The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

Author Name	sanjana
Title	report
Paper/Submission ID	566814
Submission Date	2022-07-22 16:30:39
Total Pages	21
Document type	Dissertation

Result Information

Similarity **0 %**

1 10 20 30 40 50 60 70 80 90

Exclude Information

Quotes	Excluded
References/Bibliography	Excluded
Sources: Less than 14 Words Similarity	Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

0	0	A	A-Satisfactory (0-10%) B-Upgrade (11-40%) C-Poor (41-60%) D-Unacceptable (61-100%)
SIMILARITY %	MATCHED SOURCES	GRADE	

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE

Abstract

Air pollution is a growing concern that goes beyond local, regional and continental boundaries. Everyday change in environmental conditions affects health of human, animals and disturbs the balance of ecology. As the quantity of pollutants are increasing in the environment, it increases the posed risk also. These pollutants can be of many types and arises from different sources of our everyday life without us being aware of it. Hence, it is important for people to be aware of conditions. Digital cameras are the affordable and easily available devices at public places. We can use digital images for processing data from them. We here use the property of difference in scattering, absorbing and reflection of RGB light colours in presence and absence of pollutants. With the difference we can somewhat measure the scattering level and get analyse pollution level from a digital image.

1. INTRODUCTION-AIR POLLUTION

Earth’s atmosphere is a mixture of gases like Oxygen, Carbon dioxide, nitrogen, methane and other gases in definite amount. When quantity of these gases is imbalanced maybe due to natural or human activities causes harm to living organisms. This phenomenon is known as air pollution. There are two types of pollutants-Primary and Secondary. Primary pollutants such as fossil fuels which enters directly into the atmosphere. And secondary pollutants are formed in atmosphere from primary pollutants itself. Photochemical smog constitutes of different secondary pollutants like nitric acid, peroxyacyl nitrates and ozone. Many meteorological factors such as wind, sunlight and temperature are affected by pollution and also cause changes in the pollution in turn.

Air pollution affects a person’s health adversely. Asthma, cardiovascular diseases, lung damage, breathing problems are some of the long-term effects of air pollution. According to WHO, nine out of ten people breathes polluted air in today’s world and this causes death of 7 million people each year. It also causes greenhouse effect, acid rain, damages buildings, crops and reduces eye visibility. Urban areas are worst affected by air pollution where most of industries, power plants and vehicles are located.

Different types of particles follow different trend in their concentration in various seasons. For an instance, fine mass concentration is said to be highest in summer season than in winter season. Similar case is seen in Sulphates and opposite in Nitrates. These concentrations are regularly kept in check by particle and visibility programs such as IMPROVE and NWA in USA. In India, NAQM (National Ambient Air Quality program) monitors Central Pollution and mobile applications like SAFAR INDIA are used to monitor in Ahmedabad.

2. TYPES OF APPROACHES

Three basic types of approaches used to detect air pollution are based on Machine learning, deep learning and image processing approach.

1. Machine learning approach

Machine learning is used to detect and predict air pollution in images. This primarily consists of phases-training and testing. Firstly, a lot of data is trained from set which can in form of temperature, wind, dewpoint, pressure and PM2.5. In last phase, it is provided with different inputs and tested out for increasing its accuracy. Extreme learning method (ELM) and online sequential multiple linear regression (OSMLR) are the most successful in forecasting using machine learning model.

2. Deep learning approach

Deep learning is a subset of machine learning which is inspired by functions and structure of human brain called artificial neural network [1]. It is considered as the best approach for it as it can cover a large range of are and give the most accurate level of pollution. Its only disadvantage is cost of installation is very high. LSTM (Long short-term memory) and STDL (Spatiotemporal deep learning) are famous method for future prediction of air pollution.

3. By image processing

Image processing is a process of transforming an image into a digital image to perform function on it or to extract useful information from the image. Various algorithms to calculate AQI (Air Quality Index) are based on image recognition technology. Though this technique is only able to detect the pollution, it is an inexpensive method and therefore can be used in mass or can be easily used on electronic devices such as Smartphones.

3. DIGITAL IMAGE

A digital image which is stored as bitmap in computer's memory is a matrix of pixels. Each pixel contains 3-bytes values that determines the value of that pixel. Red, blue and green are chosen as three primary colours to make a colour. Each byte can have 266 values which is in range of 0 to 255. Though this range constitutes only 0.39% of the total colours but these are the one that are visible to the human eyes. Thus, all three bytes can form a combination of between this range to form a particular colour. This way total of 16.8 million combinations are possible to form in a pixel.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & f(0,2) & \ldots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \ldots & f(1,N-1) \\ . & . & . & & . \\ . & . & . & & . \\ . & . & . & & . \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \ldots & f(M-1,N-1) \end{bmatrix}$$

Figure 1- Digital image as a matrix

Tonality in a digital image for 24-bit RGB colour is represented by number between 0 and 255. Point 0 represents black while 255 represent white here. There can be different types of digital image such as binary, black and white, 8- or 16-bit colour format images.

4. LIGHT AND ITS VISABILITY

Light is a type of energy that travels in form of wave which is of vibrating electric and magnetic fields. It travels in straight direction until it meets any object in between. The collision of light particle with any other particle can result in any of these phenomenon- absorption, reflection, scattering or transmission of light.

If light wave gets hit by any other particle which have less wavelength and very more frequency, then it leads to reflection. Since all colours of light are reflected in the same direction, it does not scatter and reflected wave appears white. In case when light collides with larger wavelength particles, it absorbs some light and some of the light is refracted. This phenomenon is also known as Scattering. The degree of scattering that occurs in light depends upon frequency and wavelength of light. Light with long wavelength and low frequency are scattered least and short wavelength are scattered most. Nitrogen and Oxygen present in atmosphere in abundant amount are actually most effective to scatter higher frequency lights in photon. Scattering of light and absorption degrades the visibility.

In a digital image, red colour will be the least scattered and blue will be the most scattered colour. Behaviour of light as particle and wave affects the colour properties of light. Since

Red, Green and Blue are the most primary colour in any digital image, it is important to know their wavelengths which is 0.65, 0.55 and 0.45 microns respectively.

Electromagnetic waves carry their energy in form of small and discrete packets known as photons. When a light falls on any object, some colours of photons are captured by molecules on that object. The colour of photons then reflected by that object captured determines the colour which we see in that object. For example, a red shirt looks blue to our eyes because the dye molecules in shirt have absorbed other wavelength colour from light and now only red colour light is reflected. Therefore, those looks red to us.

Figure 2- Visibility of colours to human eyes

5. SCATTERING

Scattering can occur due to any of these cases-diffraction, refraction or phase shift. Depending upon the size of particle and light, there can be 3 cases:

1. When particle is very small, then amount of scattering that will occur will be negligible.
2. When particle is much more than light then more scattering occurs in forward direction than in backward. Hence, scattering efficiency is maximum when size of light and the particle are almost same i.e, 1:1. At that point it can scatter, five times more than the size of particle. This phenomenon is responsible for haze colour in sky.

Due to Rayleigh scattering, which is more effective at shorter ends, so blue end of spectrum is visible and sky appears to blue.

3. When particles are of same size as that of light then photons scatters equally and haze appears grey or white.

Other important components in atmosphere that affects visibility are Carbon dioxide and Nitrogen oxide. Carbon dioxide absorbs all wavelength thus causes least scattering of light. And Nitrogen oxide can absorb blue colour thus it makes appearance dark brown.

6. MEASUREMENT METHODS

6.1 VARIANCE, COVARIANCE AND COVARIANCE MATRIX

Variance is measure is spread between data in matrix. Mathematically, it is average of squared deviation from mean value of the set. For a digital image, it can be calculated as follows-

$$\sigma_{R'}^2 = \frac{\sum_{c=0}^{c-1}\sum_{r=0}^{r-1}\left(R'_{cr} - \overline{R'}\right)^2}{RC-1}$$

$$\sigma_{G}^2 = \frac{\sum_{c=0}^{c-1}\sum_{r=0}^{r-1}(G_{cr} - \overline{G})^2}{RC-1}$$

$$\sigma_{B}^2 = \frac{\sum_{c=0}^{c-1}\sum_{r=0}^{r-1}(B_{cr} - \overline{B})^2}{RC-1}$$

Covariance is measurement of ability to retain its value when other values are linearly transformed. Positive covariance signifies those previous values are larger than later ones and negative covariance shows large values are associated with the smaller values. It can be calculated as follows-

$$\sigma_{R'G} = \frac{\sum_{c=0}^{c-1}\sum_{r=0}^{r-1}\left(R'_{cr} - \overline{R'}\right)(G_{cr} - \overline{G})}{RC-1}$$

$$\sigma_{R'B} = \frac{\sum_{c=0}^{c-1}\sum_{r=0}^{r-1}\left(R'_{cr} - \overline{R'}\right)(B_{cr} - \overline{B})}{RC-1}$$

$$\sigma_{GB} = \frac{\sum_{c=0}^{c-1}\sum_{r=0}^{r-1}(G_{cr} - \overline{G})(B_{cr} - \overline{B})}{RC-1}$$

Where

R is total rows

C is total columns

$R', G\ and\ B$ are total number of red, green and blue pixel in an image

$R'_{cr}, G_{cr}\ and\ B_{cr}$ are crth value of red, green and blue pixel

$\sigma^2_{R'}, \sigma^2_G\ and\ \sigma^2_B$ are variance of red, green and blue colour in an image

Variance and covariance can be represented together in a symmetrical matrix called variance-covariance matrix. Along diagonal elements, variance of each column appears. And at other places, there are covariances. Covariance matrix can be shown as $\sum$, a 3x3 matrix

$$\sum = \begin{matrix} \sigma^2_{R'} & \sigma_{R'G} & \sigma_{R'B} \\ \sigma_{GR'} & \sigma^2_G & \sigma_{GB} \\ \sigma_{BR'} & \sigma_{BG} & \sigma^2_B \end{matrix}$$

6.2 CORRELATION AND STANDARD DEVIATION

Correlation defines the degree in which two colours moves in according to each other. Mathematically, it can define as covariance of a colour divided by square root of product of both the colours. It is represented as

$$\rho_{R'G} = \frac{\sigma_{R'G}}{\sqrt{\sigma^2_{R'} * \sigma^2_G}}$$

$$\rho_{R'B} = \frac{\sigma_{R'B}}{\sqrt{\sigma^2_{R'} * \sigma^2_B}}$$

$$\rho_{GB} = \frac{\sigma_{GB}}{\sqrt{\sigma^2_G * \sigma^2_B}}$$

Where

$\rho_{R'G}$ is correlation between red and green

$\rho_{R'B}$ is correlation between red and blue

ρ_{GB} is correlation between blue and green

Similarly, Standard deviation represents degree of deviation in variables from the actual average value. Mathematically, it is square root of the variance. We can say that for a pollutant free image, this quantity should be less as it RGB colours are closer to the average colour value of that image. And standard deviation is higher in polluted environment images.

7. IMAGE HISTOGRAM

It is a graphical representation of distribution of pixels of particular tone in an image. Its x-axis represents different tones and y-axis represents number of pixels of each tone. In horizontal axis itself, darker tones are represented first and then tones gets lighter as we move towards the right of the horizontal axis. Histograms lie on [0,255] which are the intensity range of the image. Histograms can be modified to enhance features of image. Its threshold value most often used to detect edges, co-occurrence matrices and image segmentation in images.

Method and Implementation

Software Used-C#, Emgu and Octave(GUI)

1. **MainForm.cs**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using System.Collections;
using System.Drawing.Drawing2D;
using Emgu.CV;
namespace ntccgui
{
    public partial class MainForm : Form
    {
        public Image original;
        String text;
        public Bitmap bmp;
            public MainForm()
            {
            InitializeComponent();
            }


        private void button1_Click(object sender, EventArgs e)
        {
            OpenFileDialog openFileDialog = new OpenFileDialog();
            openFileDialog.Filter = "Image Files(*.jpg; *.jpeg; *.gif; *.bmp; *.png;)|*.jpg; *.jpeg;
*.gif; *.bmp; *.png";
            if (openFileDialog.ShowDialog() == DialogResult.OK)
            {
                textBox1.Text = Path.GetFileName(openFileDialog.FileName);
                textBox2.Text = openFileDialog.FileName;
                pictureBox1.Image = new Bitmap(openFileDialog.FileName);
                bmp = new Bitmap(openFileDialog.FileName);
                original = Image.FromFile(openFileDialog.FileName);
            }
        }

        private void button3_Click(object sender, EventArgs e)
```

```
        {
            CompAvg avg = new CompAvg(bmp);
            avg.ComputeAvg();
            text = "\nThe average of Red, Green and Blue Pixels in image is : " + avg.AvgRed + "
, " + avg.AvgGreen + " and " + avg.AvgBlue;
            richTextBox1.Text = text;
        }

        private void button4_Click(object sender, EventArgs e)
        {
            text = text + "\nThe variance-covariance matrix is : \n";
            VCVMatrix vcv = new VCVMatrix(original);
            double[][] PDMMatrix = vcv.GetMatrix();
            for (int i = 0; i < 3; i++)
            {
                for (int j = 0; j < 3; j++)
                {
                    text = text + PDMMatrix[i][j] + "        ";
                }
                text = text + "\n";
            }
            text = text + "\nCorelation between Red and Green = " + vcv.CorelationRG;
            text = text + "\nCorelation between Green and Blue = " + vcv.CorelationGB;
            text = text + "\nCorelation between Red and Blue = " + vcv.CorelationRB;
            text = text + "\n\nStandard deviation of Red =" + vcv.SDR;
            text = text + "\nStandard deviation of Green =" + vcv.SDG;
            text = text + "\nStandard deviation of Blue =" + vcv.SDB;
            richTextBox1.Text = text;
        }
    }
}
```

2. **CompAvg.Cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ntccgui
{
    internal class CompAvg
    {
        private int[][] red;
        private int[][] green;
        private int[][] blue;
        private Size size;
        public double AvgRed, AvgGreen, AvgBlue,totalPix;
        public CompAvg(Bitmap bmp)
        {
```

```
    size= bmp.Size;
    Color[][] clr = null;
    red = null;
    green = null;
    blue = null;
    int[][] CurrentRed = new int[size.Width][];
    int[][] CurrentGreen = new int[size.Width][];
    int[][] CurrentBlue=new int[size.Width][];
    clr=new Color[size.Width][];
    for(int i = 0; i < size.Width; i++)
    {
        CurrentRed[i]=new int[size.Height];
        CurrentGreen[i]=new int[size.Height];
        CurrentBlue[i]=new int[size.Height];
        clr[i] = new Color[size.Height];
    }
    if (bmp != null)
    {
        for(int i = 0; i < size.Width; i++)
        {
            for(int j=0;j<size.Height; j++)
            {
                clr[i][j] = bmp.GetPixel(i, j);
                CurrentRed[i][j] = clr[i][j].R;
                CurrentGreen[i][j] = clr[i][j].G;
                CurrentBlue[i][j] = clr[i][j].B;
            }
        }
    }
    red = CurrentRed;
    green = CurrentGreen;
    blue = CurrentBlue;
}
public int[][] GetRedPix() { return red; }
public int[][] GetGreenPix() { return green; }
public int[][] GetBluePix() { return blue; }
public void ComputeAvg()
{
    totalPix = size.Width * size.Height;
    AvgRed = 0;
    AvgGreen = 0;
    AvgBlue = 0;
    for (int i = 0; i < size.Width; i++)
    {
        for (int j = 0; j < size.Height; j++)
        {
            AvgRed+=red[i][j];
            AvgGreen+=green[i][j];
            AvgBlue+=blue[i][j];
        }
```

```
            }
            AvgRed /= totalPix;
            AvgGreen /= totalPix;
            AvgBlue /= totalPix;
        }
    }
}
```

3. **VCVMatrix.Cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ntccgui
{
    internal class VCVMatrix
    {
        public double[][] PDMMatrix;
        private Image image;
        public CompAvg RGB;
        public double CorelationRB, CorelationRG, CorelationGB, SDR, SDG, SDB;
        public VCVMatrix(Image current)
        {
            image = current;
            Bitmap OriginalImage = new Bitmap(current);
            RGB = new CompAvg(OriginalImage);
            RGB.ComputeAvg();
            PDMMatrix= new double[3][];
            for(int i = 0; i < 3; i++)
            {
                PDMMatrix[i]=new double[3];
            }
            GetMatrix();
        }
        public double[][] GetMatrix()
        {
            for (int i = 0; i < 3; i++)
            {
                for (int j = 0; j < 3; j++)
                {
                    double iAvg = 0;
                    double jAvg = 0;
                    int[][] iValue = GetColor(i, ref iAvg);
                    int[][] jValue = GetColor(j, ref jAvg);
                    PDMMatrix[i][j] = ComputeSigma(iValue, jValue, iAvg, jAvg);
                }
            }
            CorelationGB = ComputeCorelation(1, 2);
```

```
    CorelationRB = ComputeCorelation(0, 2);
    CorelationRG = ComputeCorelation(0, 1);
    SDR=GetStandardDeviation(0);
    SDG=GetStandardDeviation(1);
    SDB=GetStandardDeviation(2);
    return PDMMatrix;
}
private int[][] GetColor(int cno, ref double avg)
{
    switch (cno)
    {
        case 0:
            {
                avg = RGB.AvgRed;
                return RGB.GetRedPix();
            }
        case 1:
            {
                avg = RGB.AvgGreen;
                return RGB.GetGreenPix();
            }
        case 2:
            {
                avg = RGB.AvgBlue;
                return RGB.GetBluePix();
            }
        default:
            {
                avg = RGB.AvgRed;
                return RGB.GetRedPix();
            }
    }
}
private double ComputeSigma(int[][] iColor, int[][] jColor, double iAvg, double jAvg)
{
    double Sigma = 0;
    double check1 = 0, check2 = 0;
    double temp1=0, temp2=0;
    int count = 0;
    for(int i=0;i< image.Size.Width; i++)
    {
        for(int j=0;j< image.Size.Height; j++)
        {
            temp1 = iColor[i][j] - iAvg;
            temp2 = jColor[j][j] - jAvg;
            if (jColor[i][j] < jAvg)
            {
                count++;
                check1 = check1 + temp1;
                check2 = check2 + temp2;
```

```
                }
                Sigma = Sigma + (temp1 * temp2);
            }
        }
        int check3 = (int)check2;
        return (Sigma / RGB.totalPix - 1);
    }
    public double ComputeCorelation(int c1, int c2 )
    {
        double Corelation = 0;
        double covar1 = Math.Sqrt(PDMMatrix[c1][c1]);
        double covar2 = Math.Sqrt(PDMMatrix[c2][c2]);
        Corelation = PDMMatrix[c1][c2]/(covar1*covar2);
        return Corelation;
    }
    public double GetStandardDeviation(int c1)
    {
        return (Math.Sqrt(PDMMatrix[c1][c1]));
    }
  }
}
```

4. **histogram.m**

```
clc

clear all

clc



#Read an RGB image

image=imread('polluted sky.jpeg');



#Extract color channels

  #Red channel

  red=image(:,:,1);

  #Green channel

  green=image(:,:,2);

  #Blue channel
```

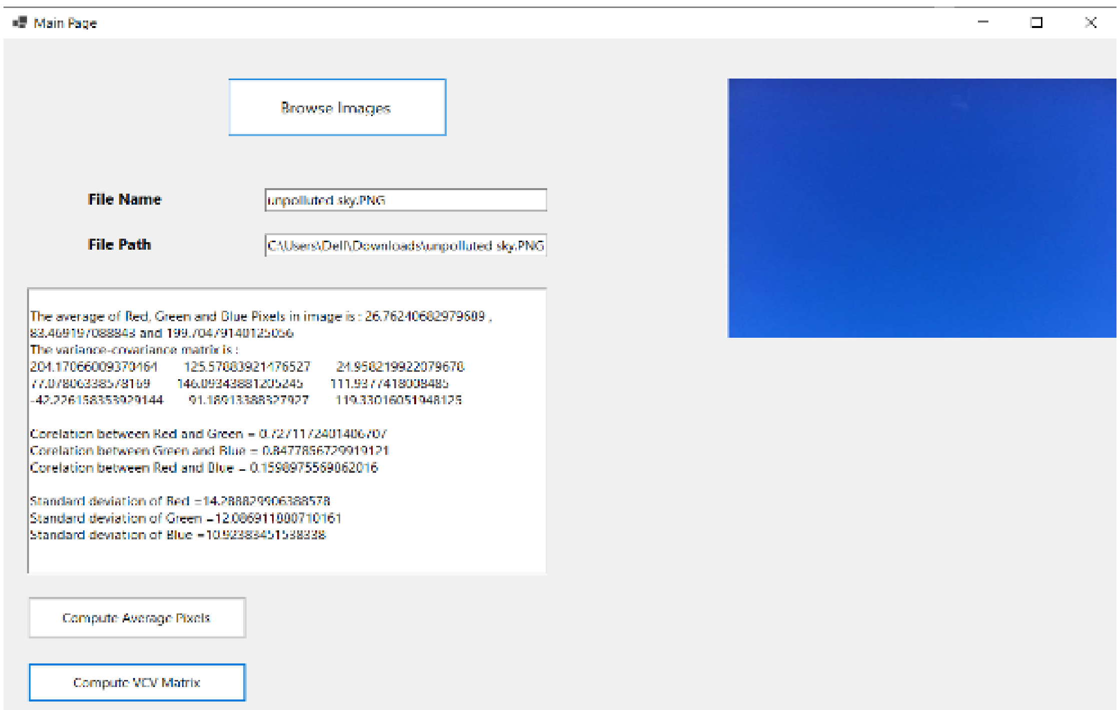
```
  blue=image(:,:,3);

#Just black channel
justBlack=zeros(size(image, 1), size(image, 2), 'uint8');

#Create RGB channels
AllRed=cat(3, red, justBlack, justBlack);
AllGreen = cat(3, justBlack, green, justBlack);
AllBlue = cat(3, justBlack, justBlack, blue);

%Displaying RGB images
subplot(3, 3, 2);
imshow(image);
title('Original RGB Image')
subplot(3, 3, 4);
imshow(AllRed);
title('Red Channel')
subplot(3, 3, 5);
imshow(AllGreen)
title('Green Channel')
subplot(3, 3, 6);
imshow(AllBlue);
title('Blue Channel')
```

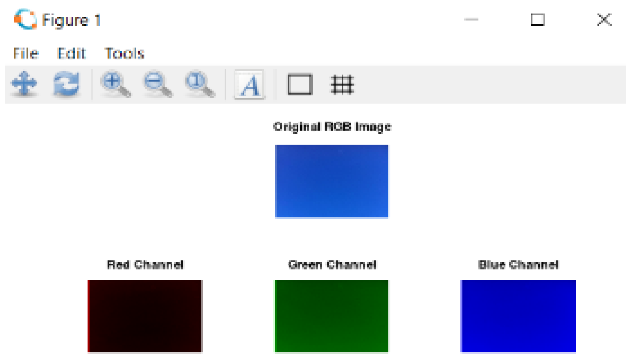
```
#Displaying RGB Histogram

red=imhist(image(:,:,1));

green=imhist(image(:,:,2));

blue=imhist(image(:,:,3));

figure

plot(red,'r')

hold on,

plot(green,'g')

hold on

plot(blue,'b')

legend(' Red channel','Green channel','Blue channel');

hold off
```

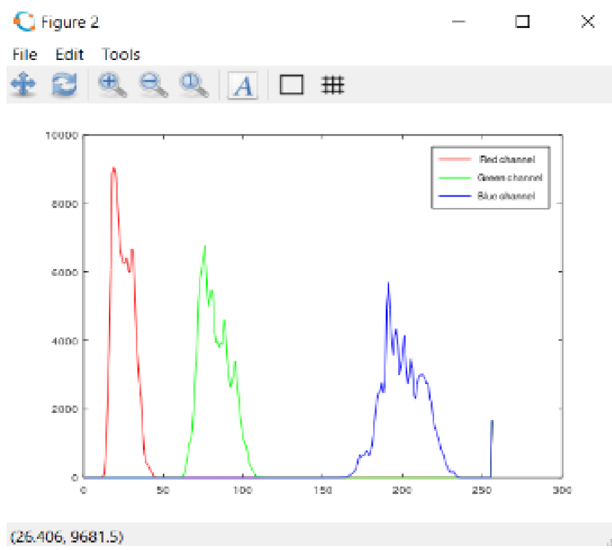
Results and Discussions

The above program gives a menu where we can choose any image from the desktop. After the user browse the file, information such as name, path and image will be displayed. Now we can choose to calculate average RGB, variance-covariance matrix, their corelation and Standard deviation. And in Octave program, we can see image, its red, green and blue channel images and histogram of the channels.

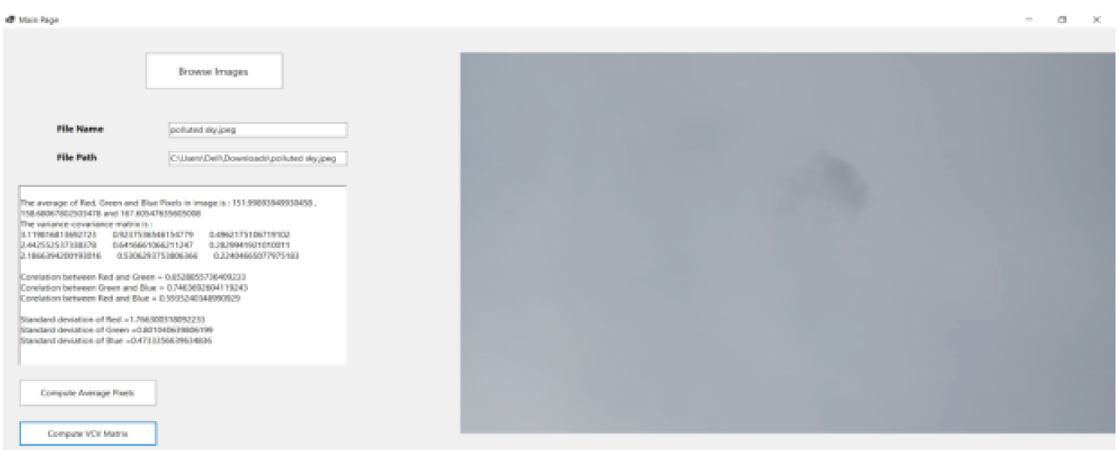
1. For unpolluted image

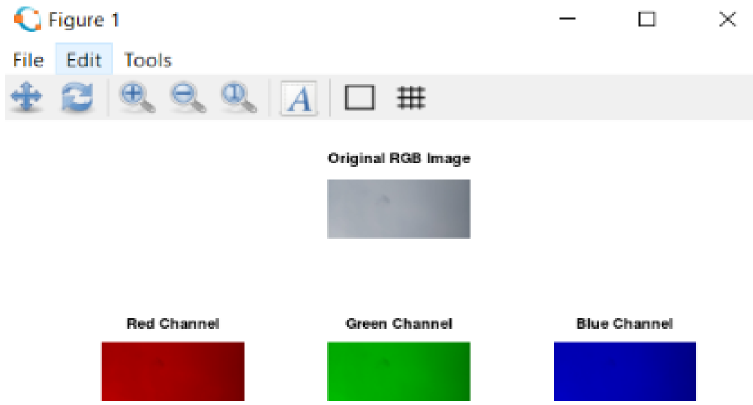


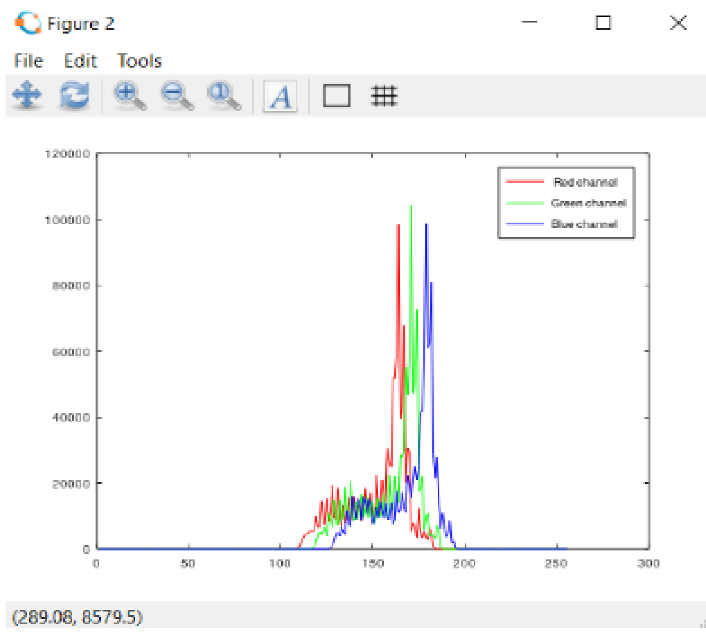




2. For polluted image







Purer an image is, more distinguishable red, green and blue channels it has and vice versa. Also, we can observe that polluted image has more shift towards right side. This shows it has more lighter tones than darker tone. With the help of histogram we can more easily distinguish polluted and non-polluted digital images.

Conclusion

Air pollution is a global issue that kills millions of people each year. We can monitor the level of air pollution using easily available digital images. A digital image contains a lot of information. We can extract some information from it and calculate the deviation that is occurring because of pollutants. Most effective representation to know characteristic of an image is histogram. We can tell a lot information at once from histograms. In simple way, we can monitor air pollution level of our environment with digital images.

References

1. Sultana, Samia. "A Comparison Study of Air Pollution Detection using Image Processing, Machine Learning and Deep Learning Approach." *Global Journal of Computer Science and Technology* (2019).
2. Amritphale, Amrita Nikhil. "A digital image processing method for detecting pollution in the atmosphere from camera video." (2013).