FLOOD MONITORING SYSTEM

MAJOR PROJECT REPORT

Submitted By

STEBIN JOSEPH *(CEC15CS058)*
SHELNA SEBASTIAN *(CEC15CS049)*
AMETA JOBY *(CEC15CS009)*
SREERAM P *(CEC15CS055)*

Under the esteemed guidance of

Ms. GREESHMA N GOPAL*(Assistant Professor)*

(Computer Science & Engineering)

May 2019

Department of Computer Science and Engineering
College of Engineering Cherthala
Pallippuram P O,Alappuzha-688541
Phone: 0478 2553416,Fax: 0478 2552714
http://www.cectl.ac.in

FLOOD MONITORING SYSTEM

A PROJECT REPORT

Submitted By

STEBIN JOSEPH *(CEC15CS058)*
SHELNA SEBASTIAN *(CEC15CS049)*
AMETA JOBY *(CEC15CS009)*
SREERAM P *(CEC15CS055)*

Under the esteemed guidance of

Ms. GREESHMA N GOPAL

In partial fulfilment of the requirements for the award of the degree
of
Bachelor of Technology
in
Computer Science and Engineering
of
A.P.J ABDUL KALAM Technological University

May 2019

Department of Computer Science and Engineering
College of Engineering Cherthala
Pallippuram P O,Alappuzha-688541
Phone: 0478 2553416,Fax: 0478 2552714
http://www.cectl.ac.in

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

COLLEGE OF ENGINEERING CHERTHALA

ALAPPUZHA-688541

C E R T I F I C A T E

This is to certify that, the project report titled ***FLOOD MONITORING SYSTEM*** is a bonafide record of the **CS492 Project** presented by **STEBIN JOSEPH** (REG NO: CEC15CS058), **SREERAM P** (REG NO: CEC15CS055), **SHELNA SEBASTIAN** (REG NO: CEC15CS049) and **AMETA JOBY** (REG NO: CEC15CS009), Eigth Semester B. Tech. Computer Science & Engineering student, under our guidance and supervision, in partial fulfillment of the requirements for the award of the degree, **B. Tech. Computer Science & Engineering** of **APJ Abdul Kalam Technological University**.

Internal Supervisor	**UG Coordinator & Head Of Dept.**	**External Supervisor**
Ms. Greeshma N G	**Dr. Preetha Theresa Joy**	
Assistant Professor	Professor	
Computer Science & Engg.	Computer Science & Engg.	

ACKNOWLEDGEMENT

We take this opportunity to express our sincere gratitude to the people who have been instrumental in the successful completion of our main project. For every endeavour in our life, the help of the Lord has always followed. We have yet again experienced his loving kindness while preparing for this project. We would like to express my sincere thanks to our Principal, **Dr. Mini M. G**, for the valuable support. We would also like to thank our Head Of Computer Science Department and project Co-ordinator, **Dr. Preetha Theresa Joy** (Professor, Department of Computer Science) for her support. We would also like to thank our project Guide, **Ms. Greeshma N Gopal** (Assistant Professor, Department of Computer Science) for her full hearted support and guidance.

We are indebted to all teaching and non-teaching staff of the Department of Computer Science & Engineering, friends and family for their co-operation and support, with out which we could never have completed our project this well. Thank you all for your whole hearted support and encouragement.

ABSTRACT

Detecting floods in real-time and taking rapid actions are of utmost importance to save human lives, loss of infrastructures, and personal properties. The main objective of our project is to develop a flash flood warning system to monitor the water level rise in rivers which are prone to severe flood. Here we develop a low-cost, low-power system using a Raspberry Pi camera to detect the rising water level. We employ image processing, edge detection and prediction method to detect the rising water level and predict the time of impact.

In the hardware section we use Raspberry-Pi, which has main tasks as an image processor and do an update to the system. OpenCV library is used as Image Processing Software. Some method which we use in the project are Region of Interest, Edge detection, Grayscale and Threshold etc. By using these methods, the system can read and monitor the water level. If the water level exceeds the specified level, this system will generate an early warning of impending floods by doing update timeline of water level conditions.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Flood becomes one of the major problems in most of the countries around the world. Although, we are able to forecast rainfall or to track storm path very precisely from the satellite images, the need to have real-time monitored data of water level is essential in order to make a reasonable decision on the actions necessary to be performed to prevent flooding. This approach will help in the implementation of effective and timely urban flood relief and management. This data can also be used to assess floods taken in the past and to plan remedial measures in the future.

1.1 GENERAL BACKGROUND

The main objectives of our project is to develop a flash flood warning system to monitor the water level rise in rivers which are prone to severe flood, to predict the level of water in an area during flood at a particular time. Beginning in July 2018, severe floods affected our state, due to unusually high rainfall during the monsoon season, which is the main reason behind the implementation of this project. In such scenarios, a record of the flood levels at different locations and at different times will be useful in order to take appropriate corrective action.

1.2 OBJECTIVE OF THE PROJECT

The main objectives of this project are:

1. To accurately monitor the increase in water level during the flood.

2. To predict the level of water in an area during flood at a particular time.

1.3 SCOPE OF THE PROJECT

This project can be used for flash flood warning. Flood monitoring system aims to monitor the water level in rivers and keeps updating the height of the water at regular intervals.Thus it will predict the level of water rise in an area during flood at a particular time. If there's a chance of flood, the authorities will be informed by our-self and easier evacuation plans and rescue of the people can be made possible.

1.4 SCHEME OF THE PROJECT

The scheme of the project can be summarized as:

- Monitoring and analyzing the water rise level in rivers.
- Implementing an efficient flash flood warning system to predict the impact of flood.

1.5 ORGANIZATION OF THE REPORT

- **Chapter 1** contains brief introduction to background of the project, objective of the project and its scope.
- **Chapter 2** includes the detailed literature survey for various techniques used for implementation.
- **Chapter 3** summarizes the implementation details of the project which includes different modules, block diagrams and algorithms used for edge detection, region of interest, height estimation.
- **Chapter 4** summarizes the implementation details which includes different python modules and libraries, methods used for this work.
- **Chapter 5** contains results of the implementation and the performance evaluation.
- **Chapter 6** Conclusion

Chapter 2

LITERATURE SURVEY

Recently, many innovations are under development in the area of disaster management. There are many systems proposed to monitor the flash floods in the flood prone area. Most of the systems use participatory sensing techniques and satellite imagery methods which does not guarantee accurate results.

- Jirapon Sunkpho and Chaiwat Ootamakorn, **Real-time flood monitoring and warning system**[1]

The developed system is composed of three major components: sensor network, processing unit, and application server. The real-time data of water condition is monitored remotely by utilizing wireless sensors network that utilizes the GPRS communication in order to transmit measured data to the application server, called VirtualCOM. The application server is a web-based system implemented using PHP and JAVA as the web application and MySQL as its relational database. Real-time data of water condition can be monitored remotely by utilizing wireless sensors network using GPRS. Users can view real-time water condition as well as the forecasting of the water condition directly from the web via web browser or via WAP. On-site data processing for flood prediction and on networking coverage area cannot be implemented. GPRS may be unstable, can cause difficulty in connecting the sensor at the remote site.

- Bhavana B Nair and Sethuraman N Rao,**Flood Monitoring using Computer Vision"**[2]

Estimate the water level in a flooded region using the human height as reference. Detection of human face and segmenting the human part in the image using deep learning algorithm. Human segmentation helps to find the pixel position of the water line. Classify each detected human face as male or female and use average male/female height as the reference. Once the pixel positions of the human feet and the water line are known, they can be used to estimate the water depth. Use Computer Vision algorithms to estimate the depth of flooding based on images. Images are geotagged and time-stamped which will be easy to locate. This system cannot prevent the impact nor it doesn't send warning messages regarding the impact. This system is used after the impact of flood. Here only the depth of the water is estimated. Human face detection algorithm may fail with low resolution images.

- RamKumar Narayanan, VM Lekshmy, Sethuraman Rao, Kalyan Sasidhar,**Urban Flood Monitoring Using Computer Vision"**[3]

This paper presents a novel method employing participatory sensing and computer vision to estimate the flood level. GPRS and SMS were used to communicate data from the remote stations to a server located at a data center. The method involves the participants capturing and uploading images of the partially submerged static structures such as buildings, lampposts etc., using their smart phones or other intelligent devices. The captured images are geo-tagged and uploaded to a server. The feature matching algorithm, SIFT finds the corresponding matching feature points between the captured and a reference image at the server. The flood line is then estimated and drawn against the reference image. Participatory sensing and computer vision to estimate the flood level. Yields immediate and considerably accurate results. Each and every observations of the flood levels at different locations and at different times is kept as a record. Here only the flood level is estimated and recorded for the future reference. It doesn't provide any warning system. Accuracy changes according to the resolution of images, as the images are taken in smart phone.

- Alvin E. Retamar, Felan Carlo C. Garcia , Jasmin Jane M. Yabut, Joven C. Javier, **Development of a Remote Station for Real-time Monitoring ofUrban Flooding"**[4]

This paper presents the design and development of urban flood monitoring stations that use pressure sensor to determine flooding levels. GPRS and SMS were used to communicate

LITERATURE SURVEY				
Ref	Technology	Advantages	Disadvantages	Keywords
[1]	GPRS, VirtualCOM, WAP, Wireless Sensor	Real-time data, Wireless sensors network which enables GPRS comminication	GPRS may be unstable, Networking coverage area cannot be implemented	Sensors, Real-time monitoring, Flood control.
[2]	Sensors, GPRS	Water depth estimation, Geo-tagged images	Detection algorithm may fail, Doesn't send warning messages	Computer Vision, Geotag.
[3]	Participatory sensing techniques	Estimate flood level, Immediate result, Observations are recorded	No warning system, No Accuracy	Computer vision, Feature matching, Flood monitoring, Participatory sensing.
[4]	GPRS, Web-based tool, Pressure Sensor	Sensing differences are small enough to issue warning	No accuracy, No reliablity	Urban flooding, Flood monitoring, Image processing.

Table 2.1: Literature Survey

data from the remote stations to a server located at a data center. A web-based visualization tool has been developed to allow access to data in real-time. Based on experiments, the sensor measurements have a difference of 0.872cm to 3.067cm with actual values. The higher differences tend to be associated with higher water levels while lower differences were noted for lower water levels. With further experimentation, these differences can be used as correction factors to get to a more accurate reading, specially for purposes of R&D or modeling, though for issuance of warnings to the public, these differences are small enough especially since warnings are mostly based on qualitative descriptions such as knee-deep or waist deep flood level. Flood levels are being monitored in real-time, information and warnings can be issued to the public. Participatory sensing and computer vision to estimate the flood level. Yields immediate and considerably accurate results. Image processing is currently under development. They use pressure sensor which may not last for long. GPRS may be unstable in low network coverage areas.

Chapter 3

METHODOLOGY

3.1 PROBLEM STATEMENT & PROBLEM SOLUTION

3.1.1 PROBLEM STATEMENT

Every year, the flood destroys lives and valuable resources throughout the world. The aim of the project is to develop a flash flood warning system to monitor the water level rise in rivers which are prone to severe flood. Since the existing systems are not much accurate.

3.1.2 PROBLEM SOLUTION

So with this system we can accurately predict the impact of flood and send warning/alert for evacuation. We used raspberry pi 3, HD webcam to monitor the water level. All the operations are carried out in raspberry pi 3.

3.2 MODULES

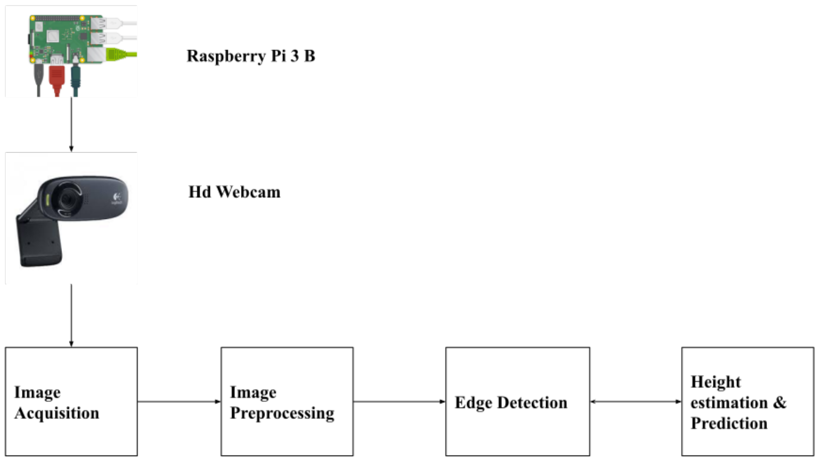

Fig. 3.1: Outline of the Proposed System

The main objective of our project is to develop a flash flood warning system to monitor the water level rise in rivers which are prone to severe flood. Here we develop a low-cost, low-power system using a Raspberry Pi camera to detect the rising water level. We employ image processing, edge detection and prediction method to detect the rising water level and predict the time of impact.

In the hardware section we use Raspberry-Pi, which has main tasks as an image processor and do an update to the system. OpenCV library is used as Image Processing Software. Some method which we use in the project are Region of Interest, Edge detection, Grayscale and Threshold etc. By using these methods, the system can read and monitor the water level. If the water level exceeds the specified level, this system will generate an early warning of impending floods by doing update timeline of water level conditions.

3.2.1 IMAGE ACQUISITION

Image processing is defined as the action of retrieving an image from some source, usually a hardware-based source for processing. It is the first step in the work flow sequence

(a) Raspberry Pi 3

(b) HD Webcam

because, without an image, no processing is possible.The image that is acquired is completely unprocessed.

Here the captured time-variant images of rising/receding water level in the stream is produced. The raspberry pi camera connected to a Raspberry Pi which takes real time video of the flowing river, which can be used for monitoring. The most crucial task for our work is being able to capture time-variant images of rising/receding water level in the stream. As our system is to be deployed in the wild, so we needed a portable image capturing unit. Therefore, we encased raspberry pi micro-computer and a camera into the watertight enclosure with its battery supply. The raspberry pi camera connected to a Raspberry Pi which takes a time-lapse picture with an increment of one minute between images.

3.2.2 IMAGE PREPROCESSING

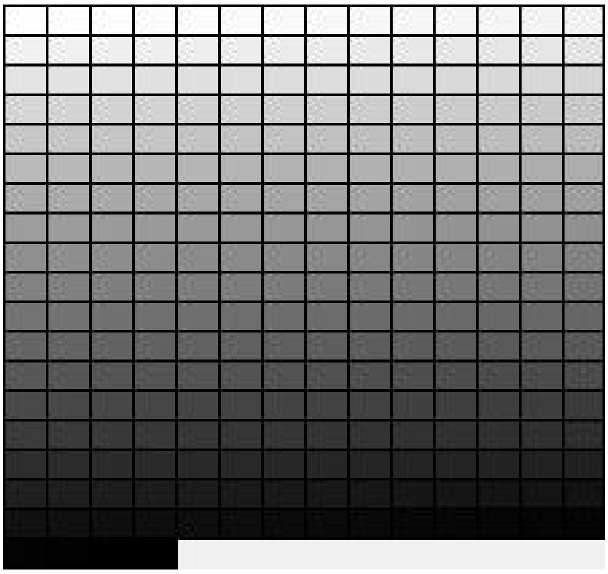

Fig. 3.3: Grayscale palette

The aim of pre processing is an improvement of the image data that suppresses unwilling distortions.This include Grayscale conversion of input image and ROI trimming . Gray scale color space codes information related to the intensity variation in the image. Through ROI, a portion of the frame which we want to filter or perform some other operations are obtained. A region of interest (ROI) is a subset of an image identified for a particular purpose. The ROI is defined by given boundaries on the image. We have the option of using a function select ROI that is natively part of OpenCV.

REGION OF INTEREST SELECTION

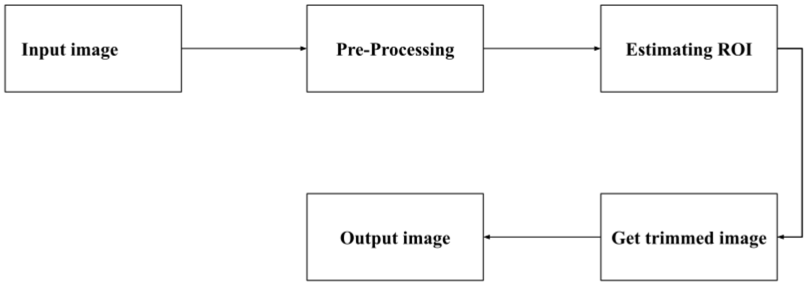

Fig. 3.4: Region of interest selection

As **selectROI** is part of the tracking API, you need to have OpenCV 3.0 (or above) installed with opencv_contrib. It allows you to select a rectangle in an image, crop the rectangular region and finally display the cropped image. A ROI allows us to operate on a rectangular subset of the image. It improves accuracy and performance (because we search for a small area).

3.2.3 EDGE DETECTION

For the purpose of the edge detection canny edge algorithm is chosen.The canny edge detector describes the isolation of the most dominant intensity differences in an image where strongly emphasized and thinned out edges will finally be delivered in the end of the process. Edge detection is done by using canny edge detection algorithm. Edges of any objects can be traced and present on a picture using canny edge detection algorithm

Canny Edge Detection

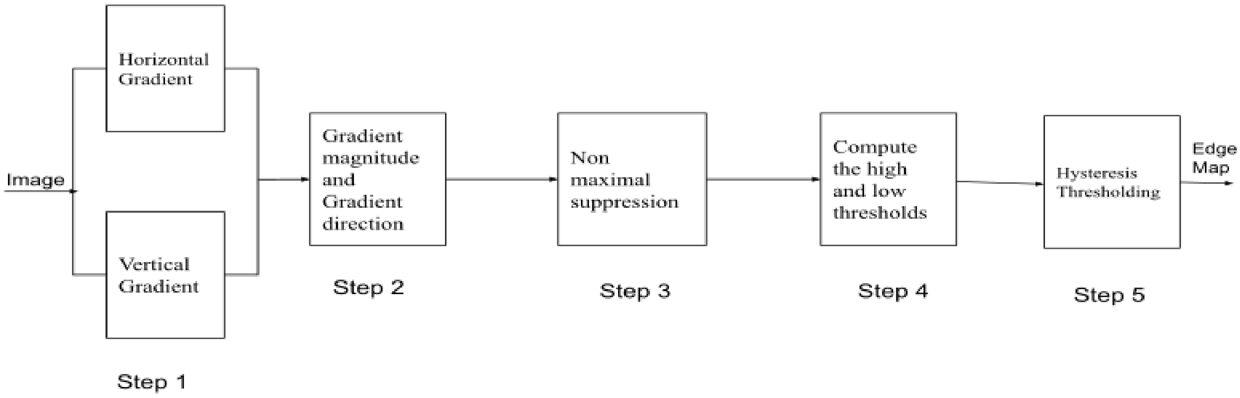

Fig. 3.5: Block Diagram of Canny Edge Detection

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images.Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. It has been widely applied in various computer vision systems. Canny has found that the requirements for the application of edge detection on diverse vision systems are relatively similar. Thus, an edge detection solution to address these requirements can be implemented in a wide range of situations. The general criteria for edge detection include:

1. Detection of edge with low error rate, which means that the detection should accurately catch as many edges shown in the image as possible

2. The edge point detected from the operator should accurately localize on the center of the edge.

3. A given edge in the image should only be marked once, and where possible, image noise should not create false edges.

To satisfy these requirements Canny used the calculus of variations a technique which finds the function which optimizes a given functional. The optimal function in Canny's detector is described by the sum of four exponential terms, but it can be approximated by the first derivative of a Gaussian.

Among the edge detection methods developed so far, Canny edge detection algorithm is one of the most strictly defined methods that provides good and reliable detection. Owing to its optimality to meet with the three criteria for edge detection and the simplicity of process for implementation, it became one of the most popular algorithms for edge detection. The Process of Canny edge detection algorithm can be broken down to 5 different steps:

1. Apply Gaussian filter to smooth the image in order to remove the noise
2. Find the intensity gradients of the image
3. Apply non-maximum suppression to get rid of spurious response to edge detection
4. Apply double threshold to determine potential edges
5. Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

The Canny algorithm contains a number of adjustable parameters, which can affect the computation time and effectiveness of the algorithm.

The size of the Gaussian filter: the smoothing filter used in the first stage directly affects the results of the Canny algorithm. Smaller filters cause less blurring, and allow detection of small, sharp lines. A larger filter causes more blurring, smearing out the value of a given pixel over a larger area of the image. Larger blurring radii are more useful for detecting larger, smoother edges for instance, the edge of a rainbow.

Thresholds: The use of two thresholds with hysteresis allows more flexibility than in a single-threshold approach, but general problems of thresholding approaches still apply. A threshold set too high can miss important information. On the other hand, a threshold set too low will falsely identify irrelevant information (such as noise) as important. It is difficult to give a generic threshold that works well on all images. No tried and tested approach to this problem yet exists.

3.2.4 HOUGH LINE TRANSFORM

The HoughLine Transform can be used in image processing to detect any shape and can be represented in mathematical form. Lines can be detected using Houghline Transform. For this purpose we need the edge detected frame.

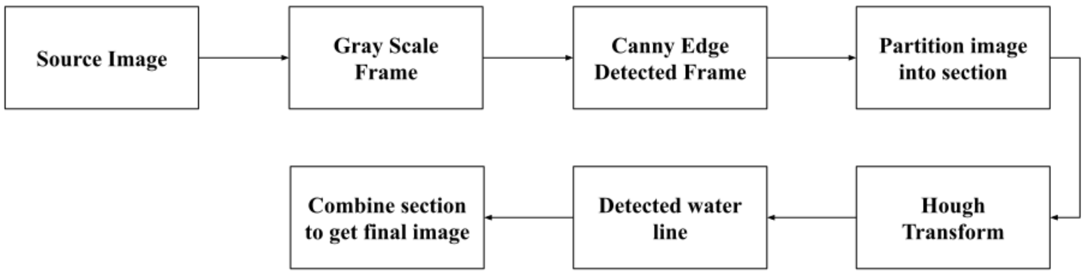

Fig. 3.6: Block Diagram of Houghline Tranform

Hough Transform is a popular technique to detect any shape, if you can represent that shape in mathematical form. It can detect the shape even if it is broken or distorted a little bit. We will see how it works for a line. In the hough transform, even for a line with two arguments, it takes a lot of computation. Probabilistic Hough Transform is an optimization of Hough Transform. It doesnt take all the points into consideration, instead take only a random subset of points and that is sufficient for line detection. Just we have to decrease the threshold. OpenCV implementation is based on Robust Detection of Lines using the Progressive Probabilistic Hough Transform ; The function used is cv2.HoughLinesP(). It has two new arguments: **minLineLength** - Minimum length of line. Line segments shorter than this are rejected. **maxLineGap** - Maximum allowed gap between line segments to treat them as single line.

3.2.5 HEIGHT ESTIMATION & PREDICTION

Height of the water level can be analyzed using the HoughLine Transform. Linear Regression is a machine learning algorithm based on supervised learning. It performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output), ie, we can plot a graph on the basis of the input and output.

In simple linear regression, we predict scores on one variable from the scores on a second variable. The variable we are predicting is called the criterion variable and is referred to as Y. The variable we are basing our predictions on is called the predictor variable and is referred to as X. When there is only one predictor variable, the prediction method is called simple regression. Linear regression consists of finding the best-fitting straight line through the points. The best-fitting line is called a regression line. Using regression to make predictions doesnt necessarily involve predicting the future. Instead, you predict the mean of the dependent variable given specific values of the dependent variable(s). For our example, well use one independent variable to predict the dependent variable.

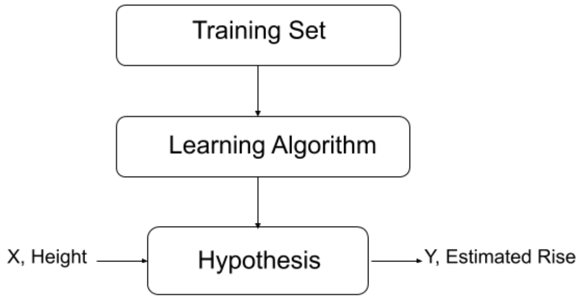

Fig. 3.7: Block Diagram of Height estimation

Regression analysis is a set of statistical processes for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables (or 'predictors'). The case of one explanatory variable is called simple linear regression. The x(independent variable) and y(dependent variable) coordinates are height and time respectively. According to the changes in the coordinates different points are plotted on the graph. From this,the best among them are joined together to form a straight line. So, formed is the predicted graph. This is implemented as the training model. Most commonly, regression analysis estimates the conditional expectation of the dependent variable given the independent variables that is, the average value of the dependent variable when the independent variables are fixed. Less commonly, the focus is on a quantile, or other location parameter of the conditional distribution of the dependent variable given the independent variables. In all cases, a function of the independent variables called the regression function is to be estimated. In regression analysis, it is also of interest to characterize the variation of the dependent variable around the prediction of the regression function using a probability distribution. A related but distinct approach is Necessary Condition Analysis which estimates the maximum (rather than average) value of the dependent variable for a given value of the independent variable in order to identify what value of the independent variable is necessary but not sufficient for a given value of the dependent variable. Regression analysis is widely used for prediction and forecasting, where its use has substantial overlap with the field of machine learning. Regression analysis is also used to understand which among the independent variables are related to the dependent variable, and to explore the forms of these relationships. In restricted circumstances, regression analysis can be used to infer causal relationships between the independent and dependent variables. However this can lead to illusions or false relationships, so caution is advisable.

3.3 SOFTWARE & HARDWARE REQUIREMENTS

3.3.1 SOFTWARE

- Software : Python 3.7+, OpenCV 4.0+, numpy, scipy, matplotlib, scikit learn
- Operating System : Windows, Raspbian
- Platform : Pycharm Community 2019.1
- Presentation & Documentation : Overleaf

3.3.2 HARDWARE

- Raspberry Pi 3
- HD webcam
- Processor : 1.2 GHz
- RAM : 1 GB
- Disk Space : 32GB

Chapter 4

IMPLEMENTATIONS

Python is used as the platform for the implementation of the proposed project. It is a software under the terms of the GNU General Public License. Python can also be used for implementation of this code.

4.1 IMAGE ACQUISITION

Here we use a Logitech HD Webcam connected to Raspberry Pi in order to capture the time-variant images of rising water level in the stream. This takes real time video of the flowing river, which can be used for monitoring. As our system is to be deployed in the wild, so we needed a portable image capturing unit. Therefore, we encased raspberry pi micro-computer and a camera into the watertight enclosure with its battery supply. The raspberry pi camera connected to a Raspberry Pi which takes a time-lapse picture with an increment of one minute between images.

VideoCapture() : cap = cv2.VideoCapture(0) :- This function captures the real time frames from the camera. And the parameter will return the video from the first webcam on your computer.

cap.read() : while(1) ret, frame = cap.read():- This code initiates an infinite loop, where we have ret and frame being defined as the cap.read(). Basically, ret is a boolean regarding whether or not there was a return at all, at the frame is each frame that is returned. If there is no frame, you wont get an error, you will get None.

4.2 IMAGE PRE-PROCESSING

In this module, two methods are used. Gray scale conversion of input image and Region of interest trimming. Here the image received will be unprocessed and they are represented using BGR format. This image will be converted into the models that are found to be suitable for the application. Gray scale color space code information relates to the intensity variation in the image. After gray scale conversion, ROI trimming method will be used. By using this method, unwanted edges can be eliminated. Region of interest is a sunset of an image identified for a particular purpose. The ROI is defined by giving boundaries on the image. We have the option of using a function select ROI that is natively part of OpenCV.

cv2.cvtColor() : gray = cv2.cvtColor(frame, cv2.COLOR-BGR2GRAY) :- Here, we define a new variable, gray, as the frame, converted to gray. It is important to note that OpenCV reads colors as BGR (Blue Green Red).

cv2.selectROI() : r = cv2.selectROI(frame) :- It selects ROI on the given image and the function creates a window and allows user to select a ROI using mouse. We used Region of Interest (ROI) trimming as the image pre-processing technique.When video capturing is done images are captured frame by frame. Those images are cropped by ROI trimming method. Further edge detection process is done on images which are cropped by ROI trimming method.

4.3 EDGE DETECTION

cv2.Canny() **: edges = cv2.Canny(gray, 1, 100)**. First argument is our input image. Second and third arguments are our minVal and maxVal respectively.

cv2.HoughLinesP() : lines = cv2.HoughLinesP(edges, 1, np.pi/180, 15, minLineLength, maxLineGap) :- HoughLine Transform is a popular technique to detect any shape, if you can represent that shape in mathematical form. It can detect the shape even if it is broken or distorted a little bit.The function has two arguments.

minLineLength - Minimum length of line. Line segments shorter than this are rejected.
maxLineGap - Maximum allowed gap between line segments to treat them as single line.

4.4 HEIGHT ESTIMATION AND PREDICTION

Here lets take the case of filling a glass with water. Using ROI we trim the top and bottom part of the glass and the HoughLine Transform algorithm detects the horizontal line which change according to the raise of the water. These values of the line are plotted in a graph. And this graph gives the slope which can be used for the prediction of the impact.

Linear Regression: Linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. The x(independent variable) and y(dependent variable) coordinates are height and time respectively. According to the changes in the coordinates different points are plotted on the graph. From this,the best among them are joined together to form a straight line. So,formed is the predicted graph.This is implemented as the training model.

Chapter 5

RESULT & ANALYSIS

5.1 IMAGE ACQUISITION

First the video recording starts using the HDWeb cam which is connected to the raspberry pi 3.

Fig. 5.1: Original Frame

5.2 IMAGE PRE-PROCESSING

5.2.1 REGION OF INTEREST:

Here we are manually selecting the area of interest using the ROI trimming methods. Now we drag around the area of interest and the area is selected. This crops the image according to the four corners we selected. The figures below shows the original frame and the trimmed image of the frame.

Output:

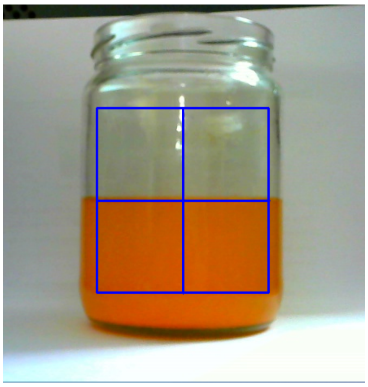

Fig. 5.2: Roi Trimming

5.3 EDGE DETECTION

Here we use the trimmed image for edge detection . The edge detecton algorithm find wide range of edges. This smooths the image inorder to remove the noise and apply non-maximum suppression to get rid of false edges. Then track the edges by hysteresis suppressing all the edges that are weak and not connected to the strong edges.Track edge by hysteresis, which finalize the detection of edge.

Output:

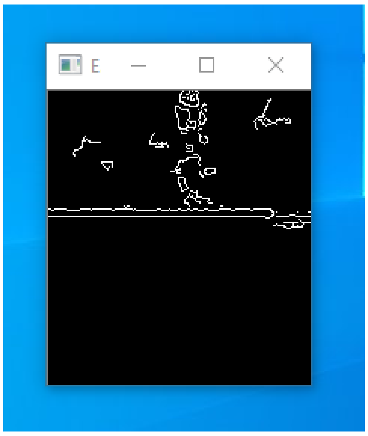

Fig. 5.3: Edge Detection

5.4 HOUGH LINE TRANSFORM

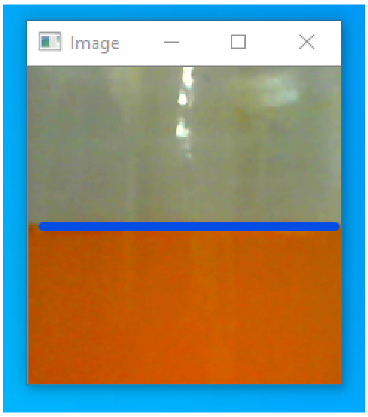

Fig. 5.4: Line detection using Hough Transform

The HoughLine Transform can be used in image processing to detect any shape and can be represented in mathematical form. Lines can be detected using Houghline Transform. For this purpose we need the edge detected frame.considering only the horizontal lines,not the vertical lines.To remove the vertival lines slope is obtained.The slope of the vertical ine does not exist.we cannot divide by zero ,which is ofcourse why this slope value is undefined.Slope a horizontal line is zero.so,only the slope ranging from 0-0.15 is considered.furteher operations are considered only when satisfying this range,else are not proceeded. Two parameteres like minimum value and maximum value of the line is determined.Those lines coming in between to the values are displayed,others are not.

5.5 HEIGHT ESTIMATION AND PREDICTION

After the edge detection and HoughLine Transform process we can clearly see the water level detected by the line. The change in the coordinates of the line is used to plot the graph. Here with the points we use the linear regression method.

Output:

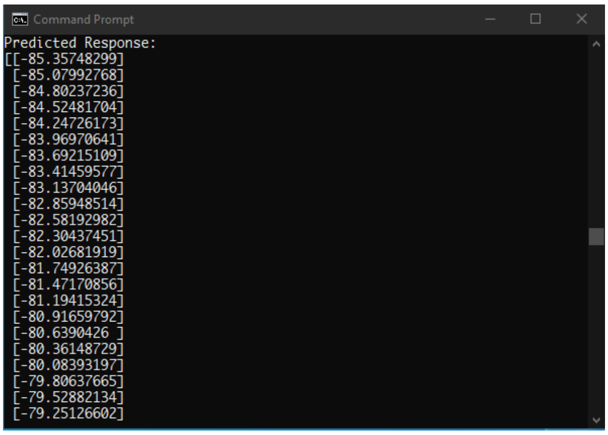

Fig. 5.5: Final Output

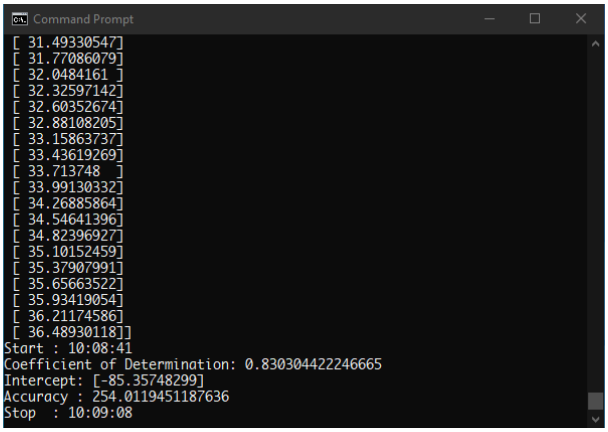

Fig. 5.6: Final Output

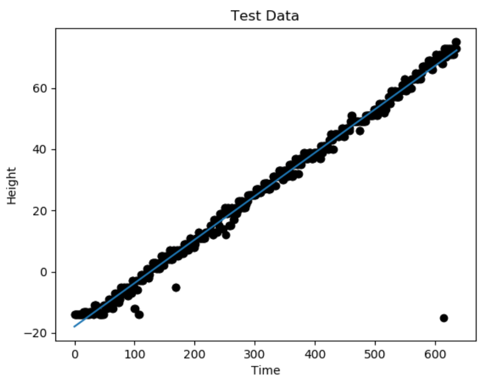

Fig. 5.7: Regression Analysis

5.6 TEST CASES

Test Cases						
Test Case ID	Test Scenario	Test Step	Test Data	Expected Result	Actual Result	Pass/Fail
01	Region of interest	Crop the portion of an image that you want to filter	Input Image	Cropped Image	As Expected	Pass
02	Canny Edge Detection	Edge to be detected	Input Video	Edge Detected Frame	As Expected	Pass
03	Hough Transform	Distorted shape/line to be detected	Input Video	Line detected frame	As Expected	Pass
04	Height Estimation and Prediction	Mark the water line using hough transform and find x,y values to plot the height, using the values plot the graph	Values from the hough lines	Height should be marked and a graph should be plotted	As Expected	Pass

Table 5.1: Test Cases

Chapter 6

CONCLUSION & SCOPE OF FUTURE WORKS

Here we describe the project flood level monitoring and early warning system via computer vision techniques. These are basically implemented under Python with the use of the OpenCV Computer Vision Library. As already mentioned, the entire application is primarily running on a raspberry pi 3 single chip-card computer. Though the setup was prototypical, we can lay a foundation for future expansion into this work. Therefore, the developed approach is very flexible in terms of adaptations and can be used under real world conditions.

Thus, an efficient and also cheap solution to a real time Flood monitoring and early warning system could be provided with plenty of potential for further improvements and optimization on the system side. These experiments were performed at the same camera distance and ambient weather condition. The effect of distance can be added in the next modification. As a future modification one could integrate the camera into an Unmanned Aerial Vehicle platform so that it can be used to analyze real time flood level by moving over water and into remote areas, which is a more complex extension. Also the predicted impact information and the height details can be made available for the public through a small ios/android app.

REFERENCES

[1] Jirapon Sunkpho and Chaiwat Ootamakorn,"Real-time flood monitoring and warning system

[2] Chang, N. and Guo Da-Hai.,"Urban Flash Flood Monitoring, Mapping and Forecasting via a Tailored Sensor Network System, Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control 2006, " *issue 23-25, pp. 757-761, April 2006*

[3] RamKumar Narayanan, VM Lekshmy, Sethuraman Rao, Kalyan Sasidhar,"A Novel Approach to Urban Flood Monitoring Using Computer Vision "*IEEE - 33044*

[4] H. H. Jaehyoung Yu,"Remote detection and monitoring of a water level using narrow band channel "*Journal of Information Science and Engineering 26,71-82, 2010.*

[5] K. Khurana and R. Awasthi,"Techniques for Object Recognition in Images and Multi-Object Detection "—2018

[6] G Natividad and J M Mendez,"Flood Monitoring and Early Warning System Using Ultrasonic Sensor "*Conf. Ser.: Mater. Sci. Eng. 325 012020 View the*

[7] Bhavana B Nair Sethuraman N Rao ,"Flood Monitoring using Computer Vision",*International Journal Of Computer Science And Applications Vol. 4, IEEE Xplore*