the optical intensity VS wavelength plot. Next the change in intensity is recorded for different kinds of obstacles including folded paper sheet, heavy smoke and light smoke.

4.2 Hardware Setup

In order to analyze the visible light spectrum, we converted a regular webcam into a working spectrometer for this project. A number of adjustments and considerations were made in order to accomplish this. Removing the webcam's infrared (IR) filter was one of the crucial measures. The removal of IR filters was essential for this project, even though they are normally used to block infrared radiation and enhance image quality for common camera applications. Even though we only looked at the visible spectrum in this work, a spectrometer needs to be able to detect all sections of the spectrum, including light components that could otherwise be filtered out.
Adding a grating substance to the webcam's lens was another crucial change. This substance disperses incoming light into its component wavelengths, which is crucial in forming the spectrum pattern. The grating needs to be angled with respect to the incident light in order to work properly and allow for spectrum analysis.

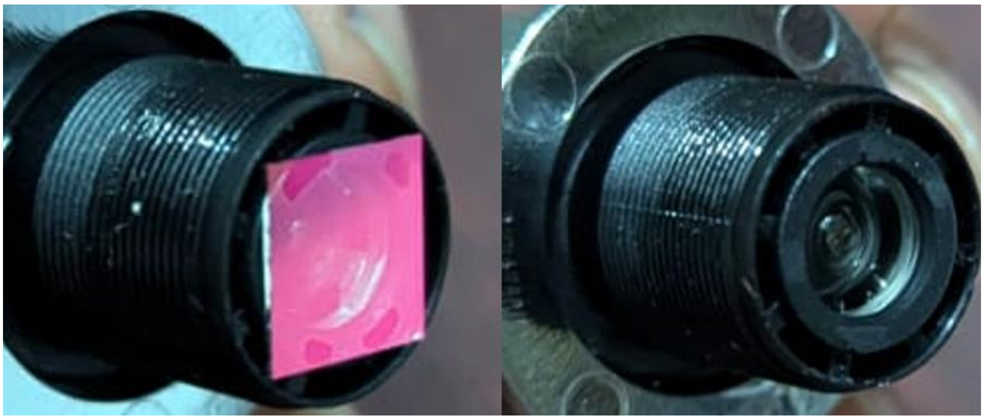


Figure 1: Removing IR filter

The complete equipment was housed in a specially made enclosure to guarantee precision and avoid interference from internal reflections or stray light. Spectral analysis was conducted in a controlled environment by painting the inside of the box black to block out any unwanted light. This enclosure preserved the integrity of the spectrometer's data collection and reduced the possibility of inaccurate readings.

An LED was incorporated into the arrangement to provide a steady and dependable light source. With the use of a potentiometer and a battery, the LED's brightness could be precisely adjusted. This made it possible to maintain an approximately consistent light level throughout the studies. Smoke reduced the amount of light that reached the camera by causing photon scattering when it passed in front of the LED. The system could successfully identify the presence of smoke by examining these variations in intensity.

To ensure all the components are sturdy and attached firmly, they are glued with the box and the box opening is closed with tape so that other lights cannot pass through.

A small slit is kept in front of the box so that only the LED light can pass through it. However, the webcam is not placed directly at the slit as it will blind the display with bright light and no spectrum can be observed. So, the webcam is kept at an angle (45°) with the slit. The following figure shows a schematic of our setup:

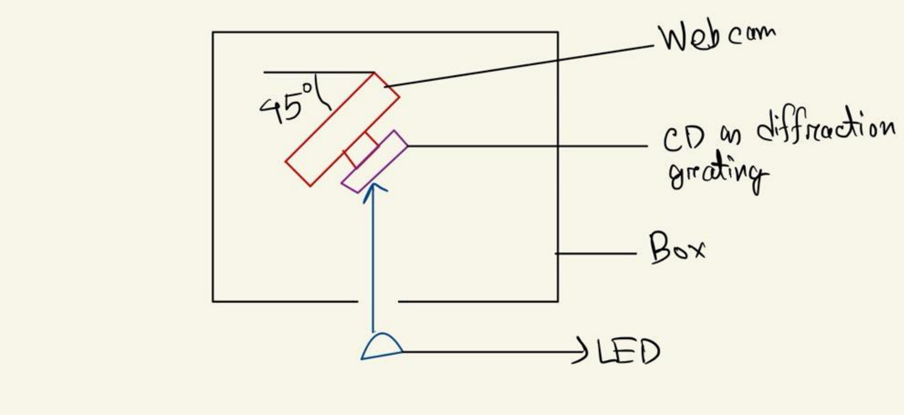


Figure 2: Schematic of setup

All things considered, this creative configuration fused optical physics with useful engineering to produce a small, effective spectrometer that can analyze variations in light intensity for uses such as smoke detection.

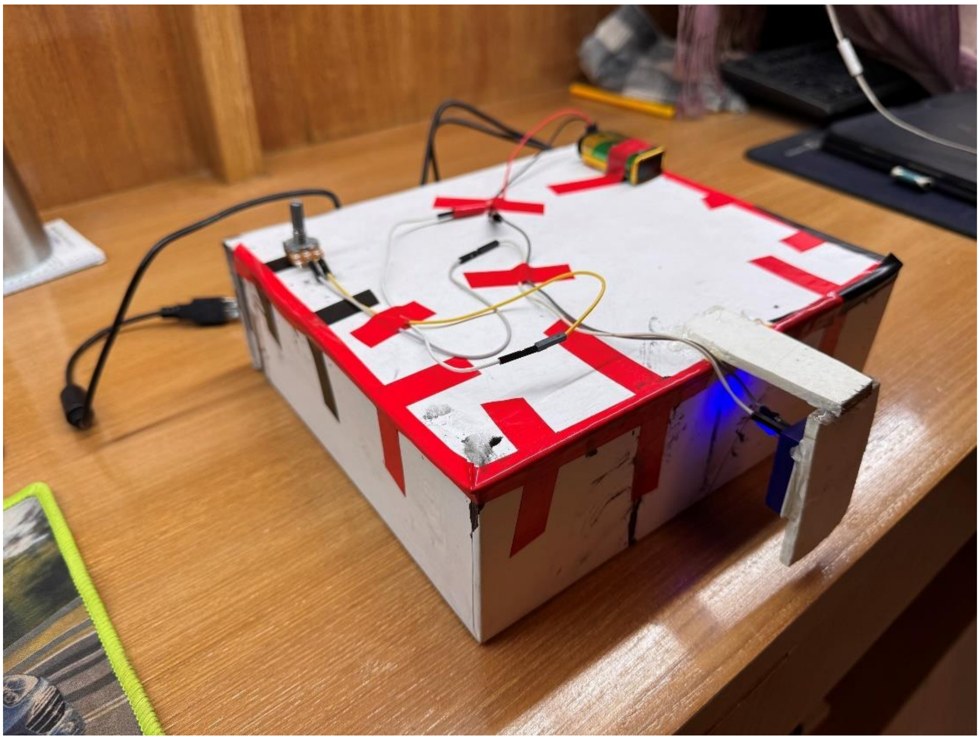


Figure 3: Complete Setup of the Project

4.3 Calibration

We calibrated our spectrometer using Red and Blue LED. Though the calibration process is not necessary for smoke detection, we still did it for precision. The calibration was done both in Thermino Spectrometer software and Python.

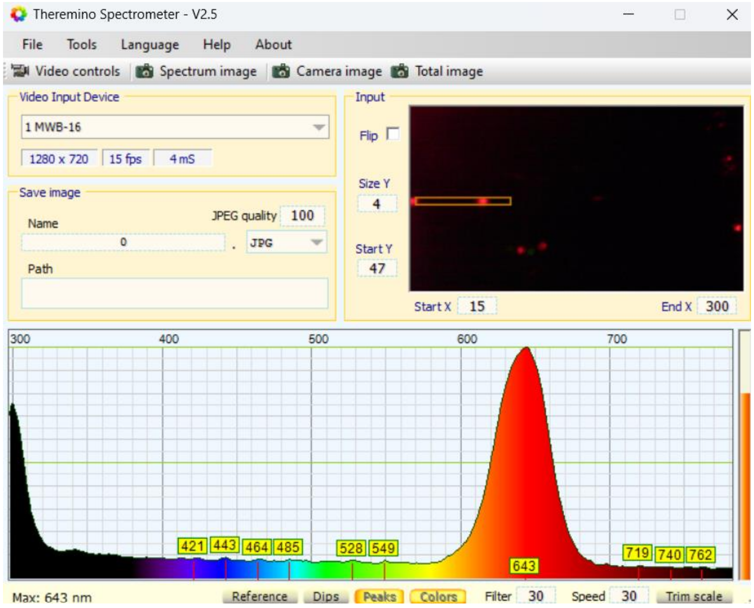


Figure 4: Calibration of spectrometer using red LED in Thermino

A typical red LED will give peak at 650nm. [2] Here, in Thermino, it shows that it gives peak at 643 nm, which is quite impressive and accurate for a typical homemade spectrometer using webcam and CD as grating.

Similarly, blue LED gives peak at 450 nm [2], this was also seen using Thermino and Python.

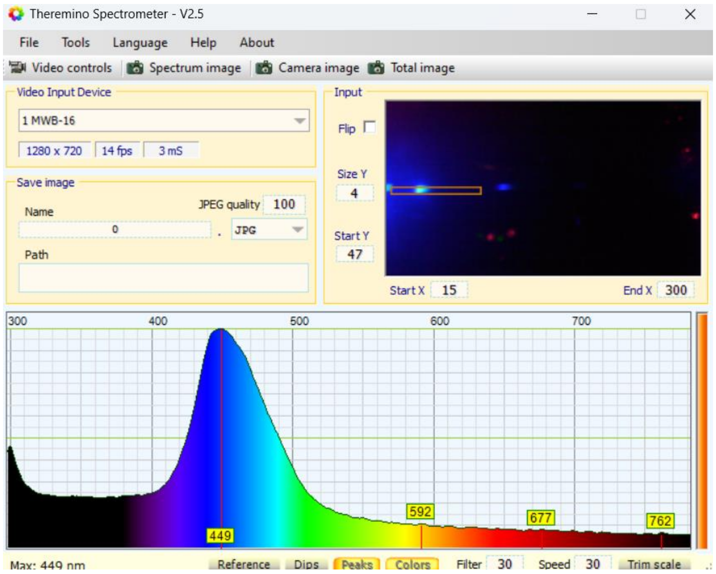


Figure 5:Calibration of spectrometer using blue LED in Thermino

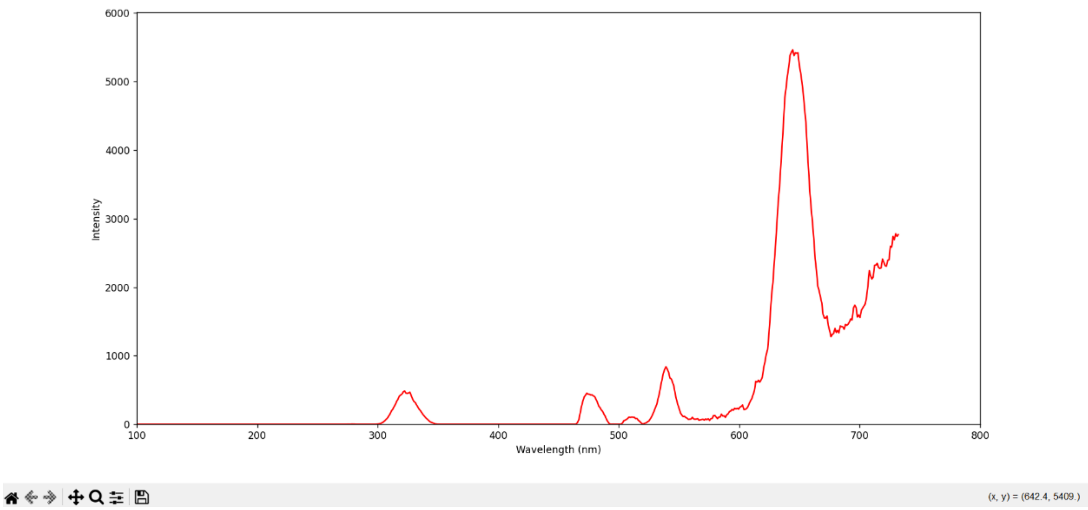


Figure 6:Calibration of spectrometer using red LED (Python)

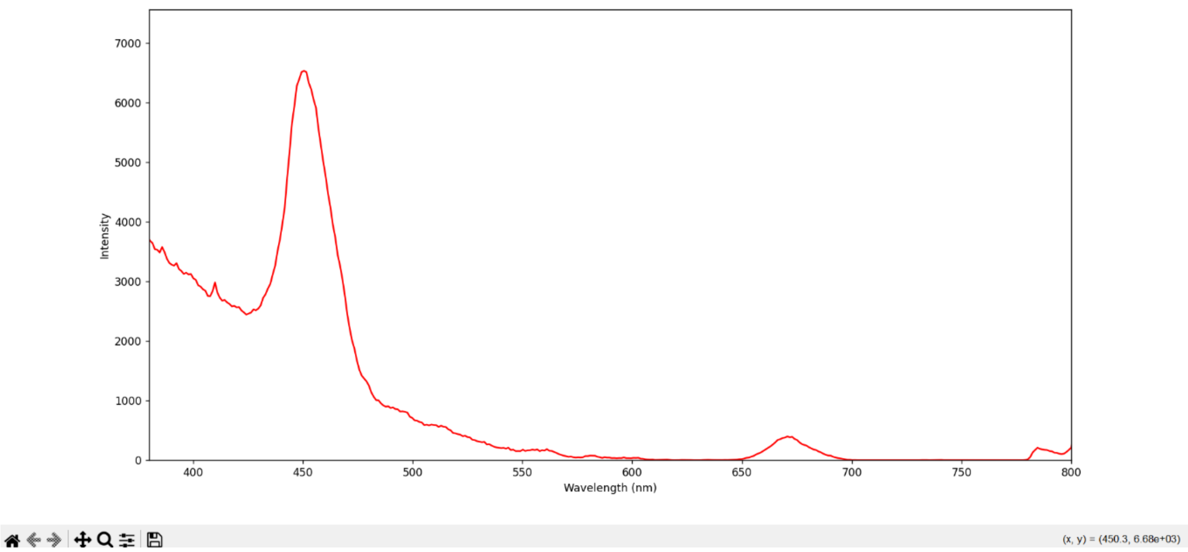


Figure 7:Calibration of spectrometer using blue LED (Python)

4.4 Threshold Selection

The most important part of our project is selecting the threshold at which it will trigger smoke detection. We classified two types of smoke according to their density, light and dense. The reasoning was light and dense smoke varied the intensity differently. We worked with various kinds of smoke and stored the intensity of smoke in array.

We also picked the reference intensity carefully. We used blue LED which gives a peak at 457nm. We varied the reference intensity as 3000, 5000, 7000 etc. Analyzing various intensity, we found that the best intensity at which the smoke can be detected and other objects do not trigger smoke alarm is 7100 to 7500. So, we used that as reference. One thing to mention here is that the intensity value is unitless as it is the sum of the grayscale image pixel column. To ensure the intensity stays within this range, we used a 10k ohm potentiometer to tune the LED current.

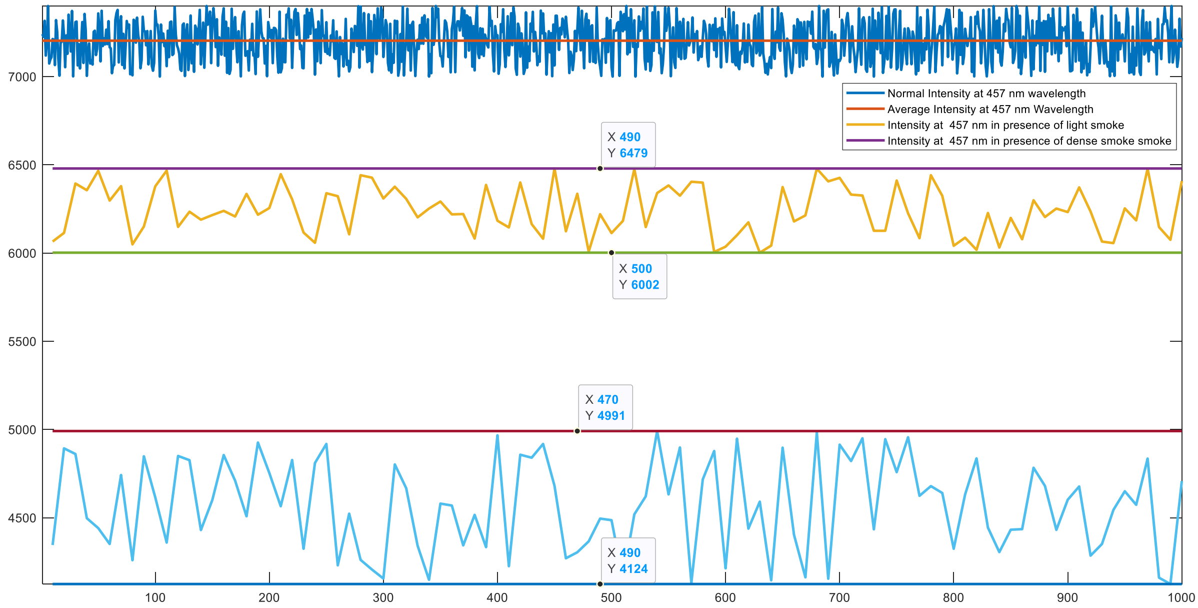


Figure 8: Plot for threshold selection

After generating arrays with different types of smokes, we plot it on MATLAB, observed the maximum and minimum value of intensity during smoke, and finally chose the intensity threshold carefully. If the reference light intensity is between 6000 and 6550, it will trigger light smoke detection. And if it is between 4000 and 5000, it will trigger heavy smoke detection.

4.5 Failsafe Mechanism

One of the key aspects of our project is failsafe mechanism. Here, we tested with various objects like dust dirt, plastic bag, tissue paper, translucent papers to check whether it triggers smoke detection or not. We found that they do not trigger at all or very rarely. These testing were also too much detrimental for threshold selection.

5 Design Analysis and Evaluation

5.1 Novelty

- Focusing on changes in light intensity rather than measuring scattering or absorption.
- Integrating a code written in Python for spectral analysis and smoke warning generation, hence increasing compatibility with IoT (Internet of Things) systems.
- Eliminating the need for a radioactive isotope like a conventional smoke detector and hence reducing cost and increasing safety.

5.2 Design Considerations

5.2.1 Considerations to public health and safety

- Detection of smoke in the earliest stage in order to minimize risk of fire hazard as much as possible.
- Eliminating the need for radioactive isotopes.

5.2.2 Considerations to environment

- Implementation of the project using a low power requiring equipment.

5.2.3 Considerations to cultural and societal needs

- Creating social awareness regarding fire safety.
- Minimizing damage from fire accidents.

5.3 Investigations

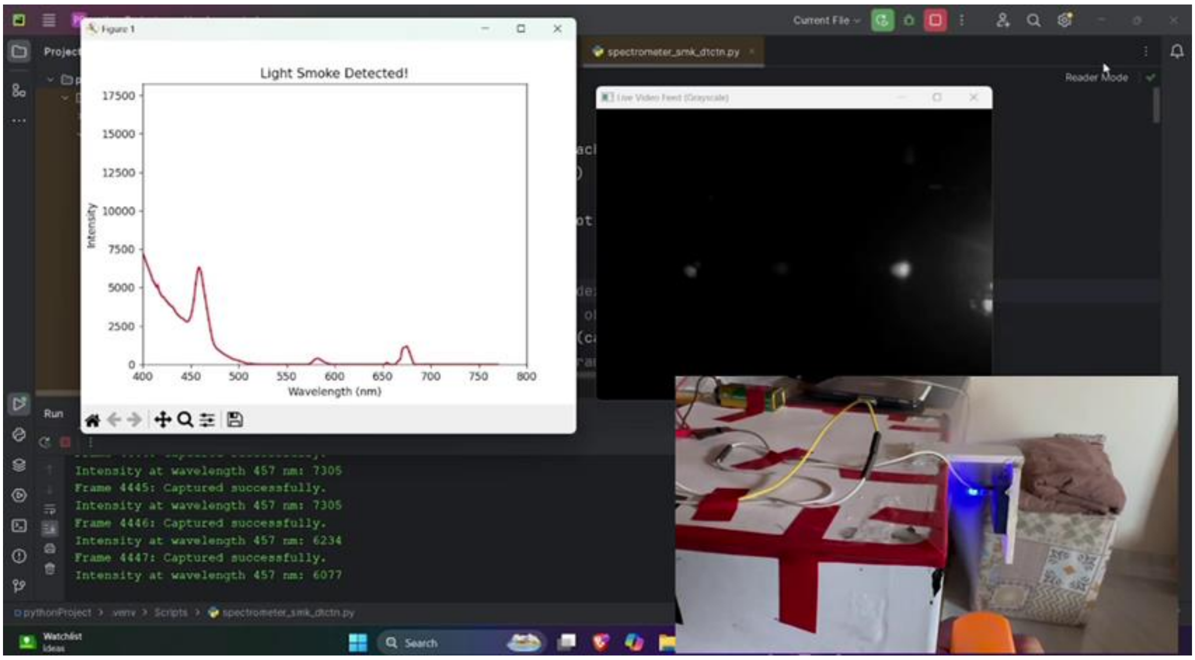
5.3.1 Data Collection

All the data was collected via common objects. Smoke was generated using burning dry leaves, foams, E-cigar, Air freshener. Then the intensity data was collected at the 457nm wavelength, as at this wavelength blue LED gives peak intensity. Then the threshold was selected using plot which was described in 4.4.

5.3.2 Results and Analysis

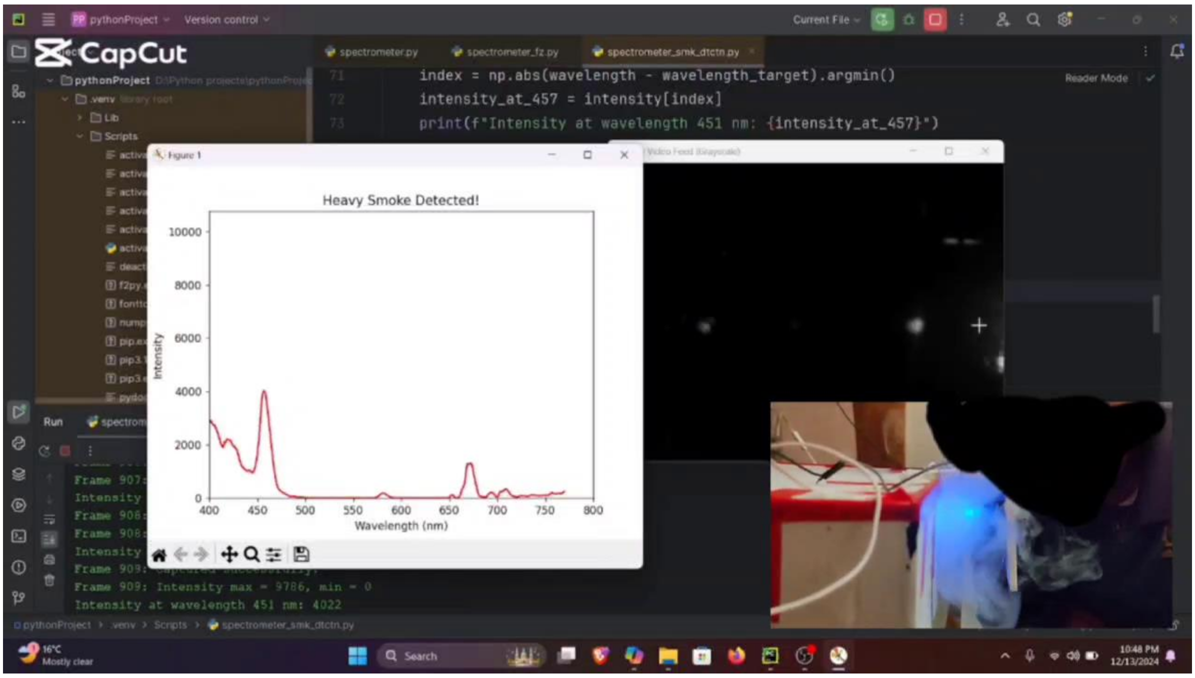
Result for light smoke (burning leaves):

Result for light smoke (air freshener fume):



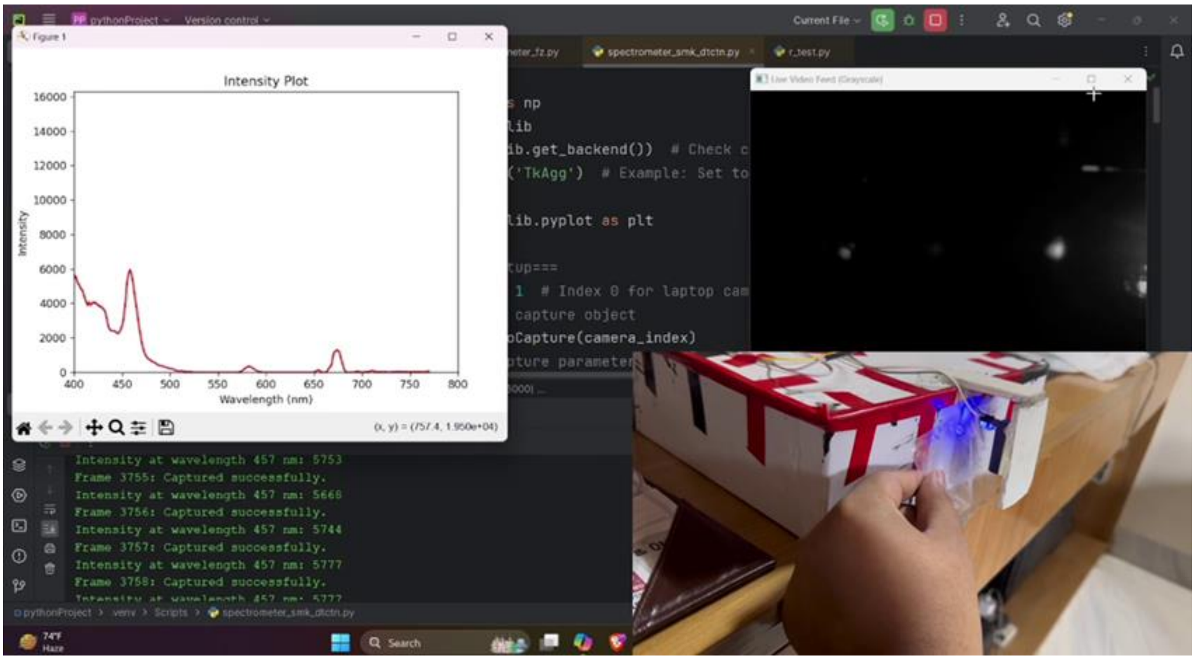
The fume emitted from the air freshener is quite thin. As we selected the threshold carefully, it detects as light smoke.

Result for dense smoke (E-cigar):



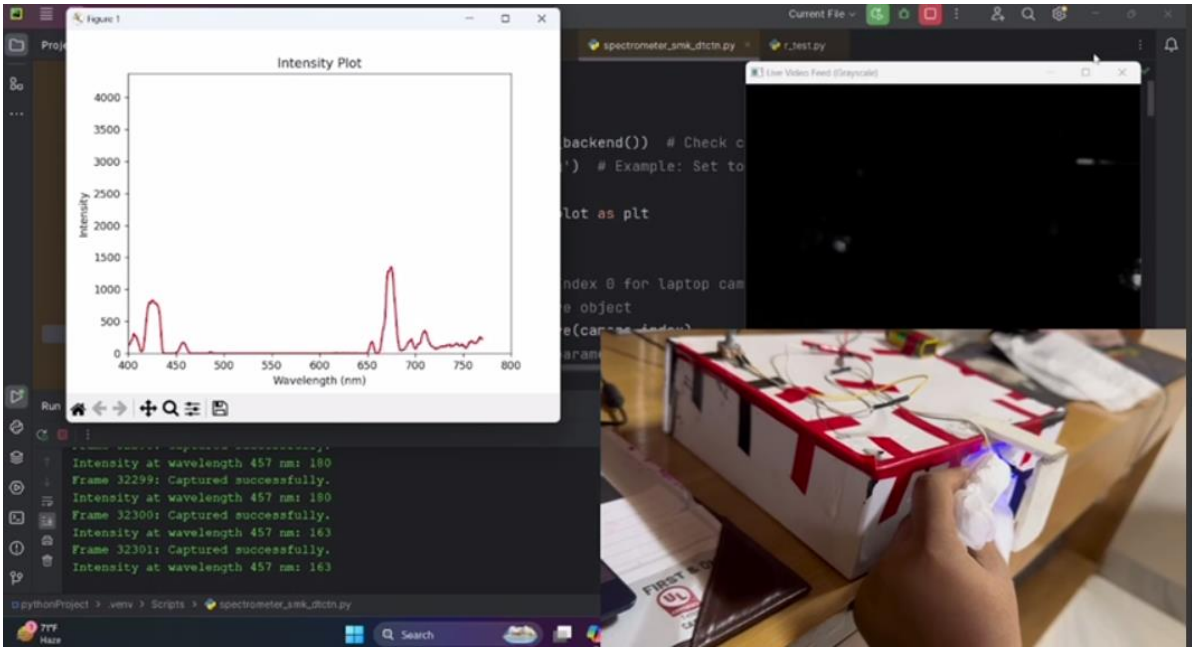
The fume emitted from the e-cigar is quite dense. As we selected the threshold carefully, it detects as dense smoke.

Testing with plastic transparent bag whether it gives false alarm or not:



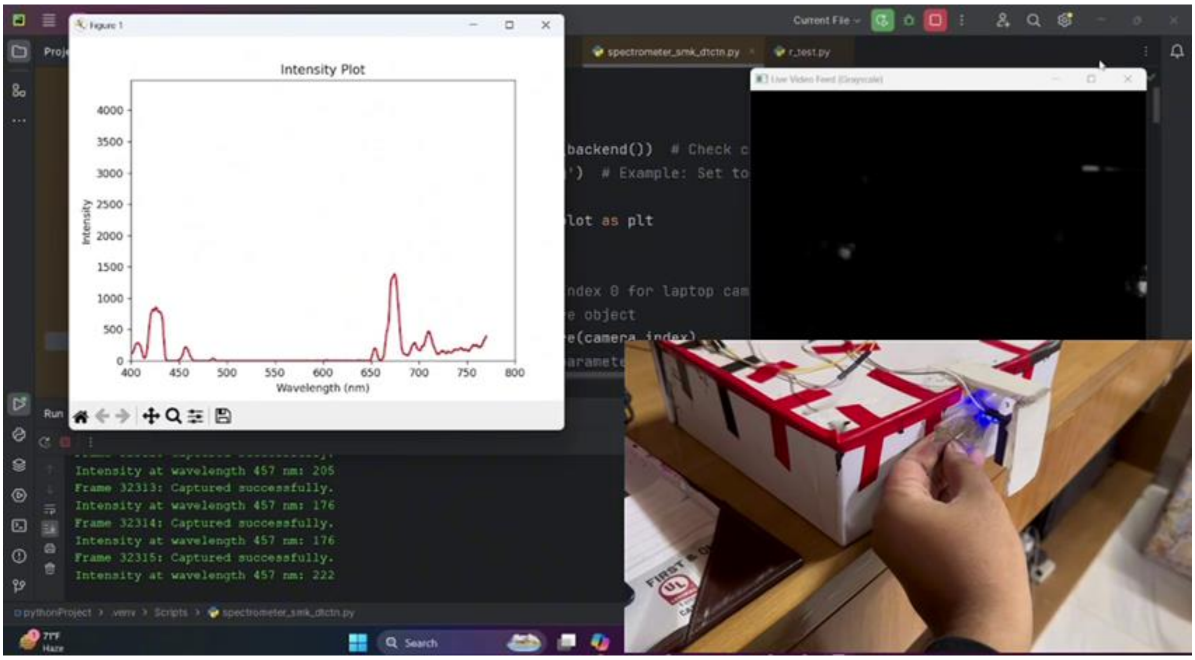
As we can see, it does not trigger as the intensity because of this out of the smoke threshold value

Testing with tissue paper whether it gives false alarm or not:



As we can see, it does not trigger as the intensity because of this out of the smoke threshold value

Testing with bag filled with dust and dirt whether it gives false alarm or not:



As we can see, it does not trigger as the intensity because of this out of the smoke threshold value

5.3.3 Interpretation and Conclusions on Data

According to the demonstrations, our device detects not only smokes, but also their type. It does not trigger for other objects that we tested.

5.4 Limitations of Tools

- Inconsistent behavior of LED light source. For example, the blue LED we have used, after prolonged usage, emits light in the violet or even the UV region because of being worn out.
- Improper diffraction patterns if CD (Used as a cheap alternative to a diffraction grating) is burned already before purchase.
- Nonideal behavior of webcam. It produces grainy images
- Some internal reflection
- Dust particles tampering with light intensity.

5.5Impact Assessment

5.5.1 Assessment of Societal and Cultural Issues

- This project will create social awareness regarding fire safety.

5.5.2 Assessment of Health and Safety Issues

- Smoke is the leading cause of death in fire accidents. This project helps eliminate the overall risk factor involved with smoke hazards leading to potential fire hazards.

- This project will help damage from fire accidents.

5.5.3 Assessment of Legal Issues

- This project will help fortify the need for stricter fire hazard protocols through legal and legislative procedures.

5.6 Sustainability Evaluation

5.6.1 Assessment of Societal and Cultural Issues

- This project will create be sustainable in the long run since in our daily lives, it is impossible to ignore fire hazards.

5.6.2 Assessment of Health and Safety Issues

- No harmful chemicals or devices have been used to materialize the project. Hence it will pose no threat to public health.

5.6.3 Assessment of Legal Issues

- Since no hazardous chemicals or components have been used in this project, it shall be compliant with evolving environmental regulations in the distant future as well.

5.7Ethical Issues (PO(h))

We had to come up with authentic ideas for materializing a smoke detector based on the principles of optoelectronics. We already had the idea of how to construct an inexpensive DIY-type spectrometer from prototypes that were not copyrighted. For projecting the spectral output correctly from the diffraction pattern, we had to look up in an online forum for correctly calibrating our Python code.

6 Reflection on Individual and Team work (PO(i))

6.1 Individual Contribution of Each Member

Name	ID	Contribution
Jarjis Mondal	1806068	Failsafe Mechanism, Calibration in Python
Jonaidul Islam Sikder	1906041	Hardware Setup, Thermino Calibration
Md Mehedi Hasan	1906075	Testing with smoke, IR Filter Removal
Md Faiyaz Abid	1906079	Code development, Threshold Selection

6.2 Log Book of Project Implementation

Date	**Milestone achieved**	**Individual Role**	**Comments**
01 December	Hardware Setup Completed	1906041	Hardware setup completed
02 December	Python code written, LED tested	1906079, 1906075, 1906041, 1906068	There are huge discrepancies in the plot. The white LEDs purchased have been burnt, one white LED gave two peaks, the blue led shows peak emission in UV region
12 December	Python code corrected	1906079	Peaks are found at correct positions

13 December	Project tested with paper and different levels of smoke	1906079, 1906075, 1906041, 1906068	Changes in optical power intensity are observed and threshold values are set
15 December	Project tested again	1906079, 1906075, 1906041, 1906068	Correct detection of smoke

7 Communication to External Stakeholders

7.1Executive Summary

A revolutionary smoke detection device was created utilizing a cheap camera and optoelectronics. Unlike standard smoke detectors, which rely on radioactive materials or fundamental optical scattering, this device uses light intensity variations to detect smoke in its early stages. The device recognizes different smoke levels and produces timely alerts by combining a low-cost spectrometer with a camera and Python-based software. This environmentally friendly and cost-effective technology promises to improve fire safety in homes, businesses, and enterprises by providing accurate detection while eliminating false alarms. This breakthrough offers a safer and more intelligent future for fire risk prevention.

7.2 User Manual

- Insert the device in a potential fire hazard environment compliant with NEPA and BNBC standards for fire risk prevention.
- Protect the device from the contaminating elements of its surrounding environment.

8 Project Management and Cost Analysis

Material Used	No. of Units Used	Unit Price (BDT)	Total Price (BDT)
Webcam	1	1,200	1,200
LED	6 (5 White, 1 Blue)	4	24
Breadboard	1	100	100
Potentiometer	1	24	24
Total Cost			1,348

9 Future Work

1. Integration with advanced Aspirating Fire Detector System for more precise detection of smoke.
2. Integration of chemical analysis of smoke particles.
3. Miniaturization
4. Integration of gas sensors, temperature sensors and motion sensors
5. Integration with IoT, embedded systems for real time distant monitoring and cloud community services
6. Open source development

10 References

1. https://www.sciencefocus.com/the-human-body/how-small-can-the-naked-eye-see
2. https://www.researchgate.net/figure/ntensity-vs-wavelength-of-light-emitting-diode-LED-with-5-mm-and-emission-colors-red_fig3_356263224