ANIMAL DETECTION ON ROADS USING DIGITAL IMAGE PROCESSING

A Project report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

A.V.G. Vandhana (319126512065) V. Sumanth(319126512125)

V. Asha Jyothi (319126512066)

Under the guidance of

Mr.A.SIVA KUMAR

Assistant Professor,Mtech,(Ph.D)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(UGC AUTONOMOUS)

(*Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC*)

Sangivalasa, bheemili mandal, visakhapatnam dist.(A.P)

2022-2023

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)
(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC)
Sangivalasa, Bheemili mandal, Visakhapatnam dist.(A.P)

PRAGNANAM BRAHMA
ANITS

CERTIFICATE

This is to certify that the project report entitled **"ANIMAL DETECTION ON ROADS USING DIGITAL IMAGE PROCESSING"** submitted by A.V.G. Vandhana (319126512065), V. Asha Jyothi (319126512066), V. Sumanth (319126512125) in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Electronics & Communication Engineering** of Anil Neerukonda Institute of technology and Sciences(A), Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

Project Guide

Mr.A. Siva Kumar

Assistant Professor,Mtech,(Ph.D)
Department of ECE
ANITS

Assistant Professor
Department of E.C.E
Anil Neerukonda
Institute of Technology & Sciences
Sangivalasa, Visakhapatnam-531 162

Head of Department

Dr. B.Jagadeesh

Professor&HOD
Department of ECE
ANITS

Head of the Department
Department of E C E
Anil Neerukonda Institute of Technology & Sciences
Sangivalasa - 531 162

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **Mr.A.SivaKumar** Mtech,(Ph.D),Assistant Professor, Department of Electronics and Communication Engineering, ANITS, for his/her guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr. B.Jagadeesh**, Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa,** for their encouragement and cooperation to carry out this work.

We express our thanks to all **teaching faculty** of Department of ECE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank **all non-teaching staff** of the Department of ECE, ANITS for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

A.V.G. Vandhana (319126512065)

V. Asha Jyothi (319126512066)

V. Sumanth (319126512125)

ABSTRACT

Road accidents are one of the major problems that cause deaths, disabilities, and hospitalisations for people worldwide. This type of collision occurs mainly during night time because the animal’s moments are more at night. Modern automotive design methods rely on security features, comfort features, and safety precautions. The top priority in an automobile is safety, so we should take the steps necessary to reduce collisions and accidents. Road accidents are increasing because there weren't any intelligence-based system for alerts and road safety. This method is inexpensive for the automatic spotting of animals on automatically identifying collisions using image processing techniques. The vehicle mechanism with the camera mounted is employed for this approach of animal detection. The proposed system is trained on more images taken from videos of animals in various vehicle types travelling on motorways speeds. This prevents animal-vehicle collisions by finding the vehicle's proximity to the animal and the animal itself. This results in giving an automatic indication to the driver .So that the brakes are applied. The obtained results are efficient and accurate. This will alert the driver and prevent road accidents.

CONTENTS

ABSTRACT

LIST OF FIGURES

LIST OF ABBREVIATIONS

CHAPTER 1 : INTRODUCTION

1.1 Project Objective 3

1.2 Project Outline 4

CHAPTER 2 : IMAGE PROCESSING

2.1 Introduction 6

2.2 Purpose of Image Processing 6

2.3 Types of Image Processing 7

2.4 Working Diagram of Image Processing 8

2.5 Characteristics of Image Processing 9

2.6 Advantages of Image Processing 10

CHAPTER 3 : FEATURE DESCRIPTOR

3.1 Introduction 12

3.2 Dimensions of the Feature Space 13

3.3 Types of Feature Descriptors 14

3.4 Feature Matching Workflow 14

3.5 Detect Interest Points 15

3.6 Applications of Feature Descriptor 16

CHAPTER 4 : HOG

4.1 Introduction 18

4.2 Calculating Gradients 19

4.2.1 Pre-process the Data(64 x 128) 19

4.2.2 Gradients in X and Y Directions 20

4.3 Creation of Histograms 22

4.4 Calculating Histogram of Gradients 25

4.5 Normalization 26

4.6 Final Stage 27

4.7 Uses of HOG 28

CHAPTER 5 : MATLAB

5.1 Introduction 30

5.2 Matlab System 31

5.3 Matlab’s Power of Computational Mathematics 33

5.4 Features of Matlab 33

5.5 Uses of Matlab 34

CHAPTER 6 : METHODOLOGY

6.1 Introduction 36

6.2 Flow Chart 37

6.3 Image Acquisition 38

6.4 Pre-processsing 38

6.5 Object Detection 40

6.6 Feature Extraction 41

6.6.1 Steps in HOG 41

6.6.2 Creation of Histogram 42

6.6.3 Normalization 43

6.6.4 Final Stage 44

6.7 Distance Calculation 45

6.8 Message Alert 46

RESULTS AND DISCUSSION 48

CONCLUSION 53

REFERENCES 54

LIST OF FIGURES

Fig 2.1	working diagram of image processing	8
Fig 3.1	visual representation of the feature vectors	15
Fig 3.2	Getting matched points by matched feature	16
Fig 4.1	Example of HOG feature descriptor	19
Fig 4.2	Input image of HOG	19
Fig 4.3	Small patch of the image	20
Fig 4.4	Matrix of the pixel values	20
Fig 4.5	Magnitude and orientation by Pythagoras theorem	21
Fig 4.6	Creation of histogram	23
Fig 4.7	Histogram is created	25
Fig 4.8	Dividing the image into 8×8 cells	26
Fig 4.9	Final stage of HOG	28
Fig 5.1	Matlab Application Program	32
Fig 6.1	Flow chart	37
Fig 6.2	Block Diagram of Normalization	43
Fig 6.3	Distance calculated	46
Fig 7.1	Input image	48
Fig 7.2	Filtered image	49
Fig 7.3	Object detection map	49
Fig 7.4	Object detection image	49
Fig 7.5	Object cropped image	50
Fig 7.6	Gray image	50
Fig 7.7	Gradient image	50
Fig 7.8	Detected image	51
Fig 7.9	Input image	51
Fig 7.10	Output image	51

LIST OF ABBREVATIONS

JPEG	Joint Photographic Experts Group
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Features
LBP	Local Binary Pattern
BRISK	Binary Robust Invariant Scalable Key Points
MSER	Maximally stable extremal regions
HOG	Histogram of oriented gradients
HOF	Histogram of Optical Flow
MBH	Motion Boundary Histogram
GPU	Graphics processing unit

CHAPTER 1: INTRODUCTION

1. INTRODUCTION

One of the main issues that contributes to human mortality, disability, and hospitalisation is traffic accidents. According to the World Health Organization, at least one in ten individuals are dead on roads worldwide are from India. The strategies used in modern automotive design are based on security features, comfort mechanisms, and safety precautions. We should take action to lessen crashes and accidents because the safety of an automobile is our top priority. Accidents still occur at a rate of 1.24 million annually. One such significant issue is the roadside accident of an animal and a vehicle. Because the animal spends more time at night, this type of collision happens more frequently at night. Two animal species, the cow and the dog, are more frequently seen on highways than other kinds of animals. This is a low-cost method for employing image processing to detect animals automatically on highways in order to reduce animal-vehicle collisions. Because there isn't a sophisticated highway safety and alert system, traffic accidents are on the rise. They were primarily influenced by three things: people, vehicle, and infrastructure. Since they were caused by animals on the road, the infrastructure is to blame for many incidents. Because of driver negligence and the lack of any alarm mechanism to prevent collisions, human factor also plays a role in many accidents. Therefore, a major goal of our initiative is to decrease the number of animal-vehicle collisions on public roads. We employ image processing techniques to locate animals at specific distances, identify them, and automatically inform drivers to their presence on the road with alert messages. So that the alarm message prompts the application of the brakes. The motorist will be warned, and an animal-vehicle collision will be avoided. This approach is suggested for discovering the animals using a vehicle with a camera installed. The suggested method is trained using additional images from video clips of animals travelling at various vehicle speeds on highways. By locating the animal and measuring the distance between the vehicle and animal, this prevents animal-car collisions and automatically alerts the driver. Consequently, brakes are engaged. This will warn the motorist and stop traffic collisions.

1.1 Project Objective

Our project's main goal is to reduce animal-vehicle collisions. The preprocessing work for the video format input received from the camera interface is entirely handled by our suggested system, which is based on image processing. A moving animal is visible in the video, which was captured by a forward-facing optical sensor camera and excludes other stationary and moving objects. The computer stores this footage and transforms it into various frames. Pre-processing procedures are carried out once the image is acquired. performed to improve the image. Image enhancement occurs during the preprocessing phase. Object detection comes next after preprocessing. Following object detection, feature extraction is used we turn the collected images into grayscale while doing edge detection. We are combining classifier with enhanced cascade with HOG classifier with enhanced cascade for extracting animal detection features and system learning. The gathered results are put through classifier with enhanced cascade the algorithm for animal detection see if any animals show up in the photographs that were taken. Finding the animal's separation from the test vehicle is the next step once the animal has been identified in the video. The separation between any discovered animals and the testing vehicle on the road is estimated using the pixels from the image that was collected. Three different types of information are sent to the driver depending on how close or how far an animal is to the camera mounted vehicle. These three categories are: animal is near, animal is little far, and animal is extremely far. This automatically alerts the motorist when they are close enough to the roadway. This will warn the motorist and stop traffic collisions.in order to warn the driver in order for him to apply the brakes in time to prevent a collision between the animal and the car. Animal-vehicle collisions and accidents are avoided as a result.

1.2 Objective outline

Applications based on animal detection are extremely important in solving real-world issues. Animal detection in the video or image serves as the foundation for the majority of applications. While driving, humans must be aware of whether they can drive their vehicle with a response time of 150 milliseconds or not. The problem is that human eyes quickly tyre and require rest, so we should suggest some preventative measures to get around this.

We looked into some of the previously suggested techniques, such as face detection, that require the animals to strike a position in front of the camera to activate. The issue is the drawback of this approach is that face detection demands that animals gaze directly into the lens, which isn't always possible in footage of road trips. Animals can come in a variety of shapes, sizes, and colors, as well as in varied positions.

Using an understanding of their mobility, one may identify animals. Although this method works well in controlled environments, such as underwater footage, it is particularly ineffective when used with road or highway side film. Given the availability of facial detection technology, this approach of animal detection was deemed inappropriate. So, in order to identify the animals and avoid vehicle-animal collisions, we developed a system.

So, using image processing, we created an animal vehicle accident to address all of these issues. Therefore, a major goal of our initiative is to decrease the number of animal-vehicle collisions on public roads. We employ image processing techniques to locate animals at specific distances, identify them, and automatically inform drivers to their presence on the road with alert messages. so that the alarm message prompts the application of the brakes. The motorist will be warned, and an animal-vehicle collision will be avoided. This approach is suggested for discovering the animals using a vehicle with a camera installed. The suggested method is trained using additional images from video clips of animals travelling at various vehicle speeds on highways. By locating the animal and measuring the distance between the vehicle and animal, this prevents animal-car collisions and automatically alerts the driver. Consequently, brakes are engaged. This will warn the motorist and stop traffic collisions.

CHAPTER 2: IMAGE PROCESSING

2. IMAGE PROCESSING

2.1 Introduction

Image processing is the process of translating a physical image to digital form so that it can be changed, added to, or information extracted from it. It is a signal-distribution technique where a picture, such as one from a video frame or a photograph, serves as the input and an additional image or the properties connected with it serves as the output. The majority of the time, image processing algorithms treat images as two-dimensional signals, which are then processed using predefined signal processing techniques. It is currently one of the technologies with the quickest growth rates, with uses across numerous industries. It has applications in numerous industries and is currently one of the technologies with the quickest growth rates. Image processing is a key area of study in the fields of engineering and computer science.

Image processing basically contains the three stages listed below:

- Making use of an optical or digital scanner import the picture.
- Analyzing and modifying images, such as through data compression and image. enhancement, as well as identifying patterns that are invisible to the human eye, such as in satellite images.
- The final phase, known as output, is where a picture or report based on image analysis may be changed.

2.2 Purpose of Image processing

There are five different categories for image processing. These are what they are:

1. Visualization - Pay attention to non-visible objects.
2. Image repair and sharpening - To make the image better.
3. Image retrieval - Locate the desired image.
4. Pattern measuring - Counts the number of items in an image.
5. Recognition of images - Recognize objects in images.

2.3 Types of Image Processing

There are two methods for processing images techniques: analogue and digital. Analogue or visual approaches is usable to process physical copies, including prints and photographs. Image analysts use various interpretational fundamentals while using these visual techniques. Image processing is limited not only to the area that needs to be explored, but also to analyst knowledge. Association is yet another crucial factor aspect of image processing using visual approaches. Analysts process images by integrating their knowledge with that of others.

The adoption of digital processing techniques allows for computer-based digital image editing. Because of imperfections in the raw data from the satellite platform's image sensors, information must go through a variety of processing phases to overcome these flaws and achieve originality. All data types must pass through three general phases when using digital techniques: pre-processing, augmentation and display, and information extra.

2.4 Working diagram of Image Processing

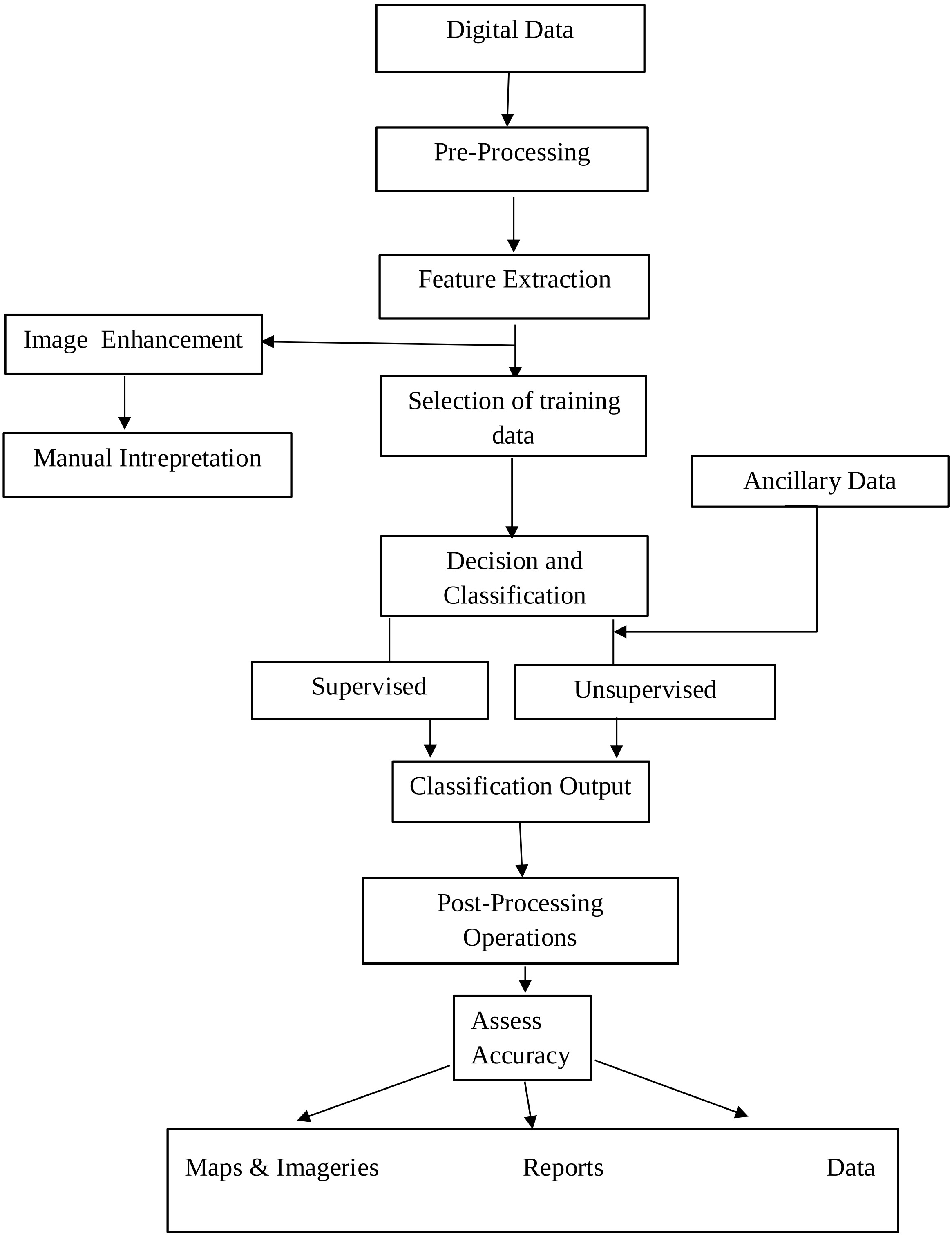


Fig 2.1 working diagram of image processing

2.5 Characteristics of Image Processing

An image is translated into a digital format before processing. Sampling an image and quantizing the sampled values are both parts of digitization. After the image has been transformed into bit information, processing takes place. picture enhancement, picture compression, and image restoration are some of these processing techniques.

Image improvement:

It means emphasising or boosting picture elements, such as boundaries or contrast, to increase a graphic display's suitability for analysis and exhibition. This procedure does not boost the data's inherent informative content. It entails adjustments to the grey level and contrast, filtering, interpolation and magnification, noise reduction, edge crispening and sharpening, pseudo coloring, and other processes.

Image restoration:

Filtering the observed image is what it is concerned with doing in order to lessen the impact of degradations. The degree and accuracy of the information about the deterioration process, as well as the filter design, determine how effectively an image may be restored. The difference between picture restoration and image enhancement is that the latter focuses more on extracting or emphasising image features.

Image compression:

The idea is to use as few bits as possible to depict a picture. Compression is used in a variety of applications, including broadcast television, satellite remote sensing, military aircraft communication, radar, teleconferencing, facsimile transfer, papers for business and education, images from digital radiography, magnetic resonance imaging, and computer tomography, movies, satellite photos, weather maps, and more.

- CCITT GROUP 3 & GROUP 4 text compression
- JPEG image compression for still images
- MPEG video image compression

2.6 Advantages of Image Processing

- Image Processing is quicker and more economical. Process duration is minimized, as is the amount of film and other photographic equipment required.
- Image processing is less harmful to the environment. Digital photography does not require the use of processing or fixing chemicals. However, printing inks are necessary when printing digital photographs.
- When taking a digital photograph, it is immediately clear whether or not the image is good.
- It is straightforward to copy a digital image, and the image quality remains consistent unless compressed. For example, saving an image the image is compressed in the jpg format. When saved in jpg format, the compressed image is recompressed, and the image quality declines with each save.
- Image repair and retouching is now easier. An updated Healing Brush Too in Photoshop 7 can smooth out face wrinkles in seconds.
- When rastering the image an imitation camera, the expensive reproduction is faster and less expensive.
- The image can be utilised in a different media by modifying picture resolution and format.

CHAPTER 3: FEATURE DESCRIPTOR

3. FEATURE DESCRIPTOR

3.1 Introduction

A feature descriptor is a technique for extracting descriptions of a feature for a site of interest or the entire image. By putting intriguing information into a string of numbers, feature descriptors act as a kind of numerical "fingerprint" that we can use to differentiate one feature from another. Any piece of information that is essential for performing the calculations required for a specific application. Features are specific elements in an such as points, edges, or objects in the image. A quality descriptor is information extracted from numerical numbers derived from visuals that are challenging for people to understand and correlate. Feature are the elements that make up a picture if it is a data representation, information derived from the data. A feature vector is a vector with one or more dimensions that serves as a mathematical representation of a feature description.

The feature vector is a linear vector that stores the feature descriptor along a multidimensional feature space, whereas the feature descriptor can be any verbal or mathematically logical representation of an interest point. A feature vector is a vector that contains different pieces of information about an object. By mixing object feature vectors, we can produce a feature space. A single pixel or an entire image may be represented by the features taken as a whole. The level of depth needed in the feature vector depends on what we want to understand or represent about the item. It is believed that the data is redundant when there is too much to process. The information's features can be reduced using this stage. Using the feature selection technique, a subset of the initial features is chosen. With this method, the initial data's complexity is reduced and the user can complete the intended task without relying entirely on the first data set.

The process of feature extraction makes data explanations less resource-intensive, which minimises the complexity of the data. It enhances the model's accuracy and stops errors from being produced. Data analysis on significant amounts of data typically necessitates a lot of computational power and memory. Enhancing the results of various design endeavours through the use of feature engineering methodologies.

3.2 Dimensions of the Feature Space

A distinguishing feature vector can be found at the n-dimensional feature space, at any position. For example, a two-dimensional feature space will have two scaler dimensions. Then, on this two-dimensional space, we can place any pair of two scalers. After assigning a meaning to each dimension, we convert this area into a description space for all possible concatenations of those meanings. The amount of data columns used to indicate variables like location, color, intensity, neighborhood, and so on is known as dimensionality. When categorising or clustering the data, we must decide which dimensions and columns to use in order to produce usable information. With the level of detail we want to express, the feature space's dimension count rises. The feature vector is a point N-dimensional space feature space. Consider the following example: one of the feature vectors is a three-dimensional coordinate in a two- or three-dimensional physical space. In image processing, these vectors could, however, cover hundreds of dimensions.

The N-dimensional feature space point is the feature vector. Consider a simple scenario where a three-dimensional coordinate is contained in one of the feature vectors in a two- or three-dimensional physical space. However, in image processing, these vectors may cover hundreds of dimensions.

A feature vector can be thought of as a long list of scalers, each representing a unique way to characterise the item being seen. We can transform a feature vector to an embedding because the properties of the feature space are commonly related. A relatively low-dimensional space called an embedding can be created from high-dimensional vectors.

Similarity Measures for Feature Vectors:

The distance in Euclid between the vectors' ends, the angle between the vectors cosines, and the dot product (cosine times the lengths of both vectors), are the three comparisons between vectors that are most frequently made. Sending these feature vectors to a similarity measure causes it to provide a similarity score for them.

An interest point in one image can now be compared to an interest point in another image to see how similar they are. This is something that we deliberately aimed to address in the correspondence problem. Similarity metrics and feature vectors are frequently used in image processing and computer vision.

3.3 Types of Feature Descriptors

Depending on the application, we can extract two types of features from images. They are defined by both local and global factors. We commonly employ global features for low-level applications such as item identifying and classifying. Characteristics are used for more advanced applications such as item identification. The combination of global and local features enhances recognition accuracy while increasing processing costs.

There is a significant distinction between identification and detection. Detection is the method of finding something or detecting an object in a picture. Recognizing a thing is the process of determining its identity.

Local Descriptors:

Local information is used to portray the texture of an image. characteristics. A local descriptor describes a patch within a picture. It is more resilient to match a picture with many local descriptors than with only one. Local descriptors include SIFT, SURF, LBP, BRISK, MSER, and FREAK.

Global Descriptors:

Local features explain the important portions of a picture of an object, whereas global features represent the complete image. Examples of global features include contour representations, shape descriptors, and texture features.

Global descriptors include shape matrices, invariant moments, optical flow histograms (HOF), motion boundary histograms, and histograms oriented towards histograms (HOG), (MBH).

3.4 Feature Matching Workflow

Finding the method is based on the reference and target's point correspondences images used in this example to identify a specific object. It can still detect objects even if the scale or in-plane rotation are changed. Additionally, it can withstand certain out-of-plane rotation and occlusion.

The image processing method seeks to confidently locate the object in the scene image, which is a target image. There is no information on whether the object is visible in the image or not. The only presumption is that the object appears only once in the image, if it does so at all.

3.5 Detect Interest Points

Finding the interest points is the initial stage in the feature matching workflow. Even though there may . There are countless areas of interest in the scene, we sometimes just take the top few hundred results into consideration. Since ,only a small number to locate the object with confidence, a certain number of exact matches are needed. This is mostly done to lower the computing needs. Identifying the focal points of the two photographs:

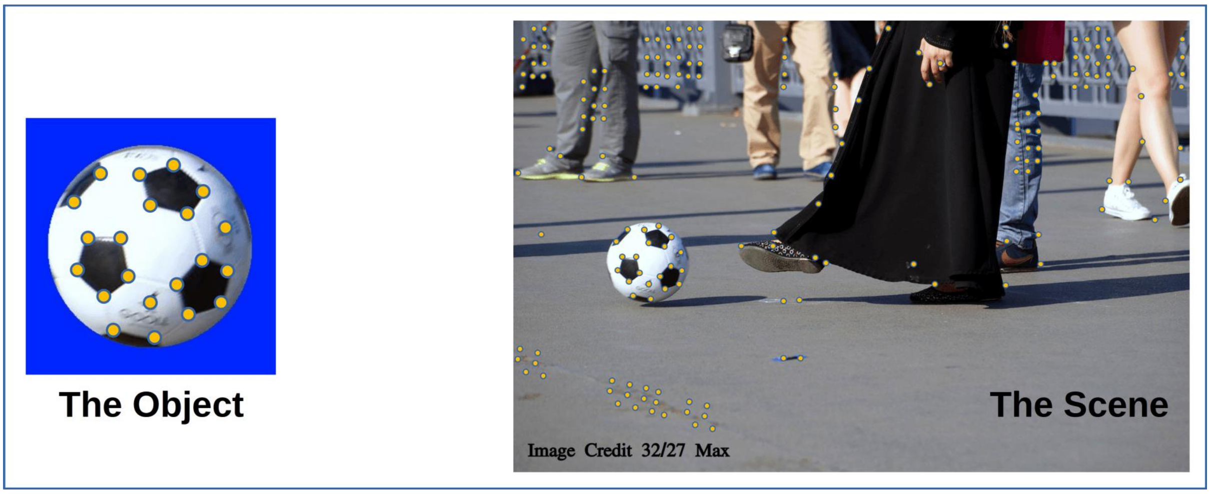


Fig 3.1 visual representation of the feature vectors

The following step is to compare the feature vectors for proximity after having them. One of the feature vector similarity measures is commonly used for this. The primary rule is that the best match must be different from the second-best match. Substantially, after the feature matching technique, we remove outliers using geometric information.. under the assumption that it is a rigid body that has not undergone any physical alterations. The object is depicted in the scene image at a different scale or rotation. By using this data, we can eliminate the outliers. Find the transformation that connects the corresponding points. while eliminating outliers. This update allows us to localise the object in the scene:

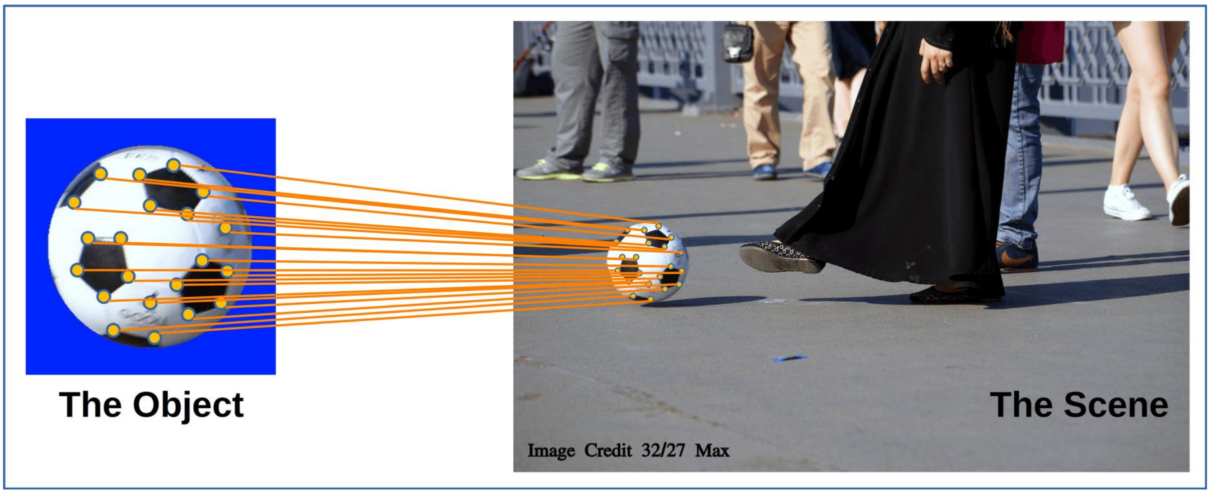


Fig 3.2 Getting matched points by matched feature

3.6 APPLICATIONS OF FEATURE DESCRIPTOR

• Using a text extraction solution, you may automatically detect and extract features from huge amounts of text data and produce a summary of all distinctive features or a mix of features.

• It's important for brands and marketers to comprehend the needs of their target market. To gather insightful knowledge and develop products with a compelling collection of features, use text extraction.

• By automatically detecting entities in a product specification, recommendation systems can find other goods that have entities that are comparable to them. You can create different groups of features by clustering similar or related features.

• Identifying new client segments or product attributes to create more individualised and alluring offerings.

• You might learn new things by making it easier to locate, arrange, and retrieve Pertinent Information.

CHAPTER 4: HOG (HISTOGRAM OF ORIENTED GRADIENTS)

4. HOG (HISTOGRAM OF ORIENTED GRADIENTS)

4.1 Introduction

HOG is a feature descriptor related to the SIFT Canny Edge Detector. It is used to detect objects in computer vision and image processing. Within a specific image area, this method determines how many instances with gradient orientation. Edge Orientation Histograms, Scale Invariant Feature Transformation (SIFT), and other techniques are extremely similar to this one and shape contexts, but it varies in that it is computed on a dense grid of uniformly spaced cells and employs the local contrast overlaps normalisation for greater accuracy.

The HOG description accentuates an object's composition or form. It outperforms because it computes the features using both the magnitude and angle of the gradient, it outperforms all other edge descriptors. Based on the size and direction of the gradient, it produces histograms for the various portions of the image.

The core concept of the distribution of intensity gradients or edge directions can be described by the histogram of directed gradients, and utilised to determine the look and shape of local objects inside an image. Each pixel in the picture's distinct, contiguous sections known as cells, a histogram of gradient axes is constructed. The description is formed by combining these histograms.

By measuring the intensity over a larger portion of the image, known as a block, and using this value to normalise all cells within the block, local histograms can be contrast-normalized to increase accuracy. This normalisation improves invariance to variations in light and shadows.

HOG creates a feature vector from an image of (width x height x dimension) that has a spatial dimension that is substantially less than the original image. As a result, HOG can be operated without a strong GPU because it requires less computational resources to detect small-scaled pictures.

HOG's computation time is long when distinguishing an item or large-scaled photos because it extracts attributes from each image pixel using a sliding window approach. As a result, the accuracy is not very reliable when compared to current convolutional neural networks..

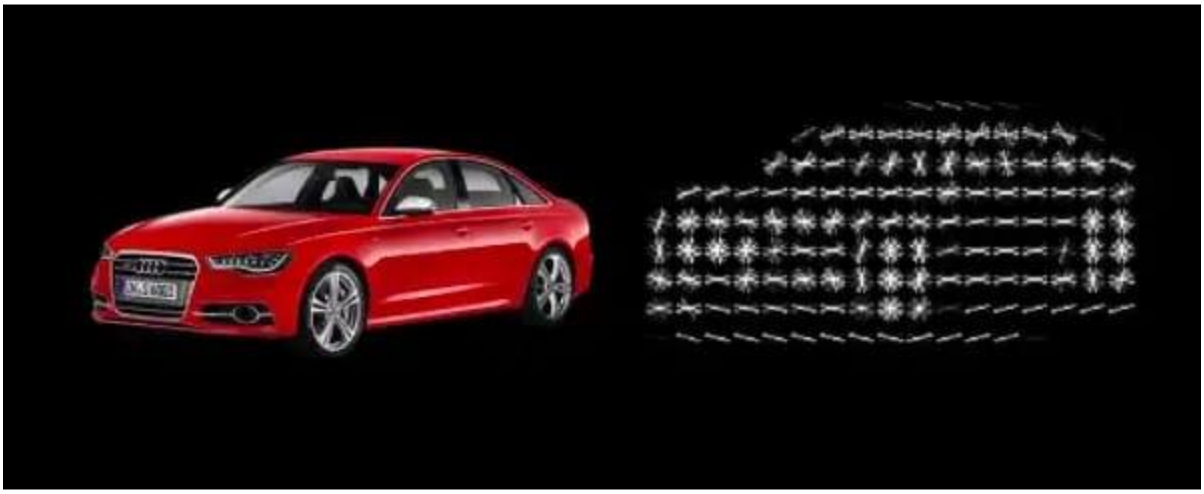


Fig 4.1 Example of HOG feature descriptor

4.2 Calculating Gradients

4.2.1Pre-process the Data (64 x 128)

Pre-processing data is a crucial step. The image needs to be pre-processed to achieve a 1:2 reduction in width to height . Preferably, the image size should be 64 x 128. In several picture sites, patches of various scales are typically examined. The patches being studied have a set aspect ratio, which is the only restriction. In our situation, the patches must have a 1:2 aspect ratio. They may, for instance, be 100200, 128256, or 10002000, but not 101205.

Let's say we chose a patch with a size of 100 by 200 to compute our HOG feature descriptor. This patch was taken from a picture and reduced in size to 64*128.

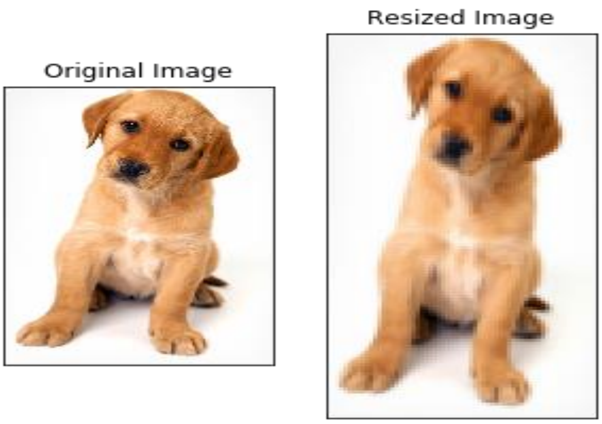


Fig 4.2 Input image of HOG

4.2.2Gradients in x and y directions

The subtle shift in the x and y directions is known as a gradient. We need to first the gradients' horizontal and vertical values. By filtering the image with the following kernels, this is simply accomplished.

1	0	-1

1
0
-1

Let, us select a small portion of the picture to calculate on it:

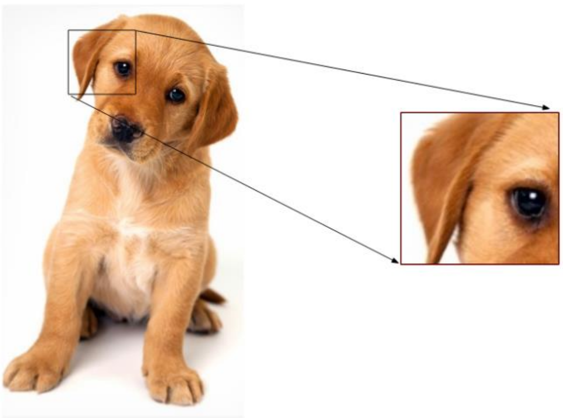


Fig 4.3 Small patch of the image

The pixel values for this patch will be obtained. Let’s take an example of matrix and they are not original pixel values.

121	10	78	96	125
48	152	68	125	111
145	78	85	89	65
154	214	56	200	66
214	87	45	102	45

Fig 4.4 Matrix of the pixel values

Here, the pixel with the value 85 has been highlighted. Now, we must subtract the value on the left from the pixel value on the right to obtain the gradient (or change) in the x-direction. We must subtract the value of the pixel beneath the chosen pixel from the value of the pixel above it, much as how we determine the gradient in the x-axis. For this pixel, the resulting gradients in the x and y directions are as follows:

- X direction change (Gx)=89-78=11
- Y direction change (G_y) =68-56=8

As a result of this approach, we will have two new matrices, one of which will contain gradients in the x and y directions, respectively. Similar to applying a size 1 Sobel Kernel, to this. When a sudden transformation occurs in intensity, like around the edges, the magnitude would be greater.

The slopes in the x and y directions were calculated individually. The same method is repeated again. .

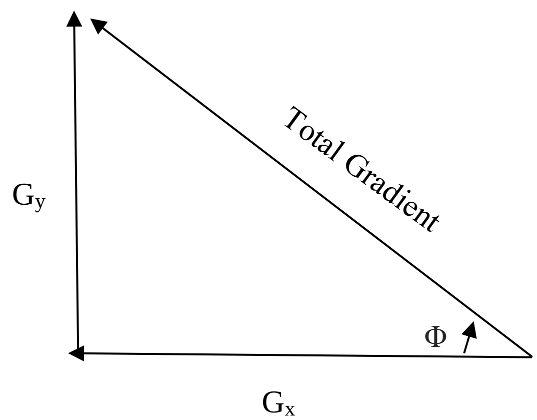


Fig 4.5 Magnitude and Orientation by Pythagoras theorem

The magnitude and orientation would then be determined using these values as the next step. We shall apply Pythagoras' theorem for this step. Here, the base and perpendicular are essentially the gradients. Gx and Gy were assigned the numbers 11 and 8, respectively.

Total Gradient to Magnitude= [(Gx)2 + (Gy)2].

Magnitude of the whole gradient equals [(11)2+(8)2] = 13.6

The direction (or orientation) of the identical pixel is determined in the following step.. We can express the angles' tans by writing:

$Tan(\Phi)= Gy/Gx$

As a result, the angle's value would be:

$\Phi = atan(Gy / Gx)$.

When we enter the data, the orientation comes out to be 36. We now get the overall gradient (magnitude) and orientation for each pixel value. (direction). The histogram must be produced using these gradients and orientations.

4.3 Creation Of Histograms

An illustration of the group of continuous data's frequency distribution is called a histogram. On the x-axis is the variable (represented by bins), and on y-axis is the frequency. Here, we're going to use the frequency on the y-axis and the angle or orientation on the x-axis.

Method 1:

Let's start by creating histograms in the simplest possible manner. The frequency table will be updated when we capture each pixel's value and determine its direction.

121	10	78	96	125
48	152	68	125	111
145	78	85	89	65
154	214	56	200	66
214	87	45	102	45

Frequency						1									
Angle	1	2	3	4 .	35	36	37	38	39 .	175	176	177	178	179	180

Fig 4.6 Creation of histograms

Below is a diagram of how the 85th highlighted pixel was processed. The orientation of this pixel is 36, so we will add a number to the angle value 36 to reflect the frequency:

The same procedure is carried out for each pixel value, and the result is a angle. Angles are listed together with how frequently they appear in the image in a frequency table. This frequency table can be used to make a histogram by placing the frequency on the y-axis and the angle values on the x-axis.

That's one way to create a histogram. Keep in mind that in this case, the histogram's bin value is 1. There are hence about 180 unique buckets, each of which corresponds to an orientation value. Another strategy is to create histogram features with higher bin values..

Method 2:

With the exception of the 20-bin size, this approach is comparable to the preceding one. Therefore, we would receive nine buckets in this situation. Again, for each pixel, we will verify the orientation and keep a 9 x 1 matrix with the frequency of the orientation values. The histogram would result from plotting this:

121	10	78	96	125
48	152	68	125	111
145	78	85	89	65
154	214	56	200	66
214	87	45	102	45

Magnitude		1							
Bin	0	20	40	60	80	100	120	140	160

Method 3:

The gradient value is not taken into account in any of the two methods mentioned above, which solely employ the orientation values to produce histograms. Another method for creating the histogram is to fill the values in the matrix with the gradient magnitude rather than the frequency. Here is an instance:

Magnitude=13.6

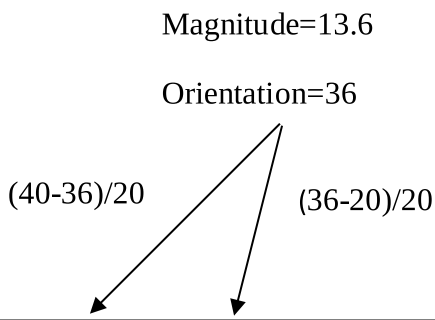
Orientation=36

Magnitude		13.6							
Bin	0	20	40	60	80	100	120	140	160

Here, we are just updating the bin 20 while using the orientation value of 30. We should also give the other bin consideration.

Method 4:

Let's tweak the aforementioned technique just a little bit. The gradient of each pixel will now be added to the bins on either side of the gradient. Keep in mind that the bin value should receive the bigger contribution since it is closer to the orientation.



Magnitude		(4/20)*13.6	(16/20)*13.6						
Bin	0	20	40	60	80	100	120	140	160

Fig 4.7 Histogram is created

This is exactly how histograms are created in the HOG feature descriptor.

4.4 Calculating Histogram of Gradients in 8×8 cells (9×1)

The whole image's histograms are not produced by the HOG feature descriptor. Instead, 8x8 cells are created from the image, each receiving a duplicate of the directed gradients histogram.

For the smaller patches, we acquire the features (or histogram), which subsequently represent the entire image, in this way. Changing this value from 8 x 8 to 16 x 16 or 32 × 32 is probably possible.

We will obtain a 9 x 1 matrix for each cell if we divide the image into 88 cells and create histograms. The fourth method was used to create this matrix.

Fig 4.8 Dividing the image into 8×8 cells

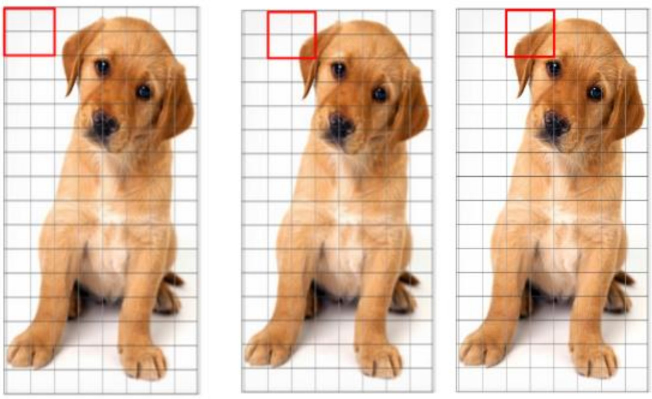
The next step is to normalise the histogram once we have generated the HOG for the 8x8 patches in the image.

4.5 Normalization

Gradients in a 16x16 cell (36x1) are normalised:

The overall illumination has an impact on the image's gradients. Despite the fact that we have previously produced the HOG features for the image's 8x8 cells. As a result, for a given image, a component of the image would be noticeably brighter than the other portions.

This remains visible in the image despite our best efforts. But if we normalise the gradients by taking 16x16 blocks, we may lessen this fluctuation in lighting. Here is an illustration that can demonstrate how 16x16 blocks are made.



In this case, we shall join four 8x8 cells to form a 16x16 block. And we already know that a histogram is made up of a 9x1 matrix for each 8x8 cell. Therefore, we would either have a single 36x1 matrix or four 9x1 matrices. In order to normalise this matrix, each of these values will be divided by the square root of the sum of the squares of the values. In math, for a particular vector V

$V = [a1, a2, a3, \ldots a36]$

We determine the sum of squares' root as follows:

$k = \sqrt{(a1)2+ (a2)2+ (a3)2+ \ldots (a36)2}$

And then divide each value in the vector V by this number, k:

$$\text{Normalised Vector} = \left(\frac{a_1}{k}, \frac{a_2}{k}, \frac{a_3}{k}, \ldots \frac{a_{36}}{k} \right)$$

The resultant would be a normalized vector of size 36×1.

4.6 Final Stage

The process of creating HOG features for the image is now complete. For the 1616 blocks of the image that we have built features for thus far. To obtain the features for the final image, we will now integrate all of these.

For a single 64x128 image, we would first need to determine how many of these 16x16 blocks we would receive.

We would have 105 blocks that are each 7 by 15 squares. These 105 blocks each have a feature vector of 36x1. As a result, there would be 105 x 36 1 = 3780 features in the final image.The creation of HOG features for a single image will now be done.

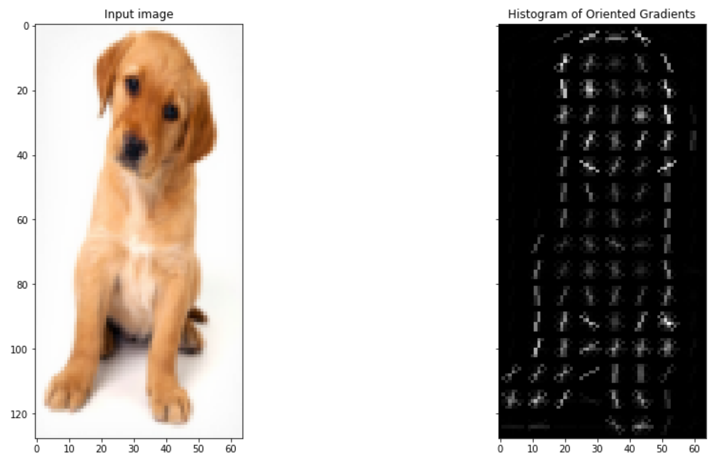


Fig 4.9 Final stage of HOG

4.7 Uses Of HOG

Future developments in image recognition and face detection could be greatly aided by the Histogram of Oriented Gradients object detection approach. This field can have a significant impact on a number of sectors, from enhancing AR tools to enhancing blind people's eyesight. HOGs can be employed in a variety of fields, including driverless cars, augmented reality, and virtual reality (mostly anything involving image detection).

CHAPTER 5: MATLAB

5. MATLAB

5.1 Introduction

We used MATLAB to implement this project. The Calculation that the Works company created the multi-paradigm programming language and environment known as MATLAB (short for "matrix laboratory"). You can use MATLAB to control matrices, visualise data and operations, construct connect to other user interfaces computer languages. Although symbolic computation is not a core capability of MATLAB, it is provided via an optional toolbox that employs the Mu PAD symbolic engine. Simulink is an additional programme that includes Multi-domain simulation in graphics and model-based design for embedded and dynamic systems.

By 2020, MATLAB will have 4 million plus users. Users of MATLAB come from a wide range of engineering, scientific, and economic backgrounds.

A high-performance language for technical computing is called MATLAB. It has an intuitive interface and blends computation, visualization, and scripting. and blends computation, visualization, and scripting. While describing concerns and solutions using well-known mathematical symbols.

Typical uses include:

- Computational mathematics
- Algorithm creation
- Prototyping, simulation, and modelling
- Data exploration, analysis, and visualisation
- Engineering and scientific graphics
- Developing applications, including GUI design

A basic a component of the interactive system's data. The array MATLAB is one that does not need to be dimensioned. This enables you to tackle a variety of technical computer problems in a fraction of the time by writing programmes in scalar noninteractive languages like C or Fortran, as well as matrix and vector formulations.

Matrix Laboratory is an abbreviation for Matrix Laboratory. The original purpose of MATLAB was to make the cutting-edge matrix software developed by the LINPACK and

EISPACK projects, which represent the cutting-edge in matrix calculating software, simple to use. As a result of user feedback, MATLAB has grown over many years. It is a popular teaching tool in universities for beginning and advanced maths, engineering, and science courses. The industry standard for extremely productive research, development, and analysis is MATLAB.

Toolboxes are MATLAB applications that provide application-specific solutions. Toolboxes are essential for the great majority of MATLAB users since they enable you to discover and use specific technology. Toolboxes are large collections of MATLAB functions (M-files) that allow the MATLAB environment to be extended to handle specific challenges. Signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many other fields are covered by toolboxes.

5.2 MATLAB Software

The MATLAB system is divided into five major components:

The MATLAB language

Object-oriented programming, control flow statements, functions, data structures, input/output, and input/output are all characteristics of this high-level matrix/array language. Both "programming in the large" and "programming in the small" can fully mature thanks to it. sophisticated application programmes as well as quickly produce subpar throw-away apps.

The MATLAB working environment

This collection of assets and equipment is used by MATLAB users and programmers. It includes tools for changing variables in your workspace as well as importing and exporting data. The offered tools enable the creation, management, debugging, and profiling of MATLAB programs, known as M-files.

Graphics Management

This image depicts the MATLAB graphics system. It includes instructions for image processing, animation, two- and three-dimensional data visualization, and presentation graphics at the highest level. Additionally, it contains low-level instructions that let you create entire Graphical User Interfaces for your MATLAB applications as well as completely customise the appearance of graphics.

The MATLAB mathematical function library

There are many different types of computer algorithms included in this, from basic ones like sum, sine, and cosine to more complex ones like inverse, eigenvalues, Bessel functions, and rapid Fourier transformations.

The Application Program Interface (API) of MATLAB

This is a library for writing C and Fortran programmes that interface with MATLAB. It offers support for invoking MATLAB routines (dynamic linking), using MATLAB as a computational engine, and reading and writing MAT-files.

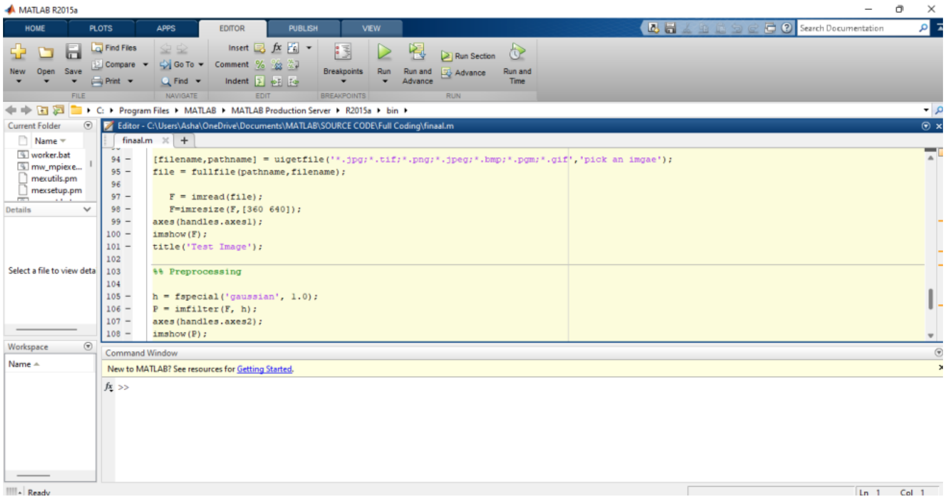


Fig 5.1 MATLAB Application Program

5.3 MATLAB's Power of Computational Mathematics

MATLAB is used in every branch of computational mathematics. The following are some common mathematical operations that make use of it.

- Working with Arrays and Matrices
- Graphics and charting in 2-D and 3-D

Calculus and Differential Equations, Linear Algebra, Algebraic Equations, Non-Linear Functions, Statistics, Data Analysis, Numerical Calculations, Integration, Transforms, Curve Fitting, and a number of more specialised functions

5.4 Features of MATLAB

The basic characteristics of MATLAB are as follows:

• It is both a high-level programming language and a dynamic environment for problem solving, design, and iterative exploration.

• It includes a large a collection of formulas for solving linear algebra, statistics, Fourier analysis, filtering, numerical integration, and ordinary differential equations.

• It includes tools for creating personalised plots as well as built-in graphics for data visualisation.

• The MATLAB programming interface provides developers with tools to improve code efficiency and maintainability.

• It includes tools for incorporating MATLAB-based algorithms into third-party software and programming languages like C, Java,.NET, and Microsoft Excel.

5.5 Uses Of MATLAB

MATLAB is a widely used computer tool in science and engineering, with applications in physics, chemistry, mathematics, and a variety of engineering courses. It is utilised in a wide range of applications, including signal processing and communication.

- Test and Measurement
- Control Systems
- Image and Video Processing
- Computational Biology
- Computational Finance

CHAPTER 6: METHODOLOGY

6. METHODOLOGY

6.1 Introduction

The primary goal of our study is to prevent animal-vehicle collisions. Our suggested system is image-processing-based, with all preprocessing work performed on the camera interfaced with the video format input. The film was shot with a optical sensor camera that is pointing ahead and features a moving animal as well as various stationary and moving things. This movie was saved on the computer and then manipulated into various frames. Following the acquisition of the image, we execute pre-processing processes to improve the image. Image enhancement is performed during the preprocessing stage. We use the guassian filter to improve the image. The next stage after preprocessing is object detection. We employ the feature extraction approach after detecting the object to convert the gathered images to grayscale and perform edge detection. The generated findings are put into the animal detection algorithm, which evaluates the collected photographs for the presence of animals. When an animal is spotted by the camera, the driver is alerted, which helps to avoid road accidents. so that the brakes are properly used to prevent a collision between the animal and the car, and then notify the driver so that the brakes can be used. This reduces the likelihood of animal-vehicle collisions and accidents.

6.2 Flow Chart

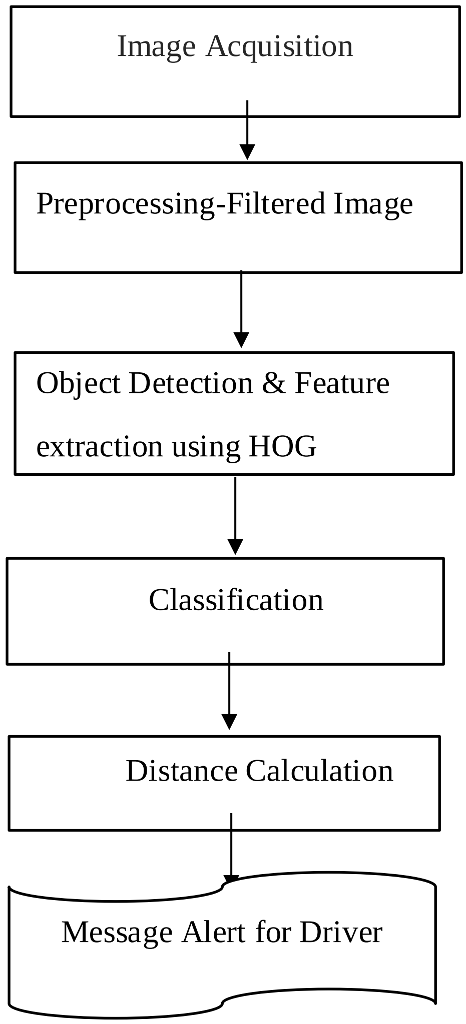


Fig 6.1 Flow chart

6.3 Image Acquisition

In image processing, getting a picture from a source, often one that uses hardware, so that it can be processed anyway you like is the process of image acquisition. Image acquisition is always the first stage in the workflow sequence in image processing because processing is impossible without a picture. The output of whichever procedure was used to generate it, the resulting image is entirely unprocessed and can be highly useful in some sectors because it offers a constant starting point from which to work. One of the ultimate goals of this method is to have an input source that is so meticulously controlled and measured that the same image can be replayed under identical circumstances, making it simpler to spot and remove undesirable characteristics.

Real-time picture acquisition is a type of image acquisition used in image processing. This usually entails collecting photos from a automated source for taking pictures. With real-time image collection, a stream of data is produced that can be automatically analyzed, put on hold for later processing, or combined into a single media format. A common technology used with real-time image processing is background image acquisition, which refers to both software and hardware that can quickly save images streaming into a system.

There are various advanced image acquisition methods in image processing that make use of customised hardware. One of these ways is the acquisition of three-dimensional (3D) images. This may necessitate the use of two or more cameras that have been properly aligned at positions around a target, resulting in a sequence of images that can be aligned to generate a 3D or stereoscopic scene, or to measure distances. Some satellites employ 3D picture acquisition techniques to create accurate replicas of various surfaces.

6.4 Preprocessing

To enable more effective analysis, pre-processing aims to raise the image's quality. Preprocessing allows us to decrease undesirable distortions and enhance some qualities needed for the application we are developing. Depending on the use, these qualities may change.

Activities involving images at the most fundamental level of abstraction, where input and output are intensity images, are referred to as pre-processing. An intensity image is frequently represented by a matrix of image function values, and these iconic images are identical to the original sensor data. In accordance with the size of the pixel neighbourhood used to calculate a new pixel brightness, four different types of image pre-processing techniques are distinguished..

Image augmentation is done during preprocessing. This is the process of modifying digital photographs to make them appropriate for display or additional image analysis. You can, for example, eliminate noise, sharpen, or brighten an image to make it easier to identify crucial characteristics.

Applications of image enhancement

Here are some picture enhancing examples:

- Deblur photos
- Adjust contrast and brightness
- Sharpen and smooth

Image enhancement techniques,

1.Apply Gaussian Filter

Image smoothing is a technique used in digital image processing to decrease and suppress visual noise. Smoothing filters that are commonly used include average smoothing, Gaussian smoothing, and adaptive smoothing Image.

Apply Gaussian Filter

In this phase, we smooth the image by applying a Gaussian filter to the test image. Image noise is introduced at the point of capture from cameras, printing, or transmission. Noises in image processing can be determined by a difference in intensity between neighbouring pixels. There are several forms of noises. Gaussian noise, for example, modifies each pixel by a modest amount, whereas salt-and-pepper noise randomly scatters white or black pixels across the image. Noise removal techniques smooth the image using linear or non-linear filters to lessen the visibility of sounds.

A Gaussian blur, also known as Gaussian smoothing in image processing, is the result of concealing a picture with a Gaussian function. It is a commonly used effect in illustration programming, frequently to reduce image noise and detail. The primary goal of smoothing is to reduce the influence of camera noise, false pixel values, missing values, and so on. This is sometimes referred to as neighbourhood averaging. Each pixel in the smoothed image is derived from the average pixel value in the input image's neighbourhood.

6.5 object detection

Finding instances of objects in photographs is a process known as object detection. Using deep learning, object detection is a subset of object recognition, which involves not only identifying the object in the image but also pinpointing its location. This makes it possible to recognise and locate several items inside the same image.

Object recognition is a computer vision technology that aids in the identification of objects in images and movies. Object recognition is a significant outcome of deep learning and machine learning algorithms. When examining images or movies, humans can swiftly identify people, objects, scenes, and other visual qualities. The goal is to teach a computer to execute behaviours that people do naturally so that it may understand some parts of what an image conveys.

While object detection and object recognition are both methods for recognising objects, they differ in how they are carried out. The distinctions between object localisation and object detection, for instance, might be difficult to understand, even though all three tasks may be collectively referred to as object recognition. Image categorization, by contrast, is straightforward.

Identifying objects and recognising them are crucial computer vision tasks. Object identification pinpoints an object's presence, size, and placements inside an image. The object class that the object corresponds to in the training database is identified via object recognition. Typically, object detection comes before object recognition.

It can be considered a two-class object recognition since one class represents the object class and another class represents the non-object class. Other types of object detection include soft detection, which only detects the existence of an object, and hard detection, which detects both the presence and the location of the object. Typically, the field of object detection locates areas of an image whose photometric or geometric

6.6 FEATURE EXTRACTION

- After preprocessing stage, animal detection is performed. It is done by feature extraction and classification.
- Before the feature extraction, object detection is implemented. Saliency map detection method is proposed for object detection. After that feature extraction is explored for detected object. HOG feature descriptor proposed for feature extraction.
- A feature descriptor similar to the Canny Edge Detector, SIFT (Scale Invariant and Feature Transform), is the histogram of oriented gradients, or HOG. It serves the aim of object detection in computer vision and image processing. The HOG descriptor emphasises the composition or form of an object.
- Due to certain significant advantages over other descriptors, the HOG descriptor is mostly appropriate for animal detection in video or photos. It is invariant to geometric and photometric alterations since it operates on local cells.
- Secondly, assuming animals maintain a nearly upright posture, varied body movements can be ignored by coarse (spatial) sampling, fine orientation sampling, and strong local photometric normalisation. Following that, extracted characteristics are used as classification input. On the basis of the extracted feature, a was discovered in that animal.

6.6.1 Steps in HOG

- **Preprocess the data**:

  Data preprocessing is an essential step. The image needs to be preprocessed to reduce the width to height ratio to 1:2. Preferably, the image size should be 64 x 128.
- **Gradients calculation:**

  The subtle shift in the x and y directions is known as a gradient. First, we must compute the gradients along the horizontal and vertical axes. By filtering the image with the following kernels, this is simply accomplished.

1	0	-1

1
0
-1

We will receive two new matrices as a result of this procedure, one of which will store gradients in the x-direction and the other in the y-direction. This is comparable to applying a size 1 Sobel Kernel. When there is a sudden change in intensity, like around the edges, the magnitude would be greater.

Applying the Pythagorean theorem, let's get the general gradient size:

$$\text{Total Gradient Magnitude} = \sqrt{[(G_x)^2+(G_y)^2]}$$

Next, to determine the direction (or orientation) for a single pixel. The angles' tan can be written as follows:

$$\tan(\Phi) = Gy / Gx$$

Hence, the value of the angle would be:

$$\Phi = \operatorname{atan}(Gy / Gx)$$

We need to generate the histogram using these gradients and orientations.

6.6.2 Creation of histograms

An illustration of the frequency distribution of a group of continuous data is called a histogram. On the x-axis is the variable (represented by bins), and on the y-axis is the frequency.

Histogram of Gradients in 8×8 cells (9×1)

Not all of the image's histograms are generated by the HOG feature descriptor. Instead, an 8x8 grid is used to divide the image into cells, and each cell's oriented gradient histogram is generated. We will obtain a 9 x 1 matrix for each cell if we divide the image into 8x8 cells and create histograms.

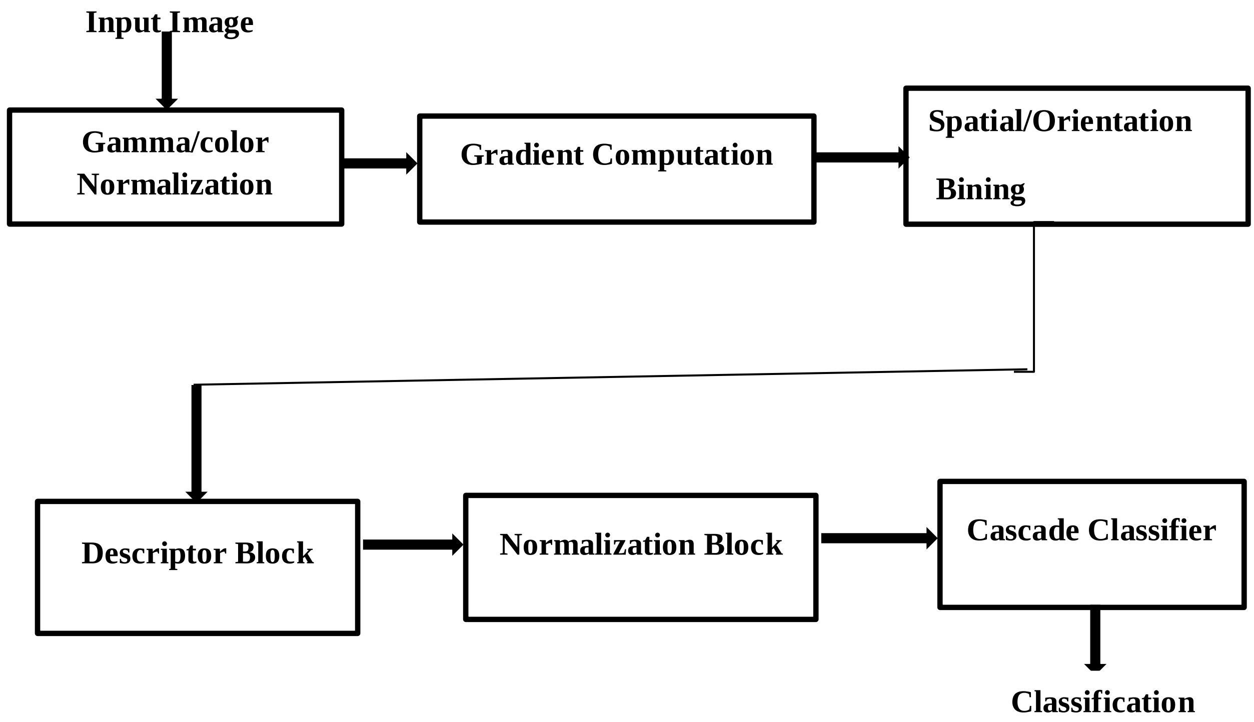


Fig 6.2 Block Diagram of Normalization

6.6.3 Normalization

Gradients in a 16x16 cell (36x1) are normalised:

The gradients of the image are responsive to the overall lighting, and the HOG features are constructed for the 8x8 cells of the image. As a result, for a given image, a component of the image would be noticeably brighter than the other portions. This remains visible in the image despite our best efforts. But if we normalise the gradients by taking 16x16 blocks, we may lessen this fluctuation in lighting.

Each of these values will be divided by the square root of the sum of the squares of the values in order to normalise this matrix. In mathematics, for a specific vector V:

$$V = [a1, a2, a3, \ldots a36]$$

We determine the sum of squares' root.

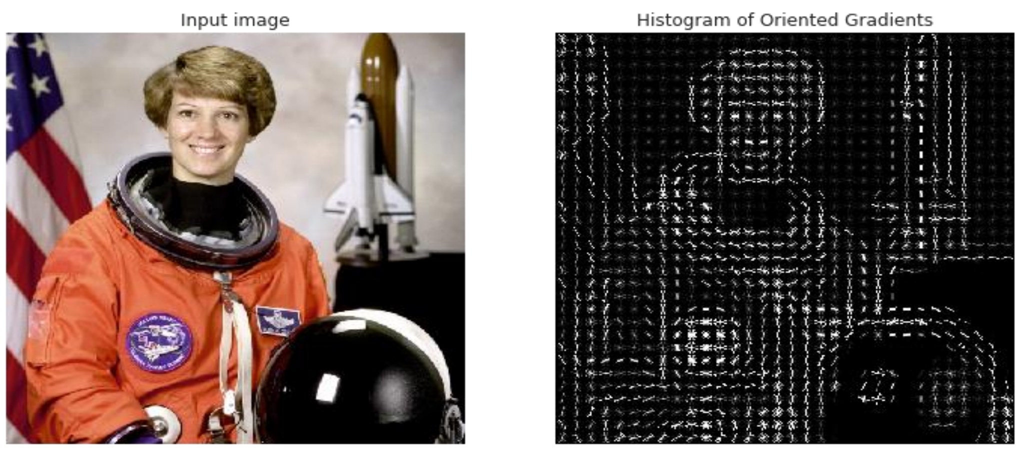
$$k = \sqrt{(a1)2+ (a2)2+ (a3)2+ \ldots (a36)2}$$

And then divide each value in the vector V by this number, k:

$$\text{Normalised Vector} = \left(\frac{a_1}{k}, \frac{a_2}{k}, \frac{a_3}{k}, \ldots \frac{a_{36}}{k} \right)$$

6.6.4 Final Stage

For each of the image's 16x16 blocks, we have developed features. To obtain the features for the final image, we will now integrate all of these. We would need to first determine how many of these 16-by-16 blocks we would receive for a single 64-by-128 image.



6.7 DISTANCE CALCULATION

The next step is to calculate distance. The steps to determine how far the identified animal is from the vehicle with the camera are as follows:

- The size of the image is 640 by 480.
- The X scale is 0–640.
- The Y scale is 0 to 480.

Let (x, y) represent the cow's right bottom coordinate. The cow is therefore 480 - y away from the lower edge (of the car or camera).

TRANSFORMATION OF PIXELS TO METER

Once the object (animal) is spotted in the picture, Between the object's depth in pixels and its distance from the camera in meters, there is some link attached with vehicle. The depth in pixels increases as the object's depth in metres from the camera-mounted vehicle grows (and the object's size reduces), and vice versa. This gave us a hint that there might be a correlation between the object's depth in pixels and metres. We collected multiple photographs of the same object held at varied depths from the camera centre after determining the camera position in the automobile and height above the ground (camera calibration). We were aware of the object's depth from the middle of the camera in metres. We then recorded the object's appropriate depth in pixels. The second order polynomial equation that fits data the best is

Y is the depth in pixels, while x is the depth in meters, as in $y = 0.0323x\ 2 + 22.208x + 1.3132$.

Fig 6.3 Distance calculated

6.8 MESSAGE ALERT

The next stage is to determine how far away. The creature came from the test vehicle. once it has been identified in the video. In the event that an animal is found, the separation between the testing vehicle and the animal on the road is computed using the pixels from the image that was collected. Three different types of information are sent to the driver depending on the distance of the camera-mounted vehicle. Animals are considered to be close to a car if they are right up next to it, slightly far from a vehicle if they are a little bit away from it, and very far from a vehicle with a camera if they are a great distance away. When the driver approaches a specific distance on the road, this automatically alerts them. This will warn the motorist and stop traffic collisions.so that the driver is made aware so that the brakes can be deployed appropriately to prevent a collision between the animal and the car. Accidents and collisions involving animals are avoided as a result.

CHAPTER 7: RESULTS AND DISCUSSION

7. RESULTS AND DISCUSSION

In this project, we have implemented procedure in this way. There are several images from the data base. Pre-processing activities are performed after the image is acquired to improve the image. Image enhancement occurs during the preprocessing phase. Object detection comes next after preprocessing. The collected images are converted into grey scale and edge detection is performed after the item has been discovered using the feature extraction approach. The gathered results are put through the algorithm for animal detection to see if any animals show up in the photographs that were taken. Finding the separation between the animal and the test vehicle is the next step once the animal has been identified in the video. The separation between any discovered animals and the testing vehicle on the road is estimated using the pixels from the image that was collected. based on the distance to the car with the camera. The device automatically warns the motorist when they get too close to the road. This will warn the motorist and stop traffic collisions.to enable the use of the brakes to warn the motorist in order for him to use the brakes in time to prevent a collision between the animal and the car. Animal-vehicle collisions and accidents are avoided as a result.

Fig 7.1 Input Image

Fig 7.2 Filtered Image

Fig 7.3 Object Detection Map

Fig 7.4 Object Detection Image

Fig 7.5 Object Cropped Image

Fig 7.6 Gray Image

\

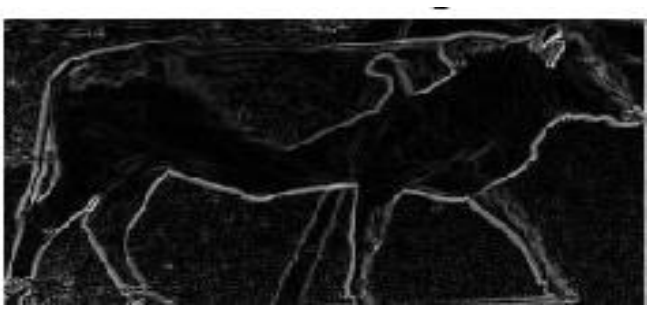


Fig 7.7 Gradient Image

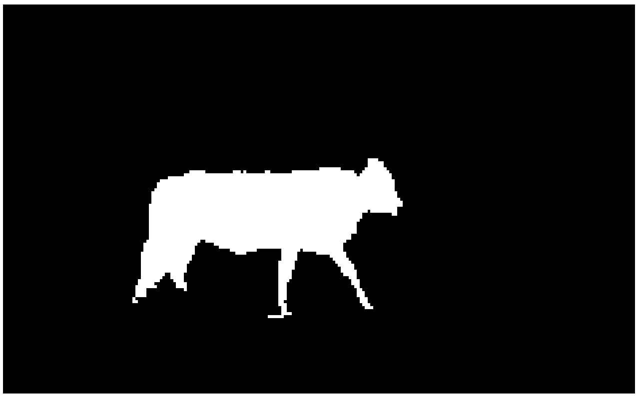


Fig 7.8 Detected Image

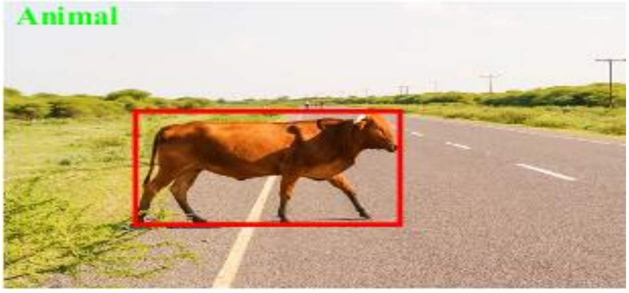


Fig 7.9 Input Image

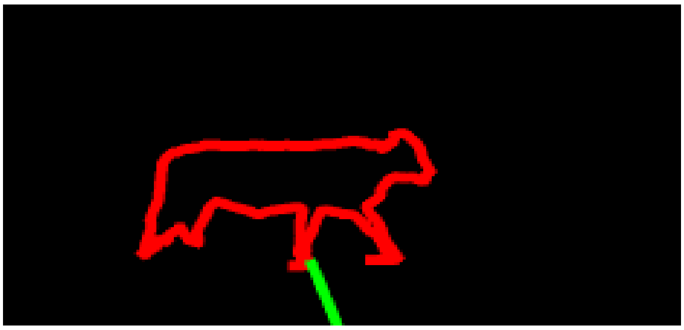


Fig 7.10 Output Image

CHAPTER 8 CONCLUSION

8. CONCLUSION

A reliable automatic animal detecting system and a warning systems can assist drivers in lowering the frequency of animal-vehicle incidents on roads and highways. Here, we covered the need for automatic animal detection systems as well as our Cascade and HOG classification animal detection algorithm. The programme is capable of finding an animal in a variety of settings. On Matlab, we implemented the suggested task.. After performing the first picture acquisition, a pre-processing approach is used to create a smooth image. We use the hog method and do classification on the generated image. We must now determine the distance before sending the driver a warning. By doing this, we can reduce the automobile and animal collisions.

REFERENCES

- Mai Ibraheam, Kin Fun Li, Fayez Gebali, "An Accurate and Fast Animal Species Detection System for Embedded Devices", IEEE Access, vol.11, pp.23462-23473, 2023.

- Mahima R, Meenu M, Manjari K, Rovenal S, Keerthana Sri S, Sruthi M. P, "Highway Collision Avoidance by Detection of Animal's Images", 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), pp.307-311, 2023.

- Naveenkumar M, Manoj R, Nandhakumar B, Rahul R, "DenseNet201 for Animal detection and repellent system", 2022 International Conference on Electronic Systems and Intelligent Computing (ICESIC), pp.213-218, 2022.

- Rashmi Gandhi, Aakankshi Gupta, Ashok Kumar Yadav, Sonia Rathee, "A Novel Approach of Object Detection using Deep Learning for Animal Safety", 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp.573-577, 2022.

- Y A Roopashree, M Bhoomika, R Priyanka, K Nisarga, Sagarika Behera, "Monitoring the Movements of Wild Animals and Alert System using Deep Learning Algorithm", 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), pp.626-630, 2021.

- Teddy Surya Gunawan, Arifah Omar, Mira Kartiwi, Mufid Ridlo Effendi, Nanang Ismail, "Prototype Development of Vehicle Collision Avoidance System using Arduino", 2021 IEEE 7th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), pp.70-75, 2021.

- S. Sharma and D. Shah, ‘‘Real-Time automatic obstacle detection and alert system for driver assistance on Indian roads,’’ Indonesian J. Elect. Eng. Comput. Sci., vol. 1, no. 3, pp. 635–646, Mar. 2016.

- J. Padmanaban, R. Rajaraman, G. Stadter, S. Narayan and B. Ramesh, "Analysis of in-depth crash data on indian national highways and impact of road design on crashes and injury severity", Proc. 4th Int. Conf. ESAR 'Expert Symp. Accident Res.', pp. 170-183, Sep. 2010.

- J. Padmanaban, R. Rajaraman, G. Stadter, S. Narayan, and B. Ramesh, ''Analysis of in-depth crash data on indian national highways and impact of road design on crashes and injury severity,'' in Proc. 4th Int. Conf. ESAR 'Expert Symp. Accident Res.', Hanover, Germany, Sep. 2010, pp. 170–183.

- S. Sharma and D. J. Shah, ''A brief overview on different animal detection method,'' Signal Image Process., Int. J., vol. 4, no. 3, pp. 77–81, Jun. 2013.

- F. A. Wichmann, J. Drewes, P. Rosas, and K. R. Gegenfurtner, ''Animal detection in natural scenes: Critical features revisited,'' J. Vis., vol. 10, no. 6, pp. 1–27, 2010.

- S. Shaikh, M. Jadhav, N. Nehe, and U. Verma, ''Automatic animal detection and warning system,'' Int. J. Adv. Found. Res. Comput., vol. 2, pp. 405–410, Jan. 2015.

- S. Sharma and D. Shah, ''Real-Time automatic obstacle detection and alert system for driver assistance on Indian roads,'' Indonesian J. Elect. Eng. Comput. Sci., vol. 1, no. 3, pp. 635–646, Mar. 2016.

- N. Dalal and B. Triggs, ''Histograms of oriented gradients for human detection,'' in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), New York, NY, USA, Jun. 2005, pp. 63–69.

- N. Gupta, A. Rawal, V. L. Narasimhan, and S. Shivani, ''Accuracy, sensitivity and specificity measurement of various classification techniques on healthcare data,'' IOSR J. Comput. Eng., vol. 11, no. 5, pp. 70–73, May/Jun. 2013.