FOG REMOVAL AND OBJECT DETECTION USING IMAGE PROCESSING

Project Report

Submitted by

Pratik Dhebe 111507014

Yash Pawar 111507042

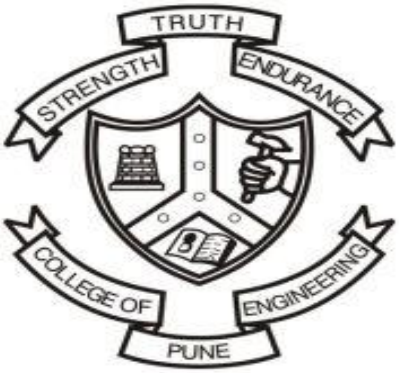
Rupal Lautre 111507048

Submitted in partial fulfilments of the requirements of the degree of

B.Tech. Electronics and Telecommunication

Under the guidance of

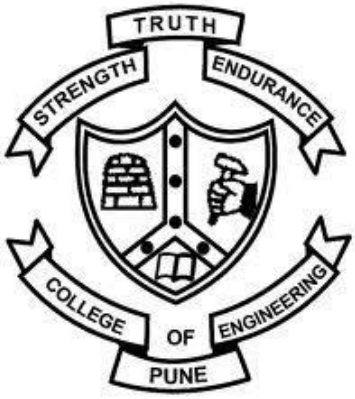
Mrs. V.N. More



DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

COLLEGE OF ENGINEERING, PUNE 2018-19

CERTIFICATE



This is to certify that the report entitled `Vision Enhancement using Image Processing ' submitted by Pratik Dhebe (MIS No. 111507014), Yash Pawar (MIS No. 111507042), Rupal Lautre (MIS No. 111507048) in the partial fulfilment of the requirement for the award of degree of Bachelor of Technology (Electronics and Telecommunication Engineering) of College of Engineering Pune, affiliated to the Savitribai Phule Pune University, is a record of their own work.

Mrs. V.N. More
Project Guide
Department of
Electronics and Telecom
College of Engineering
Pune

Dr. S.P. Mahajan
Head of the Department
Department of
Electronics and Telecom
College of Engineering
Pune

Date:

Place:

Acknowledgements

We are grateful to our College and Department of Electronics and Telecommunication for providing us this great opportunity of working on this project. It is their visionary objective to encourage student for the curriculum-oriented project that has created this extraordinary opportunity for us.

The authors sincerely thank our project guide Mrs.V.N. More for her guidance towards the design and development phases of the project. The key concepts and fundamentals could not have been understood to their fullest extent without her support. Our sincere thanks to all the faculties of department for their immense support. Special thanks to our faculty advisers Mrs. Vanita Agarwal and Mr. Swapnil Mali for their support.

The team has greatly benefited from the faculty cooperation. It is their guidance and excellence in respective fields that makes the project a sustainable venture. The various facilities and labs made available to the team by the department have been instrumental in our consistent progress. The provision of the required resources by the department has really helped us throughout the tenure of the project.

The team is immensely grateful to Head of the Department Dr. S.P. Mahajan. It is his support and encouragement to students to work on a project of their field of interest that has made this possible.

Abstract

Fog is one of the biggest problems in hilly and cold regions. Road fatality occurs every year in thousands because of weather, such as fog, and it is increasing each year because of pollution and smog. According to worldwide research, increased visibility between vehicles reduces the risks of crash by over 30 %. This report is a solution to this important problem. It deals with the processing of an image in order to produce a defogged image. We tried to implement this code on a raspberry pi and capture image with Pi camera starting from the Foggy picture using algorithms. This system can be extended further to higher levels, such as satellite images in the stormy or cyclone area. We can improve satellite imaging with high speed camera capture and high-speed processor.

Table of Contents

Introduction

The atmospheric haze causes numerous problems and restricts many applications. The visibility range may be greatly affected by the fog compared to other turbid conditions. The roads with low fogs are 300 m to 1000 m but under 100 m in dense fog. The problem is thus that the human performance is limited by low visibility. Depending on the angle of view, the human eye has a resolution of 324 to 576 megapixels, but the visibility is reduced in the event of a fog or other weather haze.

We are all aware of flight problems, train cancellation due to bad weather. Visibility is impaired due to fog and haze. Poor visibility causes delay in flight, diversion, cancellation and accidents in automobiles. Driving at night in case of fog is a challenging situation, which can lead to a fatal accident.

The atmospheric particles, mainly water droplets, cause absorption and dispersion, which causes attenuation and airlight problem to degrade the image quality. Reducing the contrast is called attenuation due to scattering. The effect of whiteness on the observer or camera is called airlight. Improving the image quality by removing fog is therefore an unavoidable technique.

Using image processing technique, we focus mainly on improving image quality in the foggy area. Thus, in the real-time application, e.g. in object detection, it can be used for better sight in the foggy atmosphere for driver.

Literature Survey

With the advancement in technology, many single image fog removal methods have been proposed.

Md. Imtiyaz Anwar and Arun Khosla (2016) proposed Vision enhancement through removal of single image fog. This technique does not require pre-processing of images. It uses High Dynamic Range (HDR) tone mapping by the Dark Channel Prior (DCP). This uses a Weighted Least Squares (WLS) method to adjust contrast and view details in the image. This technique can be used for grayscale and colour foggy images. This creates an enhanced edge that preserves the image. Different types of scenes are tested under a fog condition and gamma correction values are selected with other coefficients to make the proposed method more efficient. Subjective and objective assessments with fixed and variable x verify the efficiency of the proposed method. The algorithm used in this paper has an advantage that it can preserve the edges which can restore the sharp details at the same time it can retain the colour information as well.

Single Image Haze Removal Using Dark Channel Prior by Kaiming He, Jian Sun, and Xiaoou Tang (2011)

The method used in this paper emphasizes on the Dark Channel Prior (DCP), which is an efficient method for removing fog in a single image. This method first estimates the transmission followed by soft matting which is done using a Laplacian matrix which deals with spatially variant white balance problems. This is followed by recovering of scene radiance which is done using a transmission map.

Edge-Preserving Decompositions for Multi-Scale Tone and Detail Manipulation (2008) by Farbman, Fattal, Lischinski, Szeliski: In this paper, the edge preserving method is implemented by weighted least square technique. They showed after that how to use this to preserve edge multiscale decompositions. They have compared their techniques with different techniques, such as bilateral filters, which is better than other techniques. Also, this technique avoids halos in the image. Therefore, it can be used to map, manipulate and abstract images.

You Only Look Once: Unified, Real-Time Object Detection by Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi: In this paper a new method for object detection has been proposed. This method can perform real-time object detection at the rate of 45 frames per second. It uses unified detection method which combines all the components into the same neural network. The proposed network has 24 convolutional layers followed by 2 fully connected layers. As it is a unified network this network is extremely fast.

Objectives:

1. To develop an efficient method to remove fog content from an image and improve the visibility.
2. To detect the presence of different objects in the image.
3. To deploy the developed algorithm onto Raspberry pi.

Resources Required:

1 Software Resources:

1. MATLAB (optional)
2. OpenCV (with Python).
3. Raspbian Operating system (For Raspberry Pi).

2 Hardware Resources:

1. Raspberry Pi Model 3B.
2. Raspberry Pi Camera

Methodology:

Defog Algorithm:

The method employed for fog removal uses Initial Estimation of Air-light followed by transmission estimation through refined map and then obtaining a restoration using an accurate transmission map. A fog model can be given as:

$$I_0(z) = \frac{I(z)-A}{t(z)} + A$$

Where I(z) is an input foggy Image, $I_0(z)$ is fog-free image, A is atmospheric light, t(z) is the transmission parameter. The steps included in the Defog algorithm are:

1. Initial estimation of atmospheric and Air-Light.

Initially the image is normalized. After that the pixels with highest intensity are computed. Out of those pixels 0.1% of the brightest pixels whose value is close to 1 are treated as skylight. Dark Channel Prior (DCP) principle is used to determine the darkest of pixels. These are determined by calculating the pixels with minimum intensity across RGB channel. Dark Channel is calculated using the following equation:

$$Dark\ channel = \min_{c\ \in\ (RGB)} \left(\min_{(i,j)\in\Omega(z)} \mathrm{I}_c(i,j) \right) > 0$$

Where I_c is the colour channel of the RGB image. Cϵ {r, g, b} is used to find the minimum value in colour channels, the second 'min' operator behaves as a minimum filter across a local patch Ω(z), centred at z. Dark channel estimates a more reliable value of Air- light and can find the minimum of brightest pixels of the input foggy image.

2. Transmission Estimation.

It is important to measure the fog thickness in an image, t(z) is the fraction of light that reaches observer without scattering. For the estimation of t(z), the value of atmospheric light in Dark Channel Prior is used with a constant local patch but it is not necessarily always constant. The size of the local patch depends upon the specific application. Therefore, min operator is used in the local window on the input degraded image by dividing the previously obtained atmospheric light in all the three colour channels respectively. So, the fog model becomes:

$$min_{(c\epsilon RGB)} \left(\frac{I(z)}{A} \right) = t(z)\ min_{(c\epsilon RGB)} \left(\frac{I_0(z)}{A} \right) + (1 - t(z))$$

To get the transmission map for images without sky region is possible but it is not always suitable for images with sky region like road images as a front view from a vehicle. It Depends on the portion of sky region included. To perceive the scene depth, it is mandatory to preserve

a little quantity of fog that can vary according to the entropy of the foggy image. If the entropy is smaller, that quantity is closer to one and for higher entropy foggy images it is set nearer to zero. Equation will come out to be:

$$\hat{t}(z) = 1 - \omega \min\left(\frac{\mathrm{I(z)}}{\mathrm{A}}\right), 0 < \omega < 1$$

3. Getting Accurate Transmission map.

In order to obtain accurate Transmission map, we use guided filter. The filtering output is locally a linear transform of the guidance image. This filter has the edge-preserving smoothing property like the bilateral filter but does not suffer from the gradient reversal artifacts. It is also related to the matting Laplacian matrix, so is a more generic concept and is applicable in other applications beyond the scope of smoothing.

4. Restoration map

The final scene is restored with a little modification. This modification is required due to the very small value of t(z) (close to zero) which makes the restoration map very prone to noise. Therefore, a lower bound value of $t_0 = 0.1$ is set which allows better contrast gain and avoids very small value of t(z). The refined map can be obtained using:

$$I_o(z) = \frac{I(z) - A}{\max(\hat{t}, t_0)} + A$$

5. Object Detection:

For Object Detection from the defogged image we have used an Algorithm YOLO (You Only Look Once) version 3. This algorithm comes out to be the latest of the lot. It is faster than its previous versions although there is a compromise on the accuracy. The future advancements in our project will require the algorithm to perform real time object detection which makes this algorithm a very good choice.

Architecture network for YOLO v3:

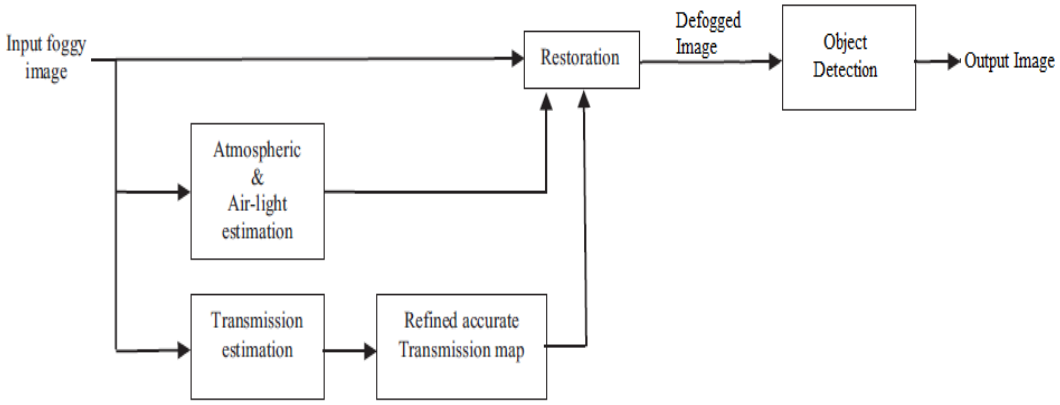
The new architecture features residual skip and upsampling connections. The most outstanding characteristic of v3 is the three-scaled detections. YOLO is a fully convolutional network and is generated through the application of a 1x 1 kernel on a feature map. In YOLO v3, detection is performed by applying 1x 1 detection kernels on three different size feature maps at three different network locations.

The kernel shape is 1× 1× (B× (5+ C)). Here B is the number of bounding boxes that can be predicted by a cell on the feature map, "5" is the number of classes for the 4 bounding box attributes and one object confidence. The size of the kernel is 1x 1×255 in YOLO v3 trained on COCO (common objects in context), B= 3 and C= 80. The feature map produced by this kernel has the same height and width of the previous feature map as described above and has detection attributes along the depth.

YOLO v3 makes prediction at three different scales, which are accurately given by downsampling the dimensions of the input image by 32, 16 and 8 respectively. The first detection is made by the 82nd layer. For the first 81 layers, the image is down sampled by the network, such that the 81st layer has a stride of 32.

Then, before being sampled by 2x, the feature map from layer 79 is subjected to a few convolutional layers. This feature map is then concatenated with the layer 61 feature map. Then again, some 1x 1 convolutional layers are subjected to the combined feature maps to fuse the features from the earlier layer (61). Then the 94th layer makes the second detection. A similar procedure is followed again, where the layer 91 feature map is subjected to a few convolutional layers before being concatenated with a layer 36 feature map. As before, the information from the previous layer is fused by a few 1x 1 convolutional layers (36).

Block Diagram:



Software Implementation:

Matlab:

For simulation and algorithm testing purposes, we have used Matlab as our software. Matlab facilitates a wide range of functionality and allows modification. The basic data element of Matlab is a matrix. Several mathematical operations that work on arrays matrices are built in Matlab environment. This is suitable for Image processing Applications. Matlab's functionality can be greatly expanded by the addition of toolboxes. There are specific functions that provide more functionality that is specialized.

OpenCV:

Matlab cannot be used for portability and application building purposes. Hence, for Application building Purpose and deploying the algorithm onto Raspberry pi, OpenCV is used. OpenCV is used for Image processing and computer Vision Applications.

Raspberry Pi operating system

Operating System: - Raspbian (Jessie), Mobaxterm (GUI)

Language: - Python 3

Package: - OpenCV, pillow

Hardware Implementation:

Raspberry Pie 3 B+

The Raspberry Pi 3 Model B+ is the latest product in the Raspberry Pi 3 range, boasting a 64-bit quad core processor running at 1.4GHz, dual-band 2.4GHz and 5GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and PoE capability via a separate PoE HAT The dual-band wireless LAN comes with modular compliance certification, allowing the board to be designed into end products with significantly reduced wireless LAN compliance testing, improving both cost and time to market. The Raspberry Pi 3 Model B+ maintains the same mechanical footprint as both the Raspberry Pi 2 Model B and the Raspberry Pi 3 Model B.

Specifications

Processor: - Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz

RAM: -1GB LPDDR2 SDRAM

Connectivity: -

- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN,
- Bluetooth 4.2,
- BLE Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps) 4 × USB 2.0 ports
- Access Extended 40-pin GPIO header

Video & Sound:

- 1 × full size HDMI
- MIPI DSI display port
- MIPI CSI camera port
- 4 pole stereo output and composite video port

Multimedia:

H.264, MPEG-4 decodes (1080p30); H.264 encode (1080p30) OpenGL ES 1.1, 2.0 graphics

Storage:

Micro SD format for loading operating system and data storage

Input Power: -

- 5V/2.5A DC via micro USB connector
- 5V DC via GPIO header
- Power over Ethernet (PoE)–enabled (requires separate PoE HAT)

Environment: -

Operating temperature, 0–50°C

Camera

Raspberry Pi Camera Board v1.3:

Fully Compatible with Both the Model A and Model B Raspberry Pi

- 5MP Omnivision 5647 Camera Module
- Still Picture Resolution: 2592 x 1944
- Video: Supports 1080p @ 30fps, 720p @ 60fps and 640x480p 60/90 Recording
- 15-pin MIPI Camera Serial Interface - Plugs Directly into the Raspberry Pi Board
- Size: 20 x 25 x 9mm
- Weight 3g
- Fully Compatible with many Raspberry Pi cases

Setup Raspberry:

Download Raspbian Jessie from https://www.raspberrypi.org/downloads/raspbian/

- Unzip the file and make a SSH file that have no extension in it and store it inside a memory card of size more than 4GB.
- Insert MicroSD in r-pi and download Mobaxterm on your PC and install it
- Power up r-pi using a 5V power source using U.S.B 2.0 and using Rj-45 cable insert both end in your PC and r-pi.
- In network and Sharing Centre use Wi-Fi option sharing select share internet over Ethernet.
- Run Mobaxterm SSH session there you get login id and password use pi as login name and raspberry as password and hit enter you will enter into Raspbian Jessie Virtual environment

Installing Python AND openCV packages

Open Terminal and start entering the following commands to download build and run the python openCV package

$ sudo apt-get update

$ sudo apt-get upgrade

$ sudo apt-get install build-essential cmake pkg-config

$ sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev

$ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev

$ sudo apt-get install libxvidcore-dev libx264-dev

$ sudo apt-get install libgtk2.0-dev

$ sudo apt-get install libatlas-base-dev gfortran

$ sudo apt-get install python2.7-dev python3-dev

$ cd ~

$ wget -O opencv.zip https://github.com/Itseez/opencv/archive/3.1.0.zip

$ unzip opencv.zip

$ wget -O opencv_contrib.zip https://github.com/Itseez/opencv_contrib/archive/3.1.0.zip

$ unzip opencv_contrib.zip

$ wget https://bootstrap.pypa.io/get-pip.py

$ sudo python get-pip.py

$ sudo pip install virtualenv virtualenvwrapper

```
$ sudo rm -rf ~/cache/pip
# virtualenv and virtualenvwrapper
export WORKON_HOME=$HOME/virtualenvs
source /usr/local/bin/virtualenvwrapper.sh
$ echo -e "\n# virtualenv and virtualenvwrapper" >> ~/.profile
$ echo "export WORKON_HOME=$HOME/virtualenvs" >> ~/profile
$ echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~/.profile

$ source ~/.profile

$ mkvirtualenv cv -p python3
$ source ~/.profile
$ workon cv
$ pip install numpy
$ cd ~/opencv-3.1.0/
$ mkdir build
$ cd build
$ cmake -D CMAKE_BUILD_TYPE=RELEASE \
    -D CMAKE_INSTALL_PREFIX=/usr/local \
    -D INSTALL_PYTHON_EXAMPLES=ON \
    -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-3.1.0/modules \
    -D BUILD_EXAMPLES=ON ..
$ make -j4

  $ sudo make install
  $ sudo ldconfig

$ ls -l /usr/local/lib/python3.4/site-packages/
total 1852
-rw-r--r-- 1 root staff 1895932 Mar 20 21:51 cv2.cpython-34m.so

  $ cd /usr/local/lib/python3.4/site-packages/
  $ sudo mv cv2.cpython-34m.so cv2.so
```

$ cd ~/.virtualenvs/cv/lib/python3.4/site-packages/

$ ln -s /usr/local/lib/python3.4/site-packages/cv2.so cv2.so

Interface with Camera

The Raspberry Pi Camera Board plugs directly into the CSI connector on the Raspberry Pi. It's able to deliver a crystal clear 5MP resolution image, or 1080p HD video recording at 30fps! Latest Version 1.3! Custom designed and manufactured by the Raspberry Pi Foundation in the UK, the Raspberry Pi Camera Board features a 5MP (2592?1944 pixels) Omnivision 5647 sensor in a fixed focus module. The module attaches to Raspberry Pi, by way of a 15 Pin Ribbon Cable, to the dedicated 15-pin MIPI Camera Serial Interface (CSI), which was designed especially for interfacing to cameras. The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data to the BCM2835 processor. While interfacing following steps were taken.

- Open up your Raspberry Pi Camera module. Be aware that the camera can be damaged by static electricity. Before removing the camera from its grey anti-static bag, make sure you have discharged yourself by touching an earthed object (e.g. a radiator or PC Chassis).
- Install the Raspberry Pi Camera module by inserting the cable into the Raspberry Pi. The cable slots into the connector situated between the Ethernet and HDMI ports, with the silver connectors facing the HDMI port.
- Boot up your Raspberry Pi.
- From the prompt, run "*sudo raspi-config*". If the "camera" option is not listed, you will need to run a few commands to update your Raspberry Pi. Run "*sudo apt-get update*" and "*sudo apt-get upgrade*"
- Navigate to the "camera" option and enable it. Select “Finish” and reboot your Raspberry Pi.

Capturing Images Though Pi-Camera Using Python

Python Program for Image Capture

```
import picamera
from time import sleep

#create object for PiCamera class
camera = picamera.PiCamera()
#set resolution
camera.resolution = (1024, 768)
camera.brightness = 60
camera.start_preview()
#add text on image
camera.annotate_text = 'Hi Pi User'
sleep(5)
#store image
camera.capture('image1.jpeg')
camera.stop_preview()
```

RASPBERRY PI'S PERFORMANCES AND CONSTRAINTS

Raspberry Pi advantages can be summarized as follows:

The Raspberry Pi is a small independent computer that runs on the various distribution of Linux operating system and can be programmed as needed.

- It has a very large working memory (RAM memory).
- It has expandable memory to store the data (up to 64GB).
- It works on multi operating processor (supports a set of instructions).
- It operates at speeds from 700 MHz to 1000 MHz
- It has support for USB 2.0 which allows its expansion with a large number of peripherals.
- Depending of the needs it is possible to expand the
- Raspberry Pi with WIFI and Bluetooth adapters (power and range can be changed by changing the adapter).
- Expansion and communication with network devices over a LAN adapter are possible.
- It can be expanded with various prototype shields (Pi-Face, GSM/GPRS & GPS, GPIO expansion board, GertBoard)
- It is possible to form an expandable system with various electronic components (sensors and electronic circuits) using digital inputs and outputs,
- I^2c or SPI protocols. C, Python or object-oriented languages such as C++ and Java can be used for programming of Raspberry Pi.
- It can be powered form battery or solar cell.
- It can be run in server mode.
- A various web server can be installed and running on Raspberry Pi

Disadvantage

The main disadvantages of Raspberry Pi are [4]:

It does not have a real-time clock (RTC) with a backup battery but it can easily work around the missing clock using a network time server, and most operating systems do this automatically.

- It does not have a Basic Input Output System (BIOS) so it always boots from an SD card.
- It does not support Bluetooth or WIFI out of the box, but these supports can be added by USB dongles.

- Unfortunately, most Linux distributions are still a bit picky about their hardware, so it should be first checked whether flavour of Linux supports particular device.
- It doesn’t have built in an Analog to Digital converter. External component must be used for AD conversion.

Performance Analysis:

Performance analysis is done to obtain the results of the proposed algorithm. Different parameters are used to perform the analysis.

Parameters used are:

Contrast Gain:

It is the mean contrast difference between the input image and the defogged image. Higher value of contrast Gain indicates better performance as fog free images have higher contrast as compared to foggy images.

The contrast Gain can be defined as:

$$CG = C_{I,(defogged)} - C_{I,(fog)}$$

The mean contrast of an image is mathematically expressed as:

$$C_I = \frac{1}{MN} \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} C(i,j)$$

C (i, j) is the contrast of the pixel (i, j). It is defined as:

$$C(i,j) = \frac{S(i,j)}{M(i,j)}$$

Where, S (i, j) and M(i, j) are the variance and mean of the image I(i, j).

Colour Information Entropy:

Entropy is a randomness statistical measurement used to characterize the input image texture. It can be determined from a gray-level image histogram; however, the image can be two-dimensional or multi-dimensional. In a coloured picture the CIE depicts the number of information. The maximum

CIE values are reached when a picture is not uniform, but CIE is minimum for a foggy or haze area. CIE is defined as mathematically:

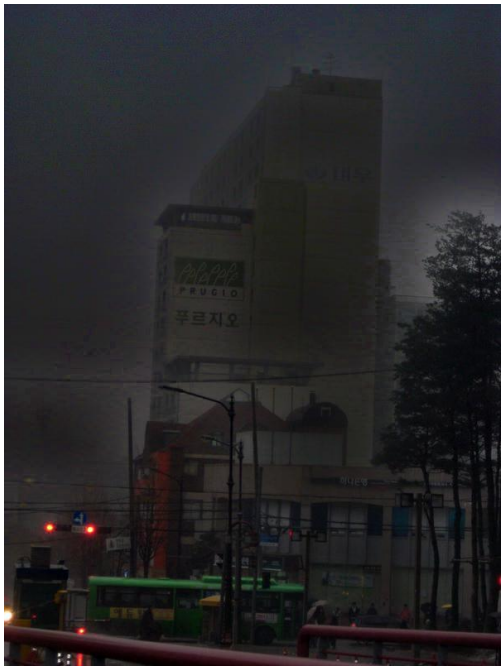
$$CIE = \sum_{k=0}^{L-1} P_k \log_2(P_k)$$

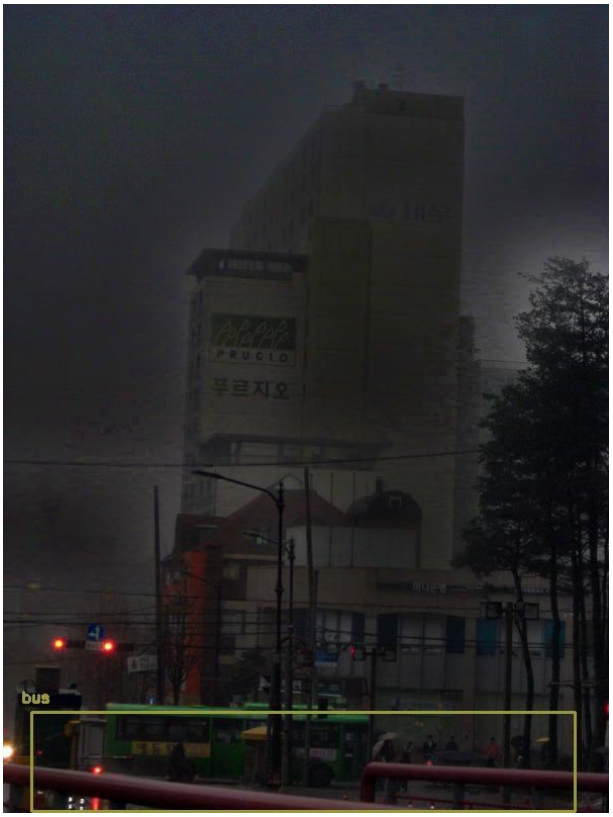
Where L is the gray level number and P_k is the gray level k or histogram count probability. A foggy or hazy image contains little texture information compared to fog-free images and therefore the CIE value is comparatively higher for the restored image. CIE therefore demonstrates the efficacy of any fog removal algorithm.

Results:

Test Image 1:

Results:
Size of Image: 800 x 600
Raw transmission rate between:
[0.2698, 0.8957]
Refined transmission rate between:
[0.2727, 0.9721]
Atmosphere :[255 255 255]
Contrast Gain :
[0.6485 0.6473 0.6443]
Color Information Index for:
Foggy image: 7.0658
Defogged Image: 5.7530

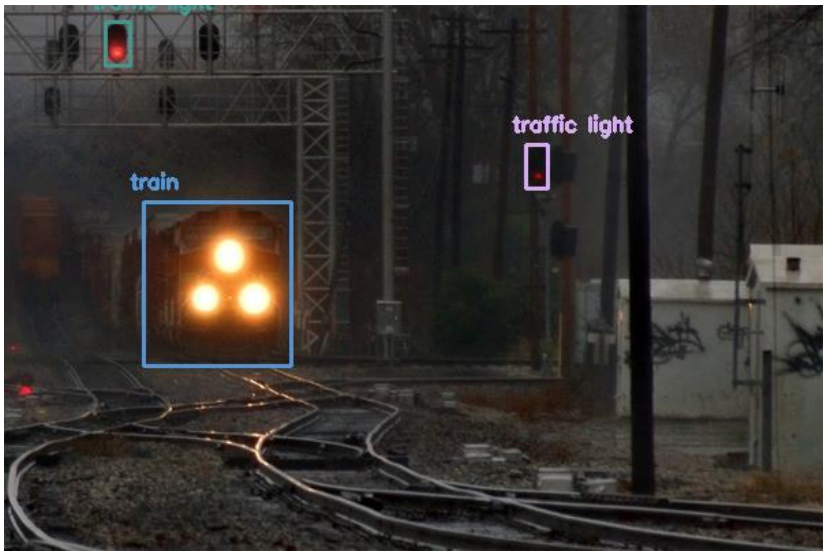




Test Image 2:

Results:
Size of Image: 531 x 800
Raw transmission rate between:
[0.0712, 0.6721]
Refined transmission rate between:
[0.0897, 0.8001]
Atmosphere :[224 222 226]
Contrast Gain :
[0.5637 0.5682 0.5758]
Color Information Index for:
Foggy image: 5.4863
Defogged Image: 6.7803

Test Image 3:



Results:
Size of Image: 400 x 600
Raw transmission rate between:
[0.1655, 0.8659]
Refined transmission rate between:
[0.3440, 0.8869]
Atmosphere: [255 255 255]
Contrast Gain:
[0.6427 0.6501 0.6590]
Color Information Index for:
Foggy image: 6.8984
Defogged Image: 6.4569

Applications

Wide variety of applications are possible through this project. Defogging the Image can be helpful in improving the visibility of the observer. It can improve the quality of vision in areas where the presence of Dust, Smog, Smoke, Fog is predominant.

Vehicular Applications: In pollution concentrated regions there is a smog formation which leads to poor visibility on roads which can cause accidents leading to casualties. This setup can be used in vehicles to reduce road accidents.

Military Applications: This setup can also be used in surveillance at borders in poor visibility conditions. To observe and detect unusual activities across the borders this setup can be instrumental. Visibility can be major issue in Military Areas like Snowy regions and dessert areas. This method can be very effective for object detection purposes in such areas.

IOT applications: With the advancements in the field of internet of things, this technique can become an integral part of this ecosystem. This technique can be combined with internet which can provide solutions at a remote location which can be useful in number of other applications.

Autonomous systems: Autonomous systems such as self-driven cars can very well use this method in situations where visibility is a concern. With further advancements in this method, real-time object detection as well as path detection can be performed in poor visibility conditions.

Future Work

This project can be modified and improved for other applications. The hardware implementation can be changed for fast processing. This sets the basics for real time video processing, which can be wisely used in vehicles, trains and many other real time applications.

For processing of big images and large number of images, we have to change the hardware. We can use the fast processors. The FPGA implementation can also be build. The image taken can be sent using Wi-Fi module or by using internet services to the data centre. In data centre this high-speed machines can process these images and then it can transmit the parameters and the detected objects to the user. This information can be used according to applications.

This project also set base to clear the foggy images. There are lot of areas of poor visibility where the vision is impaired due to the fog, smog and various other factors. This can be improve by using high quality camera.

The algorithm we are using can be improved by fusing with others to detect the various factors and improvement in results. The projects give low quality image after defogging if the atmospheric content in image is much larger. Therefore, by combining and finding the other methods to avoid a problem.

Conclusion:

Different images were used for testing the algorithm. These images depicted different atmospheric conditions. When the algorithm was tested under these conditions the fog from the image was removed. However, for images with very high atmospheric content the output obtained was predominantly darker and thus sharp edges were not retained. Algorithm performs well for the value of $\Omega = 0.95$.

Object detection when performed on the images could mostly classify different objects in the image like Bus, car, train, traffic signal etc. The method used for fog removal has been successfully implemented and tested for different types of situations. The object detection works faster and can be employed for real-time object detection as well.

References

1. Md. Imtiyaz Anwar, Arun Khosla, Vision enhancement through single image fog removal, Eng. Sci. Tech., Int. J. (2017)
2. Kaiming He, Jian Sun, and Xiaoou Tang, Guided Image Filtering, **ECCV** 2010.
3. K. He et al., Single image haze removal using dark channel prior, IEEE Trans. Pattern Anal. Mach. Intell. 33 (12) (2011) 2341
4. A. Levin et al., A closed-form solution to natural image matting, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2) (2008) 228–242