A

PROJECT REPORT

ON

“Animal Detection to Avoid Collision Using Machine Learning (YOLO algorithm)”

SUBMITTED TO

SHIVAJIUNIVERSITY, KOLHAPUR

IN THE PARTIAL FULFILLMENT OF REQUIREMENT FOR THE AWARD OF DEGREE BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

MISS. PRANITA R. GIRI	15CMPN08
MISS. SHRADDHA R. KARADGE	15CMPN20
MISS. PRIYANKA A. KHOT	15CMPN24
MISS. SHRADDHA S. RHATOLE	15CMPN47
MISS. JYOTI A. SHINDE	15CMPN51

UNDER THE GUIDANCE OF

PROF. Y. M. KAMBLE

DKTE

Promoting Excellence in
Teaching, Learning & Research

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DKTE SOCIETY’S TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI

(AN AUTONOMOUS INSTITUTE)

2018-2019

D.K.T.E.SOCIETY’S

TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI

(AN AUTONOUMOUS INSTITUTE)

(A+ Grade Accreditation by NAAC)

(ISO 9001:2015CERTIFIED)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DKTE
Promoting Excellence in
Teaching, Learning & Research

CERTIFICATE

This is to certify that, project work entitled

“Animal Detection to Avoid Collision Using Machine Learning (YOLO algorithm)”

is a bonafide record of project work carried out in this college by

MISS. PRANITA R. GIRI	**15CMPN08**
MISS. SHRADDHA R. KARADGE	**15CMPN20**
MISS. PRIYANKA A. KHOT	**15CMPN24**
MISS. SHRADDHA S. RHATOLE	**15CMPN47**
MISS. JYOTI A. SHINDE	**15CMPN51**

is in the partial fulfillment of award of degree Bachelor in Engineering in Computer Science & Engineering prescribed by Shivaji University, Kolhapur for the academic year 2018-2019.

Prof. Y. M. KAMBLE **EXAMINER:**

(PROJECT GUIDE)

Prof.(Dr.) D.V.KODAVADE
(HOD CSE DEPT.)

Prof.(Dr.) P.V.KADOLE
(DIRECTOR)

DECLARATION

We hereby declare that, the project work report entitled "Animal Detection To Avoid Collision Using Machine Learning (YOLO algorithm)" which is being submitted to D.K.T.E. Society's Textile and Engineering Institute Ichalkaranji, affiliated to Shivaji University, Kolhapur is in partial fulfillment of degree B.E. (CSE). It is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for the award of any degree. Further, we declare that we have not violated any of the provisions under Copyright and Piracy / Cyber / IPR Act amended from time to time.

STUDENT NAME	ISO ROLL NO	SIGNATURE
MISS. PRANITA R. GIRI	15CMPN08	
MISS. SHRADDHA R. KARADGE	15CMPN20	
MISS. PRIYANKA A. KHOT	15CMPN24	
MISS. SHRADDHA S. RHATOLE	15CMPN47	
MISS. JYOTI A. SHINDE	15CMPN51	

ACKNOWLEDGEMENT

With great pleasure we wish to express our deep sense of gratitude to Prof. Y. M. Kamble for his valuable guidance, support and encouragement in completion of this project report.

Also, we would like to take opportunity to thank our head of department Dr. D. V. Kodavade for his co-operation in preparing this project report.

We feel gratified to record our cordial thanks to other staff members of Computer Science and Engineering Department for their support, help and assistance which they extended as and when required.

Thank you,

MISS. PRANITA R. GIRI	**15CMPN08**
MISS. SHRADDHA R. KARADGE	**15CMPN20**
MISS. PRIYANKA A. KHOT	**15CMPN24**
MISS. SHRADDHA S. RHATOLE	**15CMPN47**
MISS. JYOTI A. SHINDE	**15CMPN51**

ABSTRACT

Today’s automobile design primarily depends on safety measures, security tools and comfort mechanism. The safety of an automobile is the highest priority. A major percentage of these road crashes and accidents involved cars, other vehicles and animal collision. So to reduce accidents we are developing the system which detects the animal in front of the vehicle by the camera mounted on the vehicle. In this project we are implement YOLO (You Only Look Once) algorithm to detect animals. YOLO is specifically designed for the practical (real) time object detection. And the main aim is to detect the animal and alert to the driver by the alarm.

INDEX

1. INTRODUCTION

a. Problem Definition

A practical animal detection using machine learning (YOLO algorithm). YOLO is the advance technique used in real time object detection. We can finally tackle the objective resulting the detection of the object.

b. Aim and objective of the project

An application for real time object detection and tracking on image or video streams from the mounted cameras on the vehicle is discussed in this paper. In many situation, it is useful to detect where the safety is important for e.g. face detection at door, at military to count the no. of peoples. Prototype is to detect object in video stream or image based on the machine learning and OpenCv. The system is based on YOLO algorithm and extract features using CNN (Convolutional Neural Network) module. The main purpose of the system is the safety of animal as well as person. A program is developed using python OpenCv and detect the animal with the help of YOLO algorithm.

c. Scope and limitation of the project

Today's automobile design primarily depends on safety measures, security tools and comfort mechanism. The system will detect types of animals. This system will be helpful for human security in forests also. So system can aware the driver from animal-vehicle collision.
This proposed system cannot alert the driver when the vehicle speed is above 35 kmph. No effort has been made to detect animals during the night, which is expected to be done in our scope of study.

d. Timeline of the project

We have implemented classic life cycle approach as called "Waterfall Model". To reach to our final objective the approach which is sequential begins at system level and progress through analysis, design, coding, testing and maintenance. We had completed software requirement analysis by the start of September 2018 which encompasses both system and software requirement gathering. By the end of December we had completed project planning and designing. On the basis of design previous stage by the mid of February 2019 we completed coding stage. After completion of coding stage testing stage carried out in the mid of March 2019. Various criteria of testing were taken into account like, we test the system for static input images and videos with different extensions (.jpg, .jpeg, .mp4,.avi). It is same as implementing first unit testing of each module, followed by integration testing in which integration of all module and finally validation testing to go with the design phase.

e. Project Management Plan

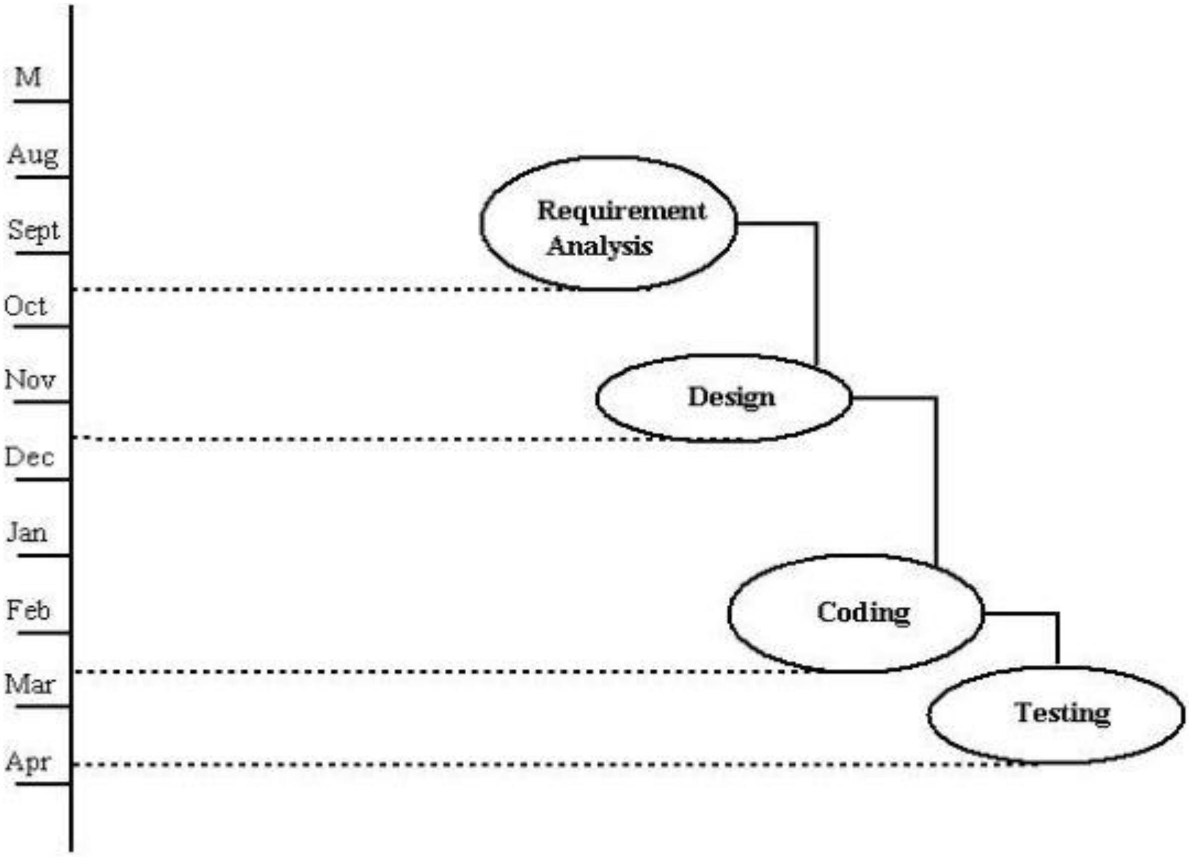


Fig. 1.5 Project Management Plan

f. Project Cost

In this project the cost estimation based on COCOMO (Constructive Cost Model), the formula for this model is as follows,

Effort = Constant × (Size) Scale Factor × Effort Multiplier

(Effort in terms of the person months)

Constant = 2.45 based on 1998 Organic Mode,

Size = Estimated size in KLOC,

Scale Factor = Combined Process Factors,

Effort Multiplier = Combined Effort Factors.

The basic COCOMO equations take the form,

Effort Applied (E) = ab (KLOC)bb[man-months]

Development Time (D) = cb(Effort Applied)db [months]

People Required (P) = Effort Applied/ Development Time [count]

Where KLOC is estimated number of delivered lines (expressed in thousands) of code of our project. The coefficient ab, bb, cb and db are given in the following table.

Software Project	**Ab**	**Bb**	**cb**	**db**
Organic	2.4	1.05	2.5	0.38
Semidetached	3.0	1.12	2.5	0.35
Embedded	3.6	1.05	2.5	0.38

Embedded Mode

EFFORT = 2.4× (0.7)^1.05= 1.65

Development Time = 2.5 1.65^0.38 = 3.02

People Required = 1.65/3.02 = 0.54 ~ 5 People.

2. Background study and Literature overview

a. Literature Overview

[1]Applications built on detection of animals play a vital role in providing solutions to various real-life problems. The base for most of the applications is the detection of animals in the video or image. A recent study shown that human beings have to take the final call while driving whether they can control their car to prevent collision with a response time of 150ms or no. The issue with the above approach is that human eyes get exhausted quickly and need rest, which is why this method is not that effective.

[2]Some scientific researchers have proposed a method that requires the animals to take a pose towards the camera for the trigger, including face detection. The problem with this technique is that face detection requires animals to see into the camera which is, not necessarily captured by the road travel video. Animals can arrive from a scene from various directions and in different sizes, poses, and color.

[3]As one of the most successful applications of image analysis and understanding, animal recognition has recently received significant attention, especially during the past several years. At least two reasons account for this trend: the first is wide range for commercial and law enforcement applications, and the second is the availability of feasible technologies after 30 years of research. In addition, relevant topics such as psychophysical studies, system evaluation are covered.

b. Investigation of current project and related work

The current technique used is YOLO with CNN. Compared to other region proposal classification networks (fast RCNN) which perform detection on multiple times for various regions in an image, YOLO architecture is more like FCNN (fully convolutional neural network) and passes the image (n×n) once through the FCNN and output is (m×m) prediction. This architecture is splitting the input image in mxm grid and for each grid generation 2 bounding boxes and class probabilities for those bounding boxes with the keep in mind that bounding box is more likely to be larger than the grid itself.

3. Requirement analysis

Open-CV (Open source Computer Vision) is a library of Programming functions mainly aimed at real time computer vision. The library has more than more than 2500 optimized algorithms which include a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identifying objects, classifying human actions in videos, track camera movements, track moving objects, extract 3D models of object, produce 3D points cloud from stereo cameras, stitch image together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality etc.

Our system will take the input as any real time image or video captured from camera saved with extension .jpeg for image and .mp4 or .avi for videos. The output after detection is stored in folder in form of image or video resp.

a. External Interfaces:

- The system will get a video.
Input: Video
Output: Verification

- The system will convert this video into frames.
Input: Valid video
Output: Frames

- The system will perform a preprocessing operation on frames.
Input: Frames
Output: Classification of frames

- The system will detect an animal.
Input: Classification of frames
Output: Detect animal and alert driver

b. Functional Requirement:

First of all, system will convert video into frames in short interval of time. Then apply preprocessing operation with the frames to enhance the image. Preprocessing will cover all the aspects of frames as sharpening, clearing image.

After getting frames apply YOLO algorithm which is generally used for real time object detection purpose. In this one bounding box is responsible for detecting any object in given image or frame. Generally there are two parts in YOLO i.e. recognition, localization. In this once the object is detected, distance will be calculated. Finally driver will get alerted.

4. System design

4.1 System Architecture:

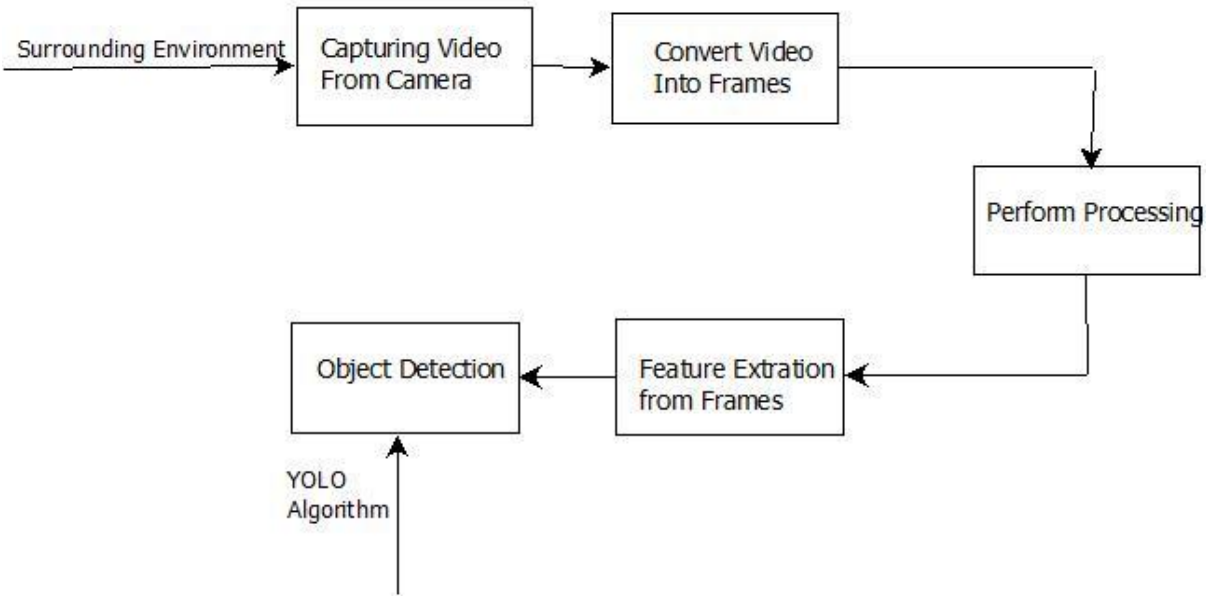

Surrounding Environment
Capturing Video From Camera
Convert Video Into Frames
Perform Processing
Feature Extration from Frames
Object Detection
YOLO Algorithm

Fig. 4.1 System Architecture Diagram

4.2 Use Case Diagram:

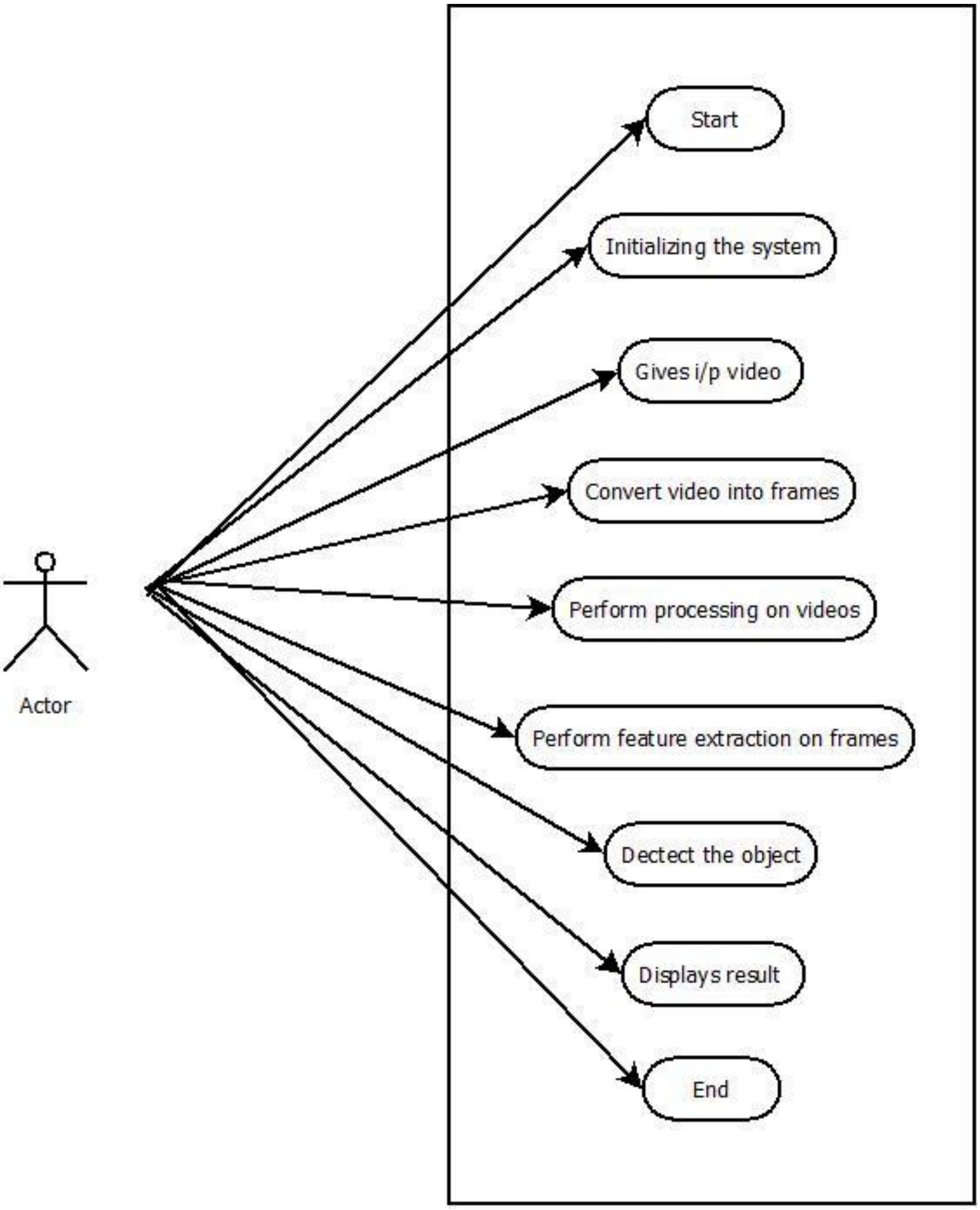

Start
Initializing the system
Gives i/p video
Convert video into frames
Perform processing on videos
Perform feature extraction on frames
Dectect the object
Displays result
End
Actor

Fig. 4.2 Use Case diagram

4.3 Algorithm description of each module:

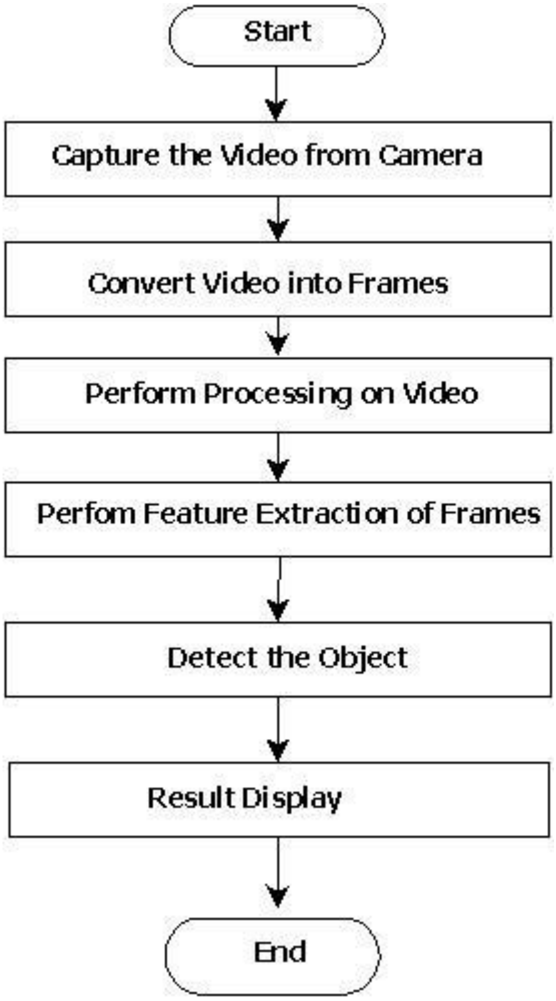

Start
Capture the Video from Camera
Convert Video into Frames
Perform Processing on Video
Perfom Feature Extraction of Frames
Detect the Object
Result Display
End

Fig. 4.3 Algorithm Description of modules

4.4 System Modeling:

4.4.1 Data-Flow Diagram:

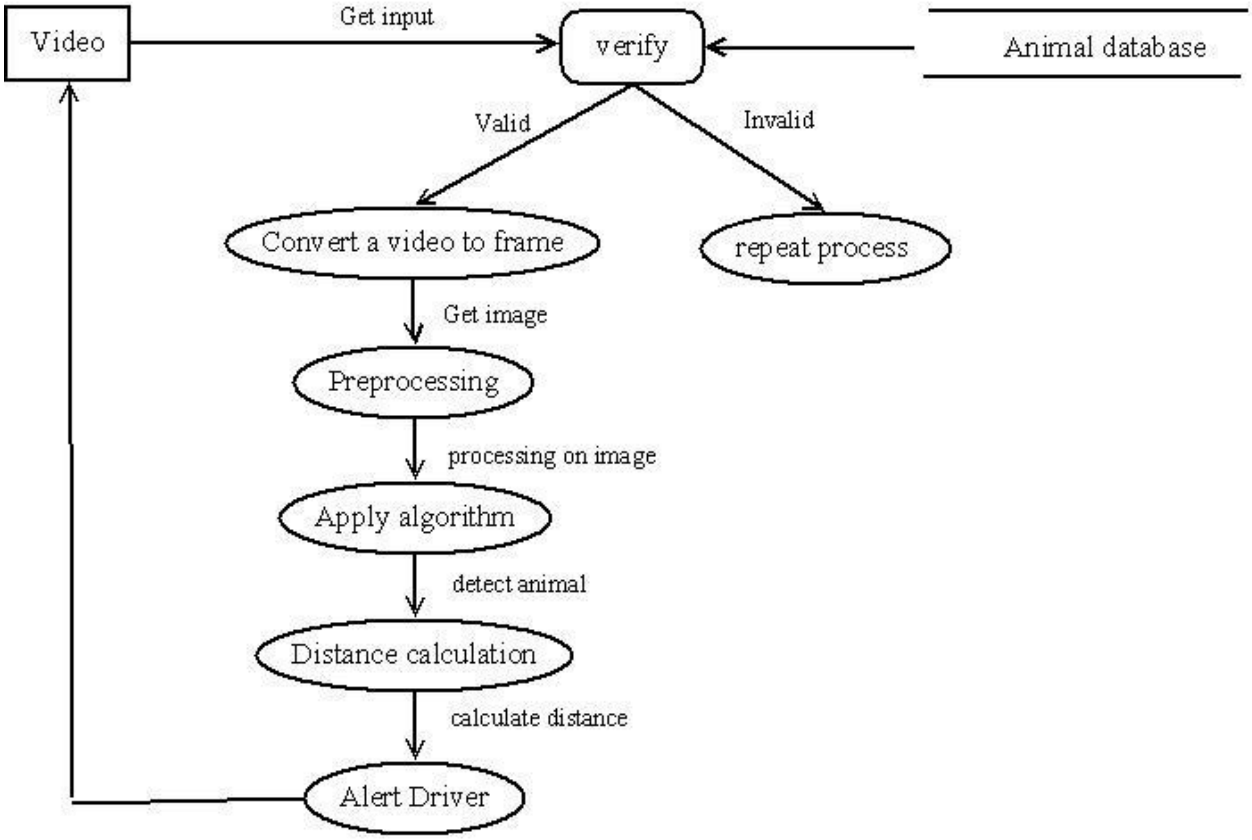


Fig.4.4.1 Data-Flow diagram

4.4.2 Sequence Diagram

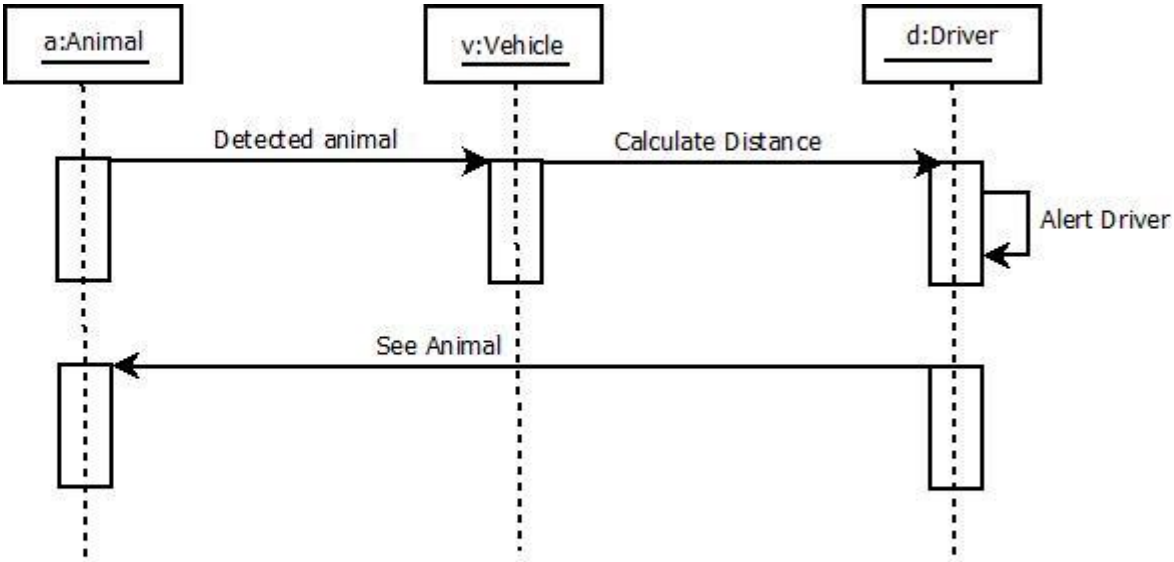


Fig. 4.4.2 Sequence diagram

4.3.3. Activity Diagram

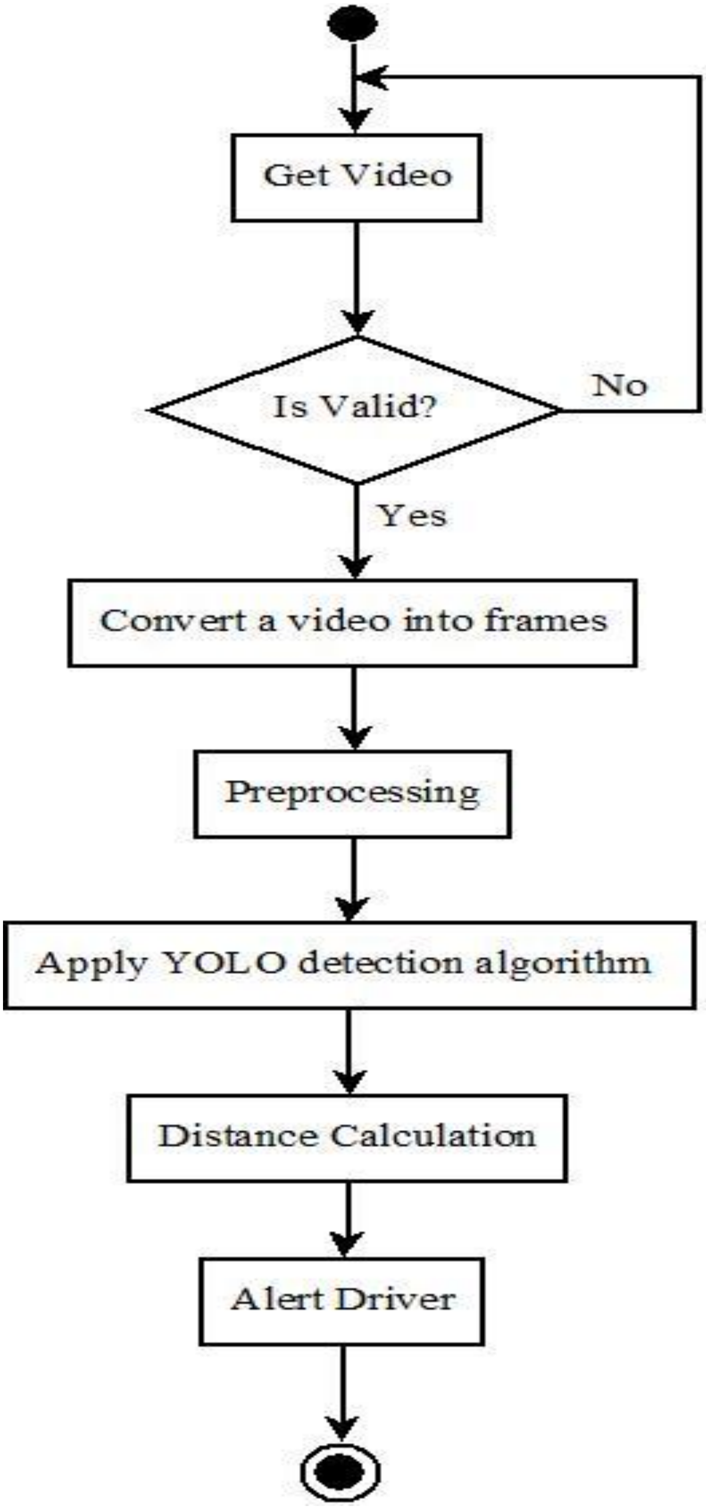


Fig.4.3.3 Activity diagram

5. Implementation

a. Environmental Setting for Running the Project

Steps to setup environment variable for python

1. Click start, then Control Panel, then System.

2. Click Advanced, then Environment Variables.

3. Add the location of the bin folder of the JDK installation for the PATH variable in the system variables. Then following is a typical value for the

Python path variable: C:\Python36:C:\Python36\Scripts;

b. Detailed Description of Methods

1. **def videoCapture()**:

In this method video is captured and video stream is initialized.

2. **def preprocessing():**

This method takes image as an input and converts it into de-noised and threshold image.

3. **def detectObject():**

This method construct a blob from the input frame and then perform a forward pass of the YOLO object detector, gives bounding boxes and associated probabilities and then extracts the class ID and confidence of the current object detection.

4. **def videoWriter():**

This method initialize video writer.

c. Implementation Details:

```
import numpy as np

import argparse

import imutils

import time
```

import cv2

import os

def videoCapture():

- Initialize the video stream.
- Initialize pointer to output video file.
- Initialize frame dimensions.

def preprocessing():

- Convert video into frames.
- Performs preprocessing on frames.

def detectObject():

- Construct a blob from the input frame.
- Perform a forward pass of the YOLO object detector, gives bounding boxes and associated probabilities.
- Initialize lists of detected bounding boxes, confidences, and class IDs.
- Extract the class ID and confidence of the current object detection.
- Filter out weak predictions.
- Scale the bounding box coordinates back relative to the size of the image.
- Use the center coordinates to derive the top and left corner of the bounding box.
- Apply non-maxima suppression to suppress weak, overlapping bounding boxes.
- Draw a bounding box rectangle and label on the frame.

def videoWriter():

- Initialize video writer.
- Write the output frame to disk.
- Release the file pointers.

6. Integration and Testing

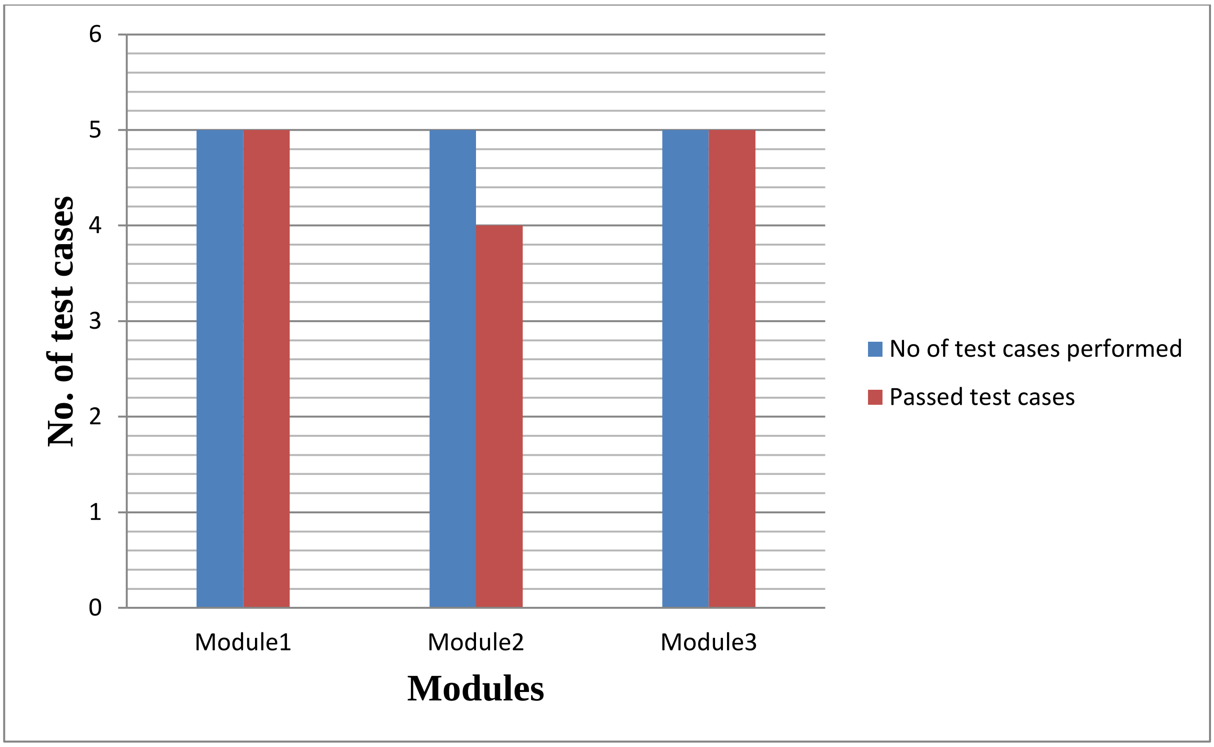
Test Performed	Expected	Actual
Sending video to the processor	The system should access the video.	Yes
Image Processing	Required to match characteristics of object.	Yes
Object Detection	Required marking must be present in bounding box for objects.	Yes

7. Performance Analysis

The performance analysis of this project is based on:

1) The quality of video as well as image.
2) Number of frames stored while processing video.
3) Probability of detection.

Performance increases as frames are stored while video in process and after getting correct object movement and animal detected.



8. Applications

The proposed project is useful at various places like:

1. Wild Life Monitoring
2. Military Purpose
3. Parking
4. Market Place
5. Tracking

9. Installation Guide and User Manual

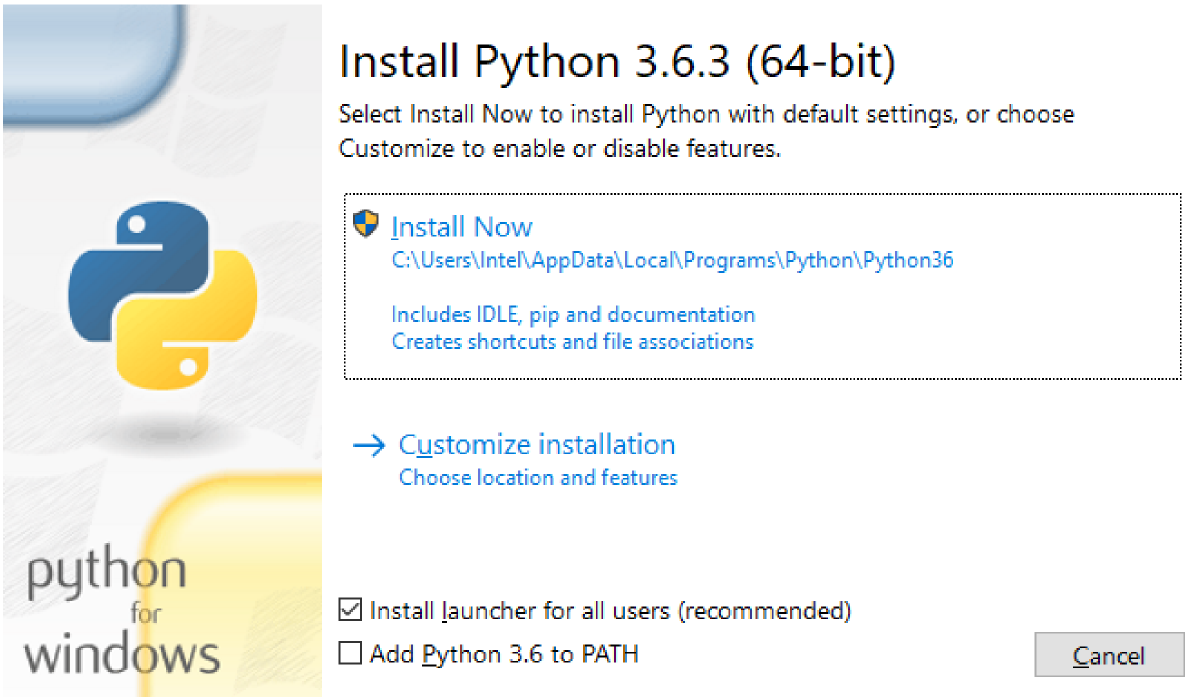
Installing python and OpenCV

To begin with, first we will download Python and OpenCV on our machine

1. Python3.6.2

2. OpenCV3.4.3

3. Installing Python:



Install Python 3.6.3 (64-bit)

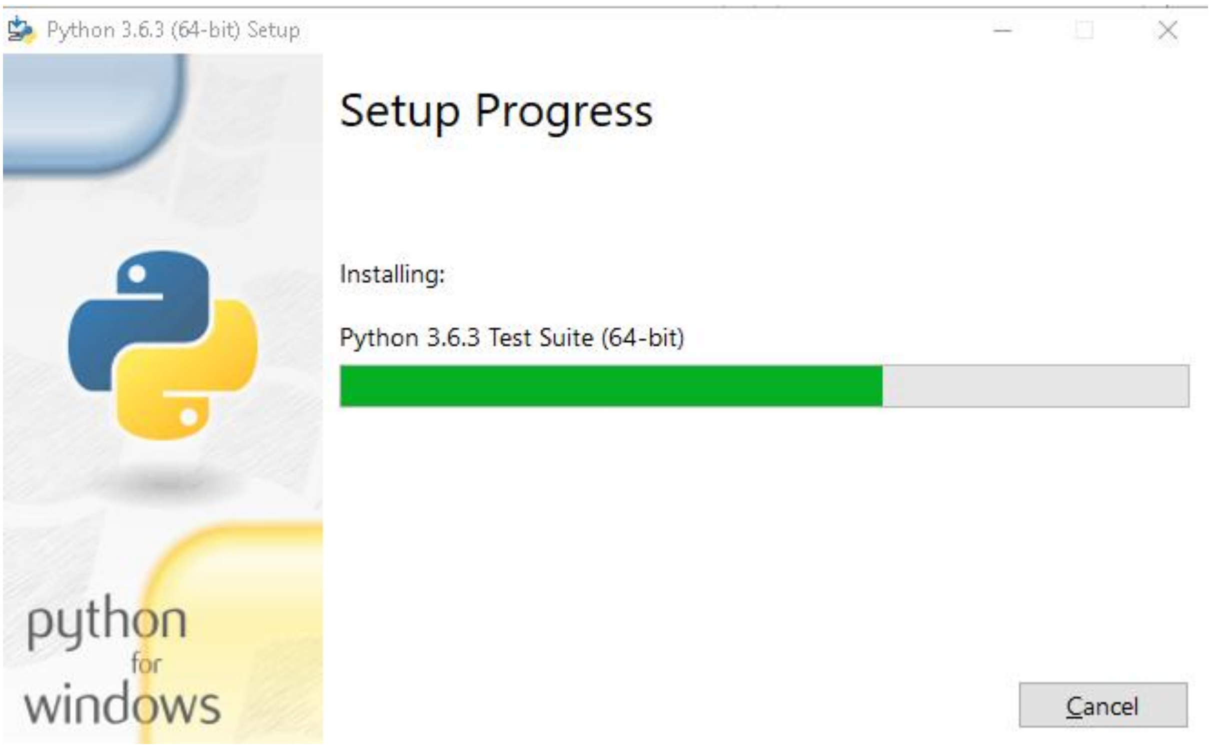
Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\Intel\AppData\Local\Programs\Python\Python36

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☐ Add Python 3.6 to PATH

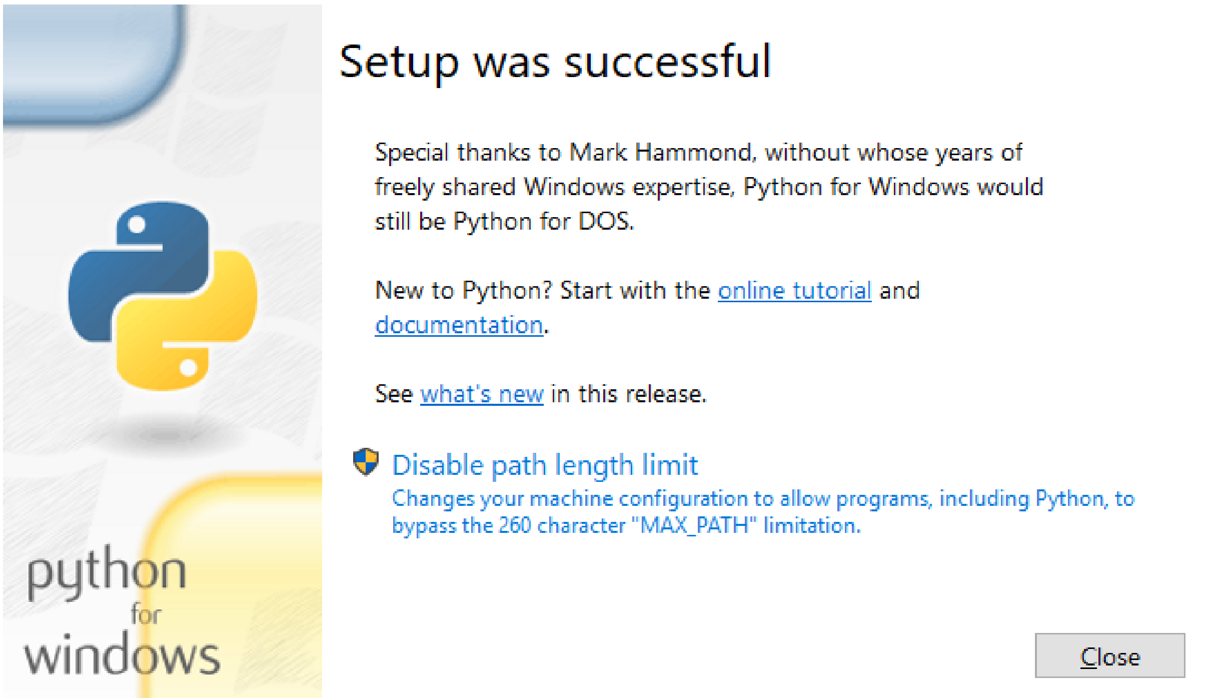


Python 3.6.3 (64-bit) Setup

Setup Progress

Installing:

Python 3.6.3 Test Suite (64-bit)



Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

See what's new in this release.

Disable path length limit

Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

10. REFERENCES

[1] Sharma S, Thaker J, Patel R. animal detection for highway security: IJSRD Vol 3, 2015; 2321-0613.

[2] David Forslund and Jon Bjarkefur Night vision animal detection: IEEE 2014.

[3] Mammeri A., Zhou D. Efficient animal detection system for smart cars using cascade classifiers:IEEE ICC 2014.

Bibliography

- www.opencv.org
- www.learnopencv.com
- www.pythonprogramming.net