Department of Computer Science & Engineering
SOUTHEAST UNIVERSITY

# CSE4000: Research Methodology

## Real life object detection by using image processing techniques

*A dissertation submitted to the Southeast University in partial fulfillment of the requirements for the degree of B. Sc. in Computer Science & Engineering*

Submitted by

Md. Ibrahim Hussain
ID: 2011100000020

Supervised by

Md. Ashiqur Rahman
Assistant Professor, Department of CSE
Southeast University


May, 2016

## Letter of Transmittal

May 7, 2016

The Chairman,
Department of Computer Science & Engineering,
Southeast University,
Banani, Dhaka.

Through: Supervisor, Md. Ashiqur Rahman

Subject: Submission of Research Paper

Dear Sir,

I am glad to submit the research paper titled "Real life object detection by using image processing techniques". It was a great pleasure to work on this effective and useful topic. The research work has been accomplished as per the requirement of the Southeast University and following your instruction.

I have given best efforts to complete this Research. I request your kind approval of this research paper in fulfillment of my degree requirement.

Thank you.

Sincerely yours,

Md. Ibrahim Hussain
ID: 2011100000020
Batch: 28th Program: CSE

Supervisor:

Md. Ashiqur Rahman
Assistant Professor, Department of CSE
Southeast University

# Certificate

This is to certify that the research paper titled "Real life object detection by image processing techniques" has been submitted by Md Ibrahim Hussain to the department of Computer Science and Engineering, Southeast University, has been accepted as satisfactory for potential fulfillment of the requirements for the degree of bachelor of Science and Engineering in CSE.

| **Author:** | **Approved by the Supervisor:** |
| --- | --- |
| Md. Ibrahim Hussain | Md. Ashiqur Rahman |
| ID: 2011100000020 | Assistant Professor, Department of CSE |
| Batch: 28th Program: CSE | Southeast University |

# Abstract

Object detection by image processing techniques is very interesting area in Computer Vision. Local Binary Pattern (LBP) features are used to detect object from video. For creating training xml file We should train positive and negative image. After creating xml file we use java application for developing object detector. We should use webcam for desirable output.

# Acknowledgements

I would like to thank my supervisor, Md. Ashiqur Rahman, Assistant professor, Department of CSE, for giving me his valuable guidelines, support and encouragement. He is very helpful and good person.

Finally, I thank my family for supporting me.

# Contents

# List of Figures

# Chapter 1

# Introduction

We can identify any individual object very easily without any attempts; by taking inspiration from human the object recognition techniques should be developed. Many techniques are introduced to identify objects in real time. Image processing is a term which shows the processing on image or video frame which is taken as an input and the result set of processing is may be a set of related parameters of an image. In the image processing, object detection and tracking plays a significant role [1]. Object detection in real life is still a challenging problem. All over the world many researcher publish new papers, new algorithms about efficient object detection method every year. Huge data set for training and testing object is helpful to detect object in real life. But process huge data set need more times even weeks, also there is accuracy factor [2].

## 1.1 Motivation

Local Binary Pattern (LBP) features are very powerful local descriptor used in object detection technology. LBP feature is widely used in texture analysis, target detecting and tracking, face recognition analysis, product quality analysis, surface inspection, etc. It is an effective image feature which transforms an image into an order or image of integer labels relating tiny-scale presences of the images. These labels or statistic histogram, are then applied for further image analysis [3].

## 1.2 Objectives

I used LBP features for detect object. LPB features are better than other features. It is very fast and simple to implement. In my research there are positives image that is include my object images and negatives image, that is without my object images. I trained them and detect by webcam. If I quickly train and detect my selected object then I can say I reached my goal.

# Chapter 2

# Literature Review

There are many kinds of works that already have done in these fields. I describe my related work shortly and I gave those references are going interest for my work.

## 2.1 Face Detection

The Working Method of Jesorsky et al. is edge-based and works on gray scale still image. This work describes an efficient implementation, making this approach suitable for real time applications [4]. Using Gabor filter feature extraction techniques in image processing for face detected Kausal and Raina describe based on arrays of the feature. Gabor filters applied input of the classifier for feature vector, which is a FFNN (Feed Forward Neural Network) on a feature subspace adept by an approach easier than PCA [5].

## 2.2 Movement Detection

Movement detection is a concept where moving objects are identified in a sequence of images. This is interesting in machine vision applications as traffic monitoring, human detection and military applications. At the University of Oulu a M. Heikkilä, M Pietikäinen and J. Heikkilä has published a paper on movement detection and background extraction by using local binary patterns. The method first divides each frame into equally sized blocks (or stationary windows). Each block overlaps each other which helps preserve the shape of the moving object. The method uses circularly defined LBP patterns where the values are estimated with bi-linear interpolation. The main strength of the LBP in this application is that it is invariant to illumination changes [6].

## 2.3 Object Counting and Monitoring

In video counting human or in a microscopic image estimating the number of cells, a learning framework for visual object counting proposed by Zisserman [7]. Electric technologies have excited the imagination of video security professional like intelligent video analytic. Through constant automated digital screening and filtering, video analytic can identify specific conditions and notify operators of potential situations, allowing security officials to make quick, informed decisions [8].

## 2.4 Bio-medical Imaging Techniques

Barbu reviews the application of artificial neural networks in medical image processing, in medical image object detection and recognition. Main advantages and drawbacks of artificial neural networks were discussed. Using neural networks survey try to asked what the main failing and potency for medical image processing [9]. Bio is developed A practical modification of the Hough transform is proposed that of law-contrast circular objects. For enhance skill and computational perplexity algorithm, Circular Hough Transfrom and its huge alternations are talked and assimilated. To verify the algorithm medical images are used. A phase contrast microscope is used for localizing cell muclei of cytological smears visualized by applying particular algorithm [10].

## 2.5 Vehicle Detection and Recognition

Existing literature in vehicle detection, counting and type recognition proposes a number of different approaches. Ye Li, Bo Li, Bin Tian, and Qingming Yao showed that even in a congested road traffic condition an AND-OR graph (AOG) using bottom-up inference can be used to represent and detect vehicle objects based on both front and rear views [11]. In a similar environment, Ehsan Adeli Mosabbeb, Maryam Sadeghi, and Mahmoud Fathy proposed the use of strong shadows as a feature to detect the presence of vehicles in a congested environment [12]. Vehicles were partitioned into three parts; road, head and body, using a tripwire technique. Subsequently Haar wavelet features extracted from each part with PCA performed on features calculated to form 3 category PCA-subspaces. Further, Multiple Discriminate Analysis (MDA) is performed on each PCA-subspace to extract features, which are subsequently trained to identify vehicles using the Hidden Markov Model-Expectation Maximization (HMMEM) algorithm [13, 14].

## 2.6 Manufacturing Quality Control

Aguilera et al describes an automatic grading system using artificial vision to improve the quality of wood panel's surfaces. The objective is to control stains on the surface. Artificial vision techniques like Thresholding and Transformed watershed methods are applied. In special intensity, area, distribution and quantity-there faults quantitative measures construct on the surface [15]. For design intelligent signal, Scotti and Piuri tried to the improvement of systems that attempt to complete all actions and visual inspection have been recognized in the literature as a good tool which can be used by the designer to achieve these goals [16].

## 2.7 Content Based Image Retrieval

Li and Shapiro address the need for object recognition in content based image retrieval. It discuss the type of images features necessary for recognition of common objects senses [17]. Agarwal and Roth present an approach for learning to detect objects. Interested object class of a set of sample images is automatically constructed for a vocabulary of information object parts. From this vocabulary images are represented with spatial relations are observed [18].

## 2.8 Visual Positioning and Tracking

In Mohedano et al, A novel 3D people tracing and positioning system is proposed, where it's strong to both possible occlusion sources: stable scene objects and other human. The system holds on a set of multiple cameras with particularly overlapped ends of view [19]. Liao and chu describe which is mainly based on a digital map of a GIS (geographical information system) and marks the location of an object on the map. An approach called GODTA (GPS-based object detection and tracking Approach) is proposed in this paper. GODTA is mainly based on the coordinate transformation method [20].

## 2.9 Global Robot Localization

Gohring and Homan describe the connection of detect image to other objects in similar camera image for basis calculating the location of the object in a global coordinate system [21]. Anti et Al describes new representation of the query images that consists of a vector of the detection scores for each object class was made. This work is a modern work for Global Robot Localization [22].

## 2.10 Optical Character Recognition

Alata and Al-Shab presents an algorithm and software to detect and recognize character in an image. These types fronts were under investigation, namely, Verdana, Arial and Lucida Console [23]. For solving bangla basic character problem, twenty famous bangle fronts have been used for character recognition. It was first time used multi-front bangla character recognition. A proposal landmark extraction scheme founded on the digital curvelet transform is described [24].

# Chapter 3

# The Basics of Local Binary Patterns

Ojala et al. introduced genuine local binary pattern feature in 1995 [25]. The local binary patern operator is an operator that tells the environments of a pixel by creating a bit-code from the binary derivatives of a pixel. The LBP features are commonly used to grey scale images and the derivative of the severities. It only works on a 3 x 3 pixel block of a gray image. As mark in Figure 3.1, for each 3 x 3 block, basic LBP can be defined as an ordered set of pixel intensities binary comparisons between the center pixel and its eight neighbor pixels. Where the center pixel's value is greater than the neighbor's value, tick "1". Otherwise, tick "0". These eight neighbor pixels labeled by a circle order (i.e. clockwise or counter-clockwise), start from top-left pixel if clockwise, or left pixel if counter-clockwise. Since the neighborhood only consists of eight pixels, a total of $2^8 = 256$ different kinds of codes can be obtained from 0 to 255 if code indicated by decimal code [3].

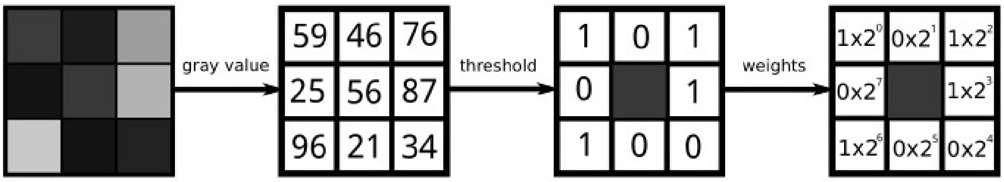



Figure 3.1: Basic LBP Operator

LBP binary code:01001101, LBP decimal code:1 x $2^0$ + 0 x $2^1$ + 1 x $2^2$ + 1 x $2^3$ + 0 x $2^4$ + 0 x $2^5$ + 1 x $2^6$ + 0 x $2^7$ = 77

As follows,

$$\mathrm{S}(\mathrm{f}_p - \mathrm{f}_c) = \begin{cases} 1, & iff_p \geq \mathrm{f}_c \\ 0, & otherwise \end{cases}$$

$$\mathrm{LBP} = \sum_{p=0}^{p-1} S(f_p - \mathrm{f}_c)2^p$$

Here,

- $\mathrm{f}_p$ = gray levels of a sampling point around $\mathrm{f}_p$ with the same frequency
- $\mathrm{f}_c$ = center pixel's gray value
- p = 0,...., P-1
- P = the number of sampling points
- $2^p$ = binomial weight

## 3.1 Generic LBP Operator

Several years later original publish paper revised by Ojala et al [26]. Original version LBP operator only works 3 x 3 pixel blocks. Generic LBP use neighborhoods of different sizes. Generic LBP extended rectangular to circular domain, which accept arbitrary number of neighborhoods in R distance. It also works on gray image.

Consider an image I(x,y) and $f_c$ is gray value of an arbitrary pixel (x,y). Also consider $f_p$ means gray levels of a sampling point, P is the number of sampling points and radius R [27]. Number of neighbors with center as Threshold denote by T.

$f_c = I(x,y)$

$f_p = I(x_p, y_p),$

$x_p = x + R\cos(2\pi p/P)$

$y_p = y + R\sin(2\pi p/P)$

Consider the coordinate of pixel should be integer, a bi-linearly interrupted method was applied when the coordinates of neighborhood pixel

$f_p$ $((x + R\cos(2\pi p/P), y + R\sin(2\pi p/P))$ are non integer.

Suppose local texture of the image I(x,y) is characterized by joint distribution of gray values of P + 1 (P>0) pixels:

$T = t(f_c, f_0, f_1, ...., f_{P-1})$

Where $f_1, ...., f_{P-1}$ corresponds to the P number of neighbor's grey-value. Without loss of information, the center pixel value can be subtracted from the neighborhood:

$T = t(f_c, f_0 - f_c, f_1 - f_c, ..., f_{P-1} - f_c)$

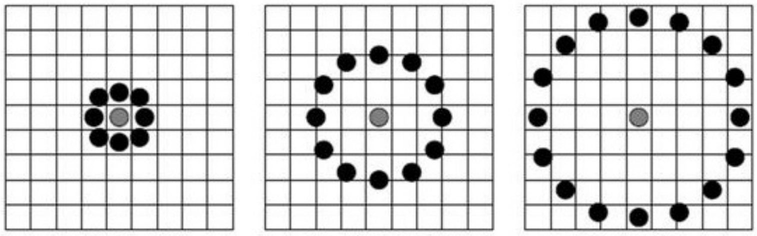



Figure 3.2: Generic LBP operators with (sampling points P, radius R)

Now joint distribution is approximated by assuming the center pixel to be statistically independent of the differences, which allows for factorization of the distribution:

$T \approx t(f_c), t(f_0 - f_c, f_1 - f_c, ..., f_{P-1} - f_c)$

Since $t(f_c)$ describe the overall luminance of image, which is unrelated local textual patterns- means it contains no useful information. But

$t(f_0 - f_c, f_1 - f_c, ..., f_{P-1} - f_c)$

for modeling the local texture associated format of differences can be used. Although reliable presumption of this multidimensional ordination from image data can be hard. For this solution Ojala et al. proposed to apply vector quantization. It runs learning vector quantization of a codebook to minimize the high dimensional feature space [27].

The learning vector quantization is still difficult. Therefore the signs of the differences are considered. It says a point on the circle has a higher gray value than the center pixel (or similar value), a one is marked to that point, and otherwise it gets a zero:

$T \approx (s(f_0 - f_c), s(f_1 - f_c), ...., s(f_{P-1} - f_c))$

where s(x) is the thresholding (step) function

$$\text{s(x)} = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

The generic local binary pattern operator is derived from this joint distribution [27]. For basic LBP, it is assumed by adding the threshold distinctions weighted by powers of two.

The $\text{LBP}_{P,R}$ operator is defined as

$$\text{LBP}_{P,R}(\text{x}_c,\text{y}_c) = \sum_{p=0}^{p-1} s(f_p \text{ - f}_c)2^p$$

Here, create the LBP for pixel ($\text{x}_c$ , $\text{y}_c$ ) a binomial weight $2^p$ is assigned to each sign s($\text{f}_p$ - $\text{f}_c$). Symbols of the distinctions in a neighborhood are explained as a P-bit binary number, become in $2^p$ distinct values for the LBP code. The local gray-scale distribution, i.e. texture, can thus be nearly described with a $2^p$ -bin discrete format of LBP features:

$$\text{T} \approx \text{t}(\text{LBP}_{P,R}(x_c ,\text{y}_c ))$$

For counting the $\text{LBP}_{P,R}$ distribution for a certain D x C image sample ($\text{x}_c$ $\epsilon$ 0,...,D - 1, $\text{y}_c$ $\epsilon$ 0,...,C - 1), the central part is only considered because a sufficiently large neighborhood cannot be used on the borders. The Local Binary Pattern features is counted for each pixel in the cut portion of the image, and the ordering of the feature is used as a feature vector, denoted by S:

$$\text{S} = \text{t}(\text{LBP}_{P,R} \text{ (x,y)}),$$

$x \epsilon \lceil R \rceil, \ldots, D - 1 - \lceil R \rceil$, $y \epsilon \lceil R \rceil, \ldots, C - 1 - \lceil R \rceil$

The original LBP (Fig. 3.1) is very similar to LBP 8,1 , with two differences. First, the neighborhood in the general definition is indexed circularly, making it easier to derive rotation invariant texture descriptors. Second, the diagonal pixels in the 3x3 neighborhood are interpolated in LBP 8,1 [28].

# Chapter 4

# Implementation

## 4.1 Object detection Algorithm

Input: Training Image set.

Output: Feature extracted from object image and compared with centre pixel and recognition with unknown object image.

1. Initialize temp = 0

2. FOR each image I in the training image set

3. Initialize the pattern histogram, H = 0

4. FOR each center pixel tc $\epsilon$ I

5. Compute the pattern label of tc , LBP(1)

6. Increase the corresponding bin by 1.

7. END FOR

8. END FOR

9. Find the highest LBP feature for each object image and combined into single vector.

10. Compare with test object image.

11. If it match it most similar object in database then successfully recognized

## 4.2 Flowchart for Object Detection

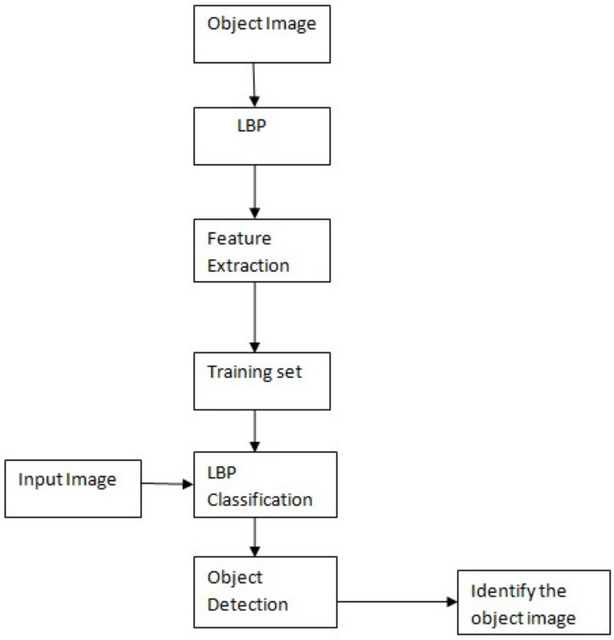



Fig 4.1: Flowchart for Object Detection

## 4.3 Application of LBP

LBP feature is widely used in

- texture analysis,
- target detecting and tracking,
- face recognition analysis,
- product quality analysis,
- Surface inspection,
- Movement Detection etc.

## 4.4 Required Tools and software

OS: Windows 8.1 Pro 64 Bit

Java 8u65

OpenCV 3.01

IDE: NetBeans 8.0.2

JavaFX Scene Builder 2.0

Webcam

ImageBatch

IfranView 4.40

Positive Image and negative Image (Background image)

## 4.5 Positive Image and Negative Image


Fig 4.2: Positive image

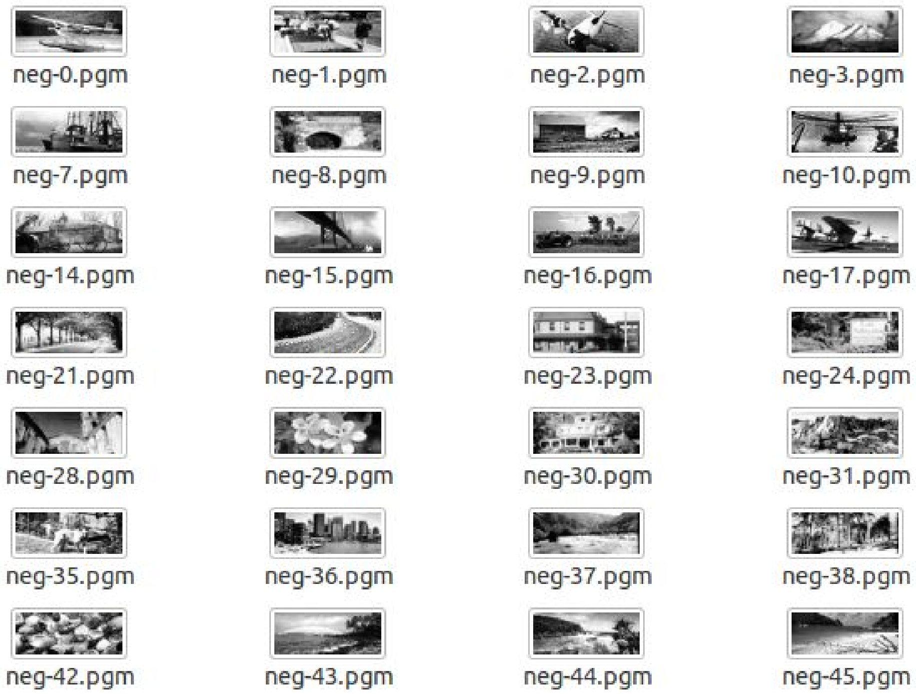



Fig 4.3: Negative image

## 4.6 Create sample vector and Train cascade command

Create Sample vector file command:

C:\Users\bijoy\Desktop\caropencv\opencv_createsamples.exe -info info.txt -num 550 -w 48 -h 24 -vec cars.vec

Train object file command:

C:\Users\bijoy\Desktop\caropencv\opencv_traincascade.exe -data data -vec cars.vec -bg bg.txt -numPos 500 -numNeg 500 -numStages 16 -w 48 -h 24 -featureType LBP

Here,

cars.vec file that i have created with createsamples.exe

info.txt list of positives pgm from /pos/

-bg bg.txt list of negatives pgm files from /neg/

-numPos 500 number of positive pgm files

-numNeg 500 number of negative pgm files

-nstages 16 number of stages to complete

-w 48 -h 24 sample size- width 48, height 24

Note: When create vector file then positive image number must be greater. Also when create vector file and train object then image width and height should be smaller then original image size.

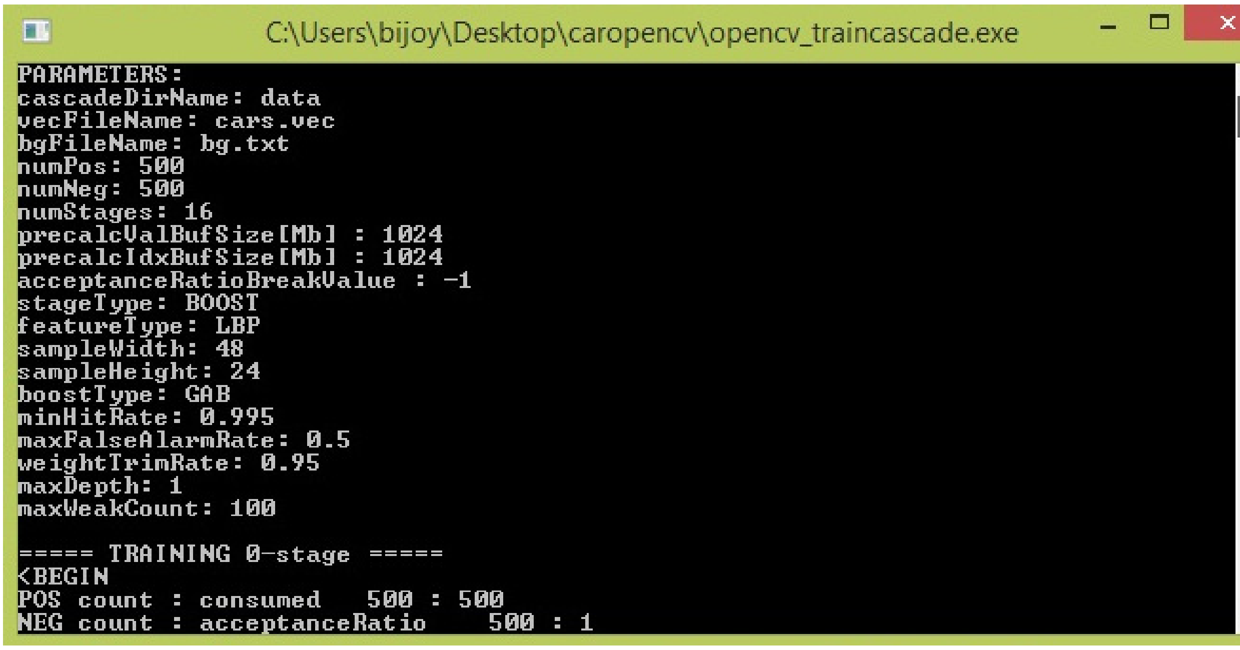
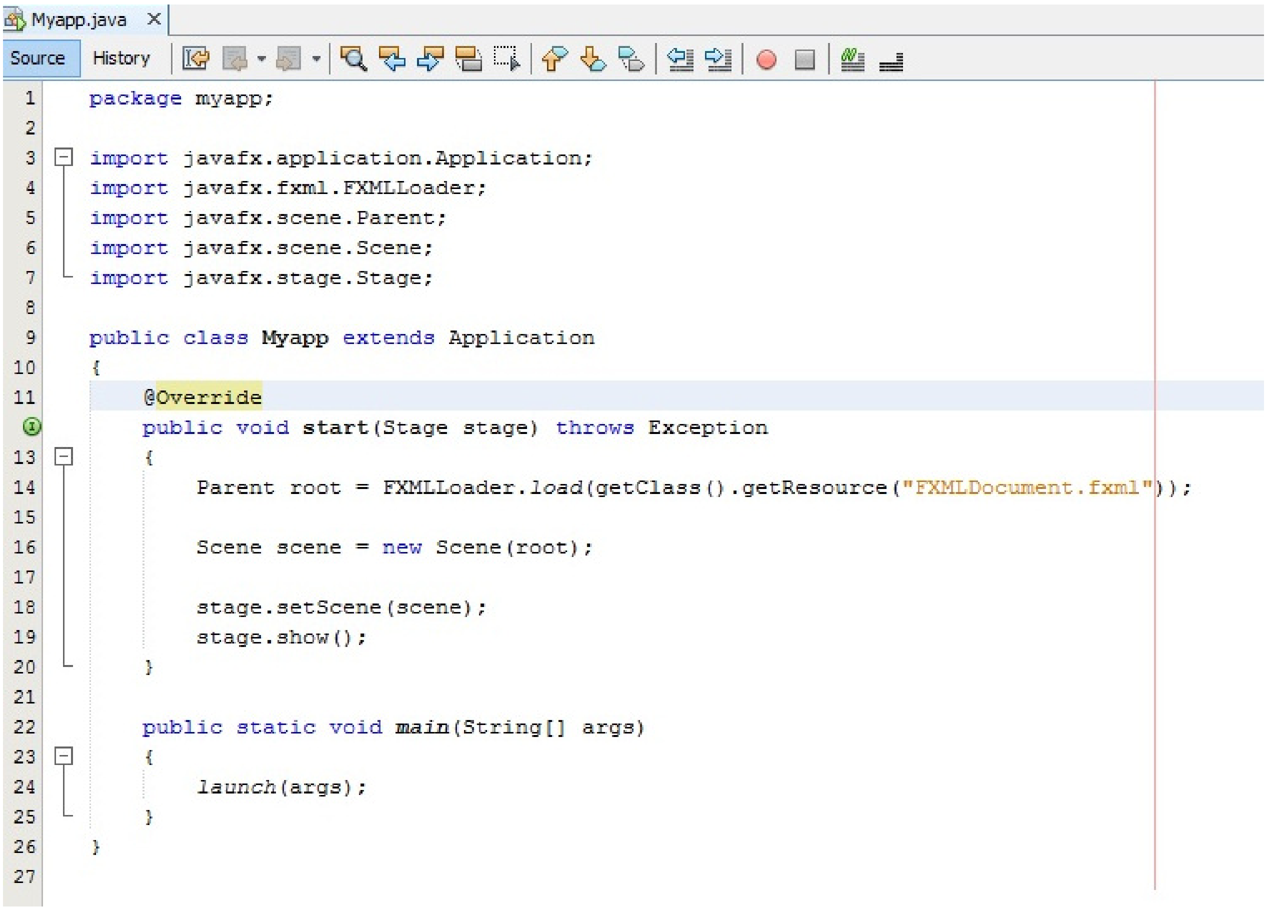



Fig 4.4: Object image training

## 4.7 Code

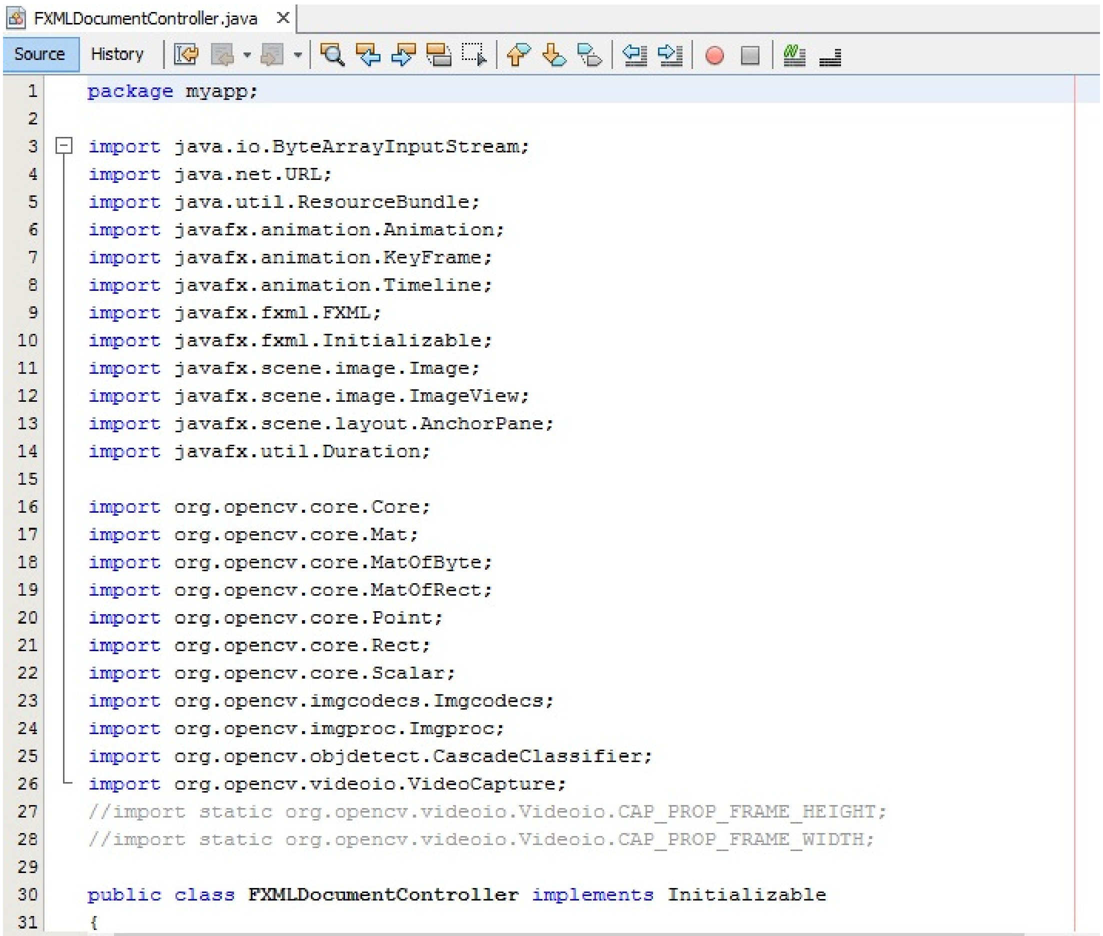
Main java package source code



```java
package myapp;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class Myapp extends Application
{
    @Override
    public void start(Stage stage) throws Exception
    {
        Parent root = FXMLLoader.load(getClass().getResource("FXMLDocument.fxml"));

        Scene scene = new Scene(root);

        stage.setScene(scene);
        stage.show();
    }

    public static void main(String[] args)
    {
        launch(args);
    }
}
```

Figure 4.5: Main java package source code

FXML Document Controller source code:



```
package myapp;

import java.io.ByteArrayInputStream;
import java.net.URL;
import java.util.ResourceBundle;
import javafx.animation.Animation;
import javafx.animation.KeyFrame;
import javafx.animation.Timeline;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.layout.AnchorPane;
import javafx.util.Duration;

import org.opencv.core.Core;
import org.opencv.core.Mat;
import org.opencv.core.MatOfByte;
import org.opencv.core.MatOfRect;
import org.opencv.core.Point;
import org.opencv.core.Rect;
import org.opencv.core.Scalar;
import org.opencv.imgcodecs.Imgcodecs;
import org.opencv.imgproc.Imgproc;
import org.opencv.objdetect.CascadeClassifier;
import org.opencv.videoio.VideoCapture;
//import static org.opencv.videoio.Videoio.CAP_PROP_FRAME_HEIGHT;
//import static org.opencv.videoio.Videoio.CAP_PROP_FRAME_WIDTH;

public class FXMLDocumentController implements Initializable
{
```

Figure 4.6: FXML Document Controller source code

```java
    private MatOfByte memory;
    private Mat mat;
    private VideoCapture cap;
    private Image image1;
    private MatOfRect obj;
    private CascadeClassifier objdetect;

    @FXML
    private AnchorPane ap;
    @FXML
    private ImageView p1;

    private Image Mat2BufferedImage(Mat m)
    {
        memory = new MatOfByte();
        Imgcodecs.imencode(".bmp", m, memory);
        return (new Image(new ByteArrayInputStream(memory.toArray())));
    }

    @Override
    public void initialize(URL url, ResourceBundle rb)
    {
        System.loadLibrary(Core.NATIVE_LIBRARY_NAME);
        mat = new Mat();
        obj = new MatOfRect();
        objdetect = new CascadeClassifier("cascade.xml");
        p1 = new ImageView();

        //p1.setFitWidth(640);
        //p1.setFitHeight(480);
```

Figure 4.7: FXML Document Controller source code

```java

 cap = new VideoCapture();
 //cap.set(CAP_PROP_FRAME_WIDTH, 640);
//cap.set(CAP_PROP_FRAME_HEIGHT, 480);
 cap.open(0);

 Timeline timeline = new Timeline(new KeyFrame(Duration.millis(1000), ev -> {
if(cap.isOpened())
{
cap.read(mat);
if(!mat.empty())
{
//image1 = Mat2BufferedImage(mat);
objdetect.detectMultiScale(mat, obj);
for(Rect rect : obj.toArray())
{
    Imgproc.rectangle(mat, new Point(rect.x, rect.y), new Point(rect.x + rect.width, rect.y + rect.height),
            new Scalar(0,255,0));
}
image1 = Mat2BufferedImage(mat);
p1.setImage(image1);
}
}
}));
timeline.setCycleCount(Animation.INDEFINITE);
timeline.play();
 ap.getChildren().add(p1);
 }
   }
```

Figure 4.8: FXML Document Controller source code

FXML Document source code:

FXMLDocument.fxml

```xml
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.image.*?>
<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<AnchorPane id="AnchorPane" fx:id="ap" prefHeight="346.0" prefWidth="543.0"
    xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="myapp.FXMLDocumentController">
   <children>
      <ImageView fx:id="p1" fitHeight="185.0" fitWidth="237.0" layoutX="292.0" layoutY="14.0" pickOnBounds="true" preserveRatio="true" />
   </children>
</AnchorPane>
```

Figure 4.9: FXML Document source code

# Chapter 5

# Result and Evaluation

Fig 5.1: Output

We got desirable output. In my research, We tried to detect object by using image processing techniques. We used LBP features. Although We faced some difficulties, especially when train data set. Good data set and better algorithm We should get better output. For train data set Linux is better and image format portable gray map (pgm) is suitable.

# Chapter 6

# Conclusion

We have achieved our research goal. We used local binary patterns (LBP) feature to detect object. For further development We should use better algorithm and good data set for detect object. Using LBP + Adaboost features We should get good result.

# Appendix A

CV: Computer Vision

FFNN: Feed Forward Neural Network

GIS: Geographical Information System

GODTA: GPS-based Object Detection and Tracking Approach

LBP: Local Binary Pattern

OCR: Optical Character Recognition

PCA: Principal Component Analysis

# Bibliography


[1] S. Jain and S. Chadokar, "A Object Detection in Image Processing: A Review," *International Journal of Electrical, Electronics and Computer Engineering*, vol. 4, no. 2, pp. 26–29, 2015.

[2] M. P. L. L.-T. Frank Dellaert, Jan-Michael Frahm and B. Rosenhahn, "Outdoor and Large-Scale Real-World Scene Analysis," p. 243, 2011.

[3] D. Zhou, *Real-time Animal Detection System for Intelligent Vehicles*. Master Thesis, Faculty of Engineering, University of Ottawa, 2014.

[4] K. J. K. e. Olive R Jesorsky and R. ert W. Frischholz, *Robust face detection using the hausdor distance*. Berlin, Germany: BioID AG, 2011.

[5] A. Kaushal and J. P. S. Raina, "Fcae detection using neural network and gabor wavelet tarnsfrom," 2010.

[6] M. M. Heikkilä and J. Heikkilä, *A Texture-based Method for Detecting Moving Objects*. Finland: Machine Vision Group, Infotech Oulu Dept. of Electrical and Information Engineering, University of Oulu, 2004.

[7] A. Zisserman, *Learning to count objects in images vector lempitsky*. Visual Geometry Group, University of Oxford.

[8] S. Electric, *Video analytics extending security with technology*. Worldwide Headquarters 3500 pelco Way Clovis, California 93612 USA: Pelco, Inc., 2011.

[9] A. Barbu, *Marginal space learning for fast object detection in medical imaging*. Statistics Department, Florida State University, 2009.

[10] M. Smereka and I. Duleba, *Circular object detection using a modified hough transform*. Janiszewskiego 11/17, 50-372 Wroclaw, Poland: Institute of computer Engineering, Control and Robotics, 2008.

[11] B. T. Bo Li and Q. Yao, "A Vehicle detection based on the and-or graph for congested traffic conditions," *Intelligent Transportation Systems*, vol. 14, no. 2, pp. 984–993, 2013.

[12] M. S. Ehsan Adeli Mosabbeb and M. Fathy, *A new approach for vehicle detection in congested traffic scenes based on strong shadow segmentation*. Springer, 2007.

[13] H. M. Ming Yin, Hao Zhang and X. Wang, *An hmm-based algorithm for vehicle detection in congested traffic situations.* IEEE, 2007.

[14] M. E. Irhebhude, "Object detection, recognition and re-identification in video footage," Ph.D. dissertation, 2015.

[15] M. R. Cristhian Aguilera and G. Roa, "An automatic grading system for panels surfaces using artificial vision," *International Journal of Computers, Communications and control*, vol. I, no. 2, pp. 15–22, 2006.

[16] V. Piuri and F. Scotti, *Computational intelligence in industrial quality control.* via Bramante 65, 26013 Crema (CR), Italy: Department of information Technologies, 2005.

[17] Y. Li and L. G. Shapiro, *Object recognition for content-based image retrieval.* Seattle, WA 89195-2350: Dept. of Computer Science and Engineering, University of Washington.

[18] S. Agarwal and D. Roth, *Learning a sparse representation for object detection.* IL 61801, USA: Department of Computer Science, University of Illinois at Urbana-Champion Urbana, 2002.

[19] F. J. Raul Mohedano, Carlos R. del Blanco and L. Salgado, *Robust 3d people tracking and positioning system in a semi-overlapped multi-camera environment.* 28040, Madrid, Spain: Universidad politecnica de Madrid, 2008.

[20] H.-C. Liao and P.-T. Chu, "A novel visual tracking approach incorporating global positioning system in a ubiquitous camera environment," *Information Technology Journal*, vol. 8, no. 4, pp. 465–475, 2009.

[21] D. Gohring and J. Homann, *Sensor modeling using visual object relation in multi robot object tracking.* Zu Berlin Unter den Linden 6, 10099 Berlin, Germany: Institut fur informatik LFG Kunstliche Intelligenz Humboldt-Universit, 2009.

[22] K. G. D. Roy Anati, Davide Scaramuzza and K. Daniilidis, "Robot localization using soft object detection," 2012.

[23] M. Alata and M. Al-Shabi, "Text detection and character recognition using fuzzy image processing," *Journal of Electrical Engineering*, vol. 57, no. 5, pp. 258–267, 2007.

[24] M. Angshul, "Bangla Basic character recognition using digital curvelet transform," *Journal of Pattern Recognition Research*, vol. 2, no. 1, 2007.

[25] M. P. Timo Ojala and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[26] P. M. Ojala T and M. T, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," vol. 24, no. 7, pp. 971–987, 2002.

[27] T. W. Md. Abdur Rahim, Md. Najmul Hossain and M. S. Azam, "Face Recognition using Local Binary Patterns (LBP)," vol. 13, no. 4, 2013.

[28] G. M. Pietikäinen et al, Abdenour Hadi and Timo Ahonen, "Computer Vision Using Local Binary Patterns," vol. 40, pp. 15–16, 2011.