Ambulance Minor project

Presented By
Logapriya S

AGENDA

- ABSTRACT
- OBJECTIVE
- INTRODUCTION
- LITERATURE REVIEW
- DESIGN AND METHODOLOGIES
- IMPLEMENTATION
- TESTING
- INPUT AND OUTPUT
- CONCLUSION
- WEB REFERENCES LINK
- PLAGIARISM REPORT OF PPT
- REFERENCES

ABSTRACT

This Intelligent Traffic Management System helps the Emergency Vehicles to pass through the signal without any delay. Traffic congestion and transportation delay on urban requirements are increasing worldwide. To improve the safety for both pedestrians and the vehicles Traffic signal is must. Emergency vehicles, like Ambulance have responsibility to reach patients or those who are met with accidents have to quickly transfer them to hospital.

Due to traffic signals they may be delayed for rescue operations. The Emergency vehicles are determined with the help of the camera. A camera for each road is placed in the signal. The cameras monitor the images and record continuously. The Haar Cascade algorithm is used to process the image and identify Emergency Vehicles in the signal. The algorithm continuously process the images to identify the emergency vehicles in each road and is compared with all the other roads. If it identifies the Emergency Vehicle it changes the signal to green for the particular road. If no Emergency Vehicle is detected the signal

OBJECTIVES

Aim of the Project:

The goal of each one is to reach at destination without wasting time. But resources provided by current infrastructures are limited. So the Traffic management at road is crucial to reduce waiting and traveling times. Even though present traffic light controlling system handles the traffic at intersections, many times congestion, accidents happened due to its poor performance. So Intelligent Traffic Management System is implemented.

Scope of the Project:

We have described a prototype system to detect the Emergency Vehicles using Haar Cascade algorithm. The camera in the signal records the images and then it is stored and the Haar Cascade algorithm processes the images to identify the Emergency Vehicles. The Emergency Vehicles is detected using the trained dataset that is trained to detect the Emergency Vehicles

INTRODUCTION

:

Due to the increased number of vehicles, Traffic controlling is a challenge. And hundreds of vehicles wait at every signal at any given. This traffic makes the Emergency Vehicle not to reach the destination in proper time. Every few minutes an ambulance crosses a signal to save someone who is in critical health condition.The traffic congestion of India is 71%.The average number of vehicles waiting at a signal at peak hours is reported to be 150 – 170 vehicles.The average time for an ambulance to cross a signal is 5 to 7 minutes.According to the survey from Road Transport and Highway Authority of India an ambulance should take 10 minutes to cross the distance of 4 km but due to traffic congestion the time taken to cover a distance of 4 km is 18 minutes.40 – 45 % of deaths takes place while the patient is in the ambulance due to delayed arrival to the destination.This death rate in the ambulance can be reduced to 5 – 8 % by making the ambulance to cross the signal without any delay and to reach the destination in time.

LITERATURE REVIEW

- Hashmi, Mohammad Farukh, and Avinash G. Keskar. "Analysis and monitoring of a high density traffic flow at T-intersection using statistical computer vision based approach. " Intelligent Systems Design and Applications (ISDA), 2012 12th International Conference on. IEEE, 2012.
- Hasan, Md Munir, et al. "Smart traffic control system with application of image processing techniques." Informatics, Electronics & Vision (ICIEV), 2014 International Conference on. IEEE, 2014.
- Kaviani, Razie, Parvin Ahmadi, and Iman Gholampour. "A new method for traffic density estimation based on topic model." Signal Processing and Intelligent Systems Conference (SPIS), 2015. IEEE, 2015.

LITERATURE REVIEW

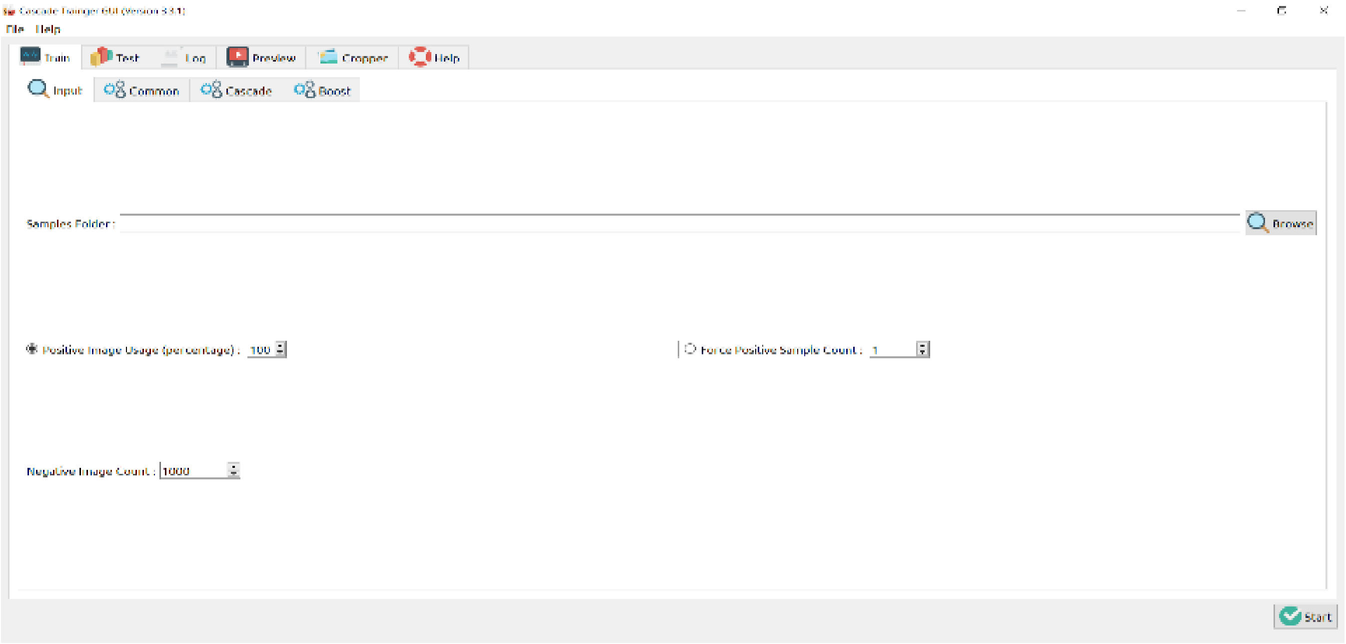
- Shi, Jianbo. "Good features to track." Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94.,1994 IEEE Computer Society Conference on. IEEE, 2017.
- Zhaoxiang Zhang, Yuqing Hou, Yunhong Wang, and Jie Qin. "A traffic flow detection system combining optical flow and shadow removal." In Intelligent Visual Surveillance (IVS), 2011 Third Chinese Conference on, pp. 45-48. IEEE,2011. International Journal of Computer Applications (0975 – 8887) Volume 180 –No.42, May 2018
  - Suárez, P.D. , Conci, A. , de Oliveira Nunes, E. “Video- Based Distance Traffic Analysis: Application to Vehicle Tracking and Counting” ,IEEE CS Journals and Magazines ,Volume: 13 , Issue:3 ,pp. 38- 45,2019

DESIGN AND METHODOLOGIES

- MODULE 1

CLASSIFIER TRAINING

Step:1 Install Cascade Trainer GUI . It is a windows application used to train the classifier.

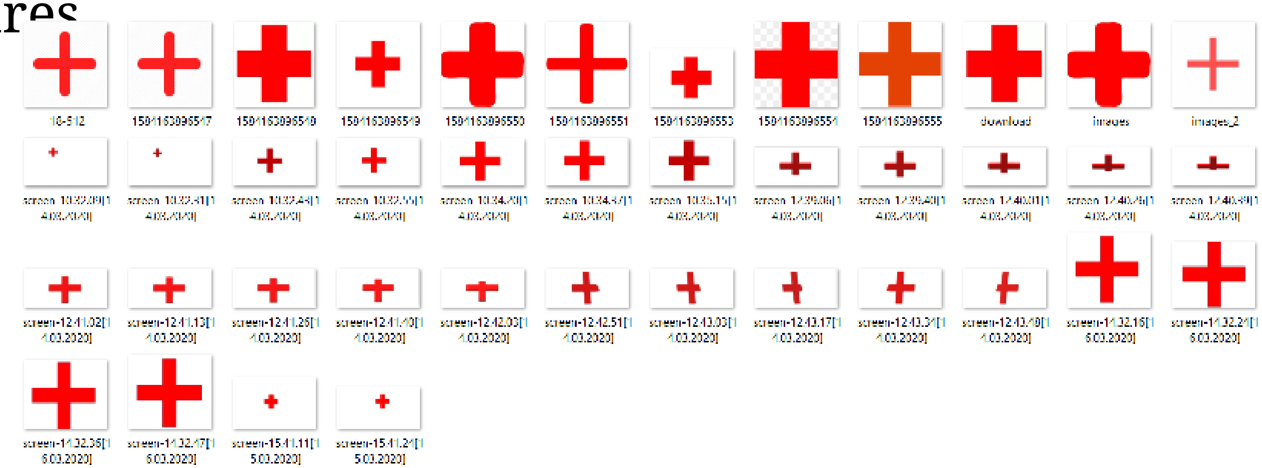


DESIGN AND METHODOLOGIES

- MODULE 2

IMAGE PROCESSING

Step 2: Training Positive Pictures



DESIGN AND METHODOLOGIES

- MODULE 3

Step 3: Training Negative Pictu

DESIGN AND METHODOLOGIES

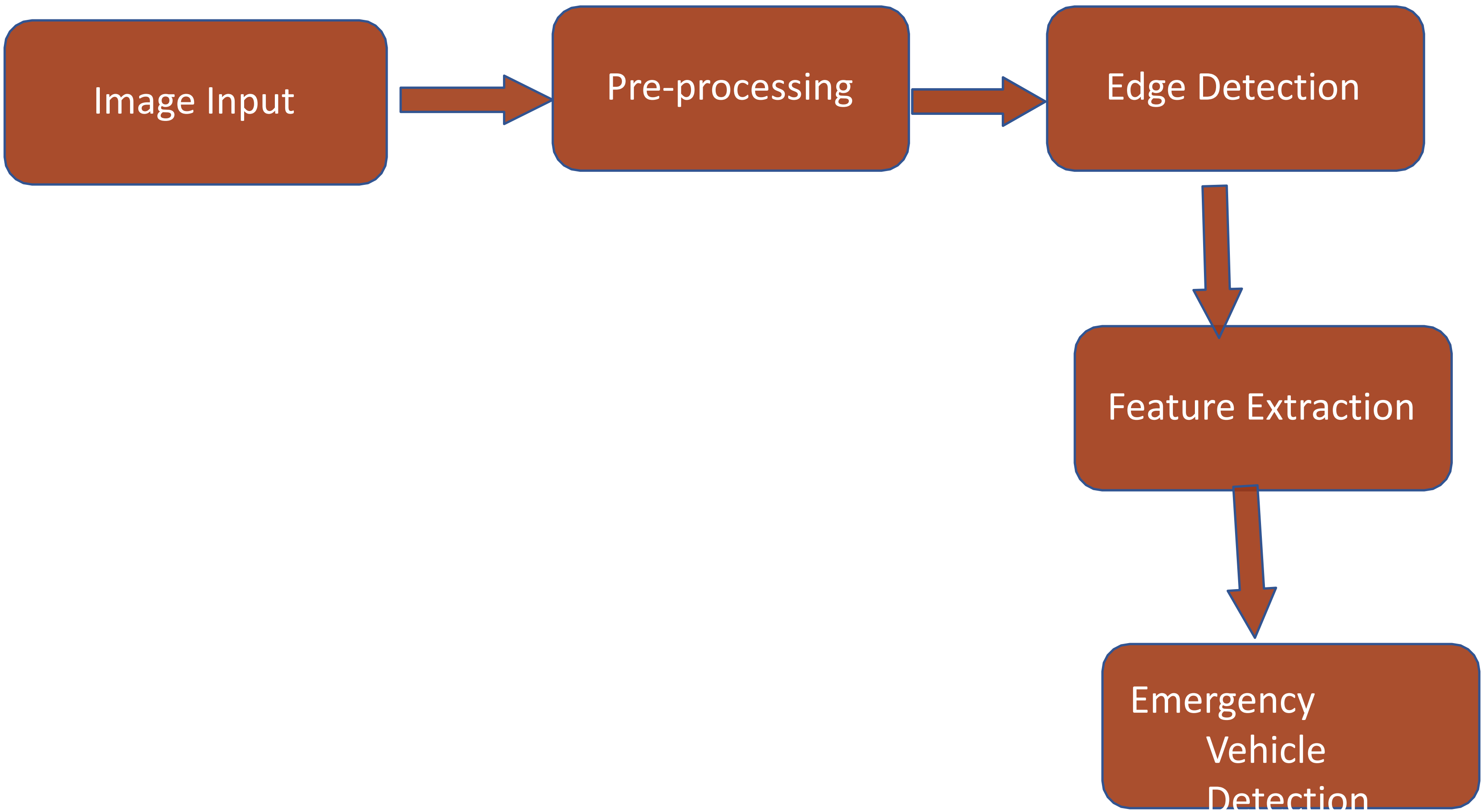
- MODULE 4

Step 4: Get the Output in XML file


```
<?xml version="1.0"?>
<opencv storage>
<cascade>
  <stageType>BOOST</stageType>
  <featureType>HAAR</featureType>
  <height>24</height>
  <width>24</width>
  <stageParams>
    <boostType>GAB</boostType>
    <minHitRate>9.9500000476837158e 01</minHitRate>
    <maxFalseAlarm>5.0000000000000000e 01</maxFalseAlarm>
    <weightTrimRate>9.4999998807907104e 01</weightTrimRate>
    <maxDepth>1</maxDepth>
    <maxWeakCount>100</maxWeakCount></stageParams>
  <featureParams>
    <maxCatCount>0</maxCatCount>
    <featSize>1</featSize>
    <mode>BASIC</mode></featureParams>
  <stageNum>3</stageNum>
  <stages>
    <!   stage 0   >
    < >
      <maxWeakCount>1</maxWeakCount>
      <stageThreshold>1.</stageThreshold>
      <weakClassifiers>
        < >
          <internalNodes>
            0  1 1  2.0211753249168396e 01</internalNodes>
          <leafValues>
            1.  1.</leafValues></ ></weakClassifiers></ >
    <!   stage 1   >
    < >
      <maxWeakCount>1</maxWeakCount>
      <stageThreshold>1.4285714924335480e 01</stageThreshold>
      <weakClassifiers>
        < >
          <internalNodes>
            0  1 2  4.0387761592864990e 01</internalNodes>
```

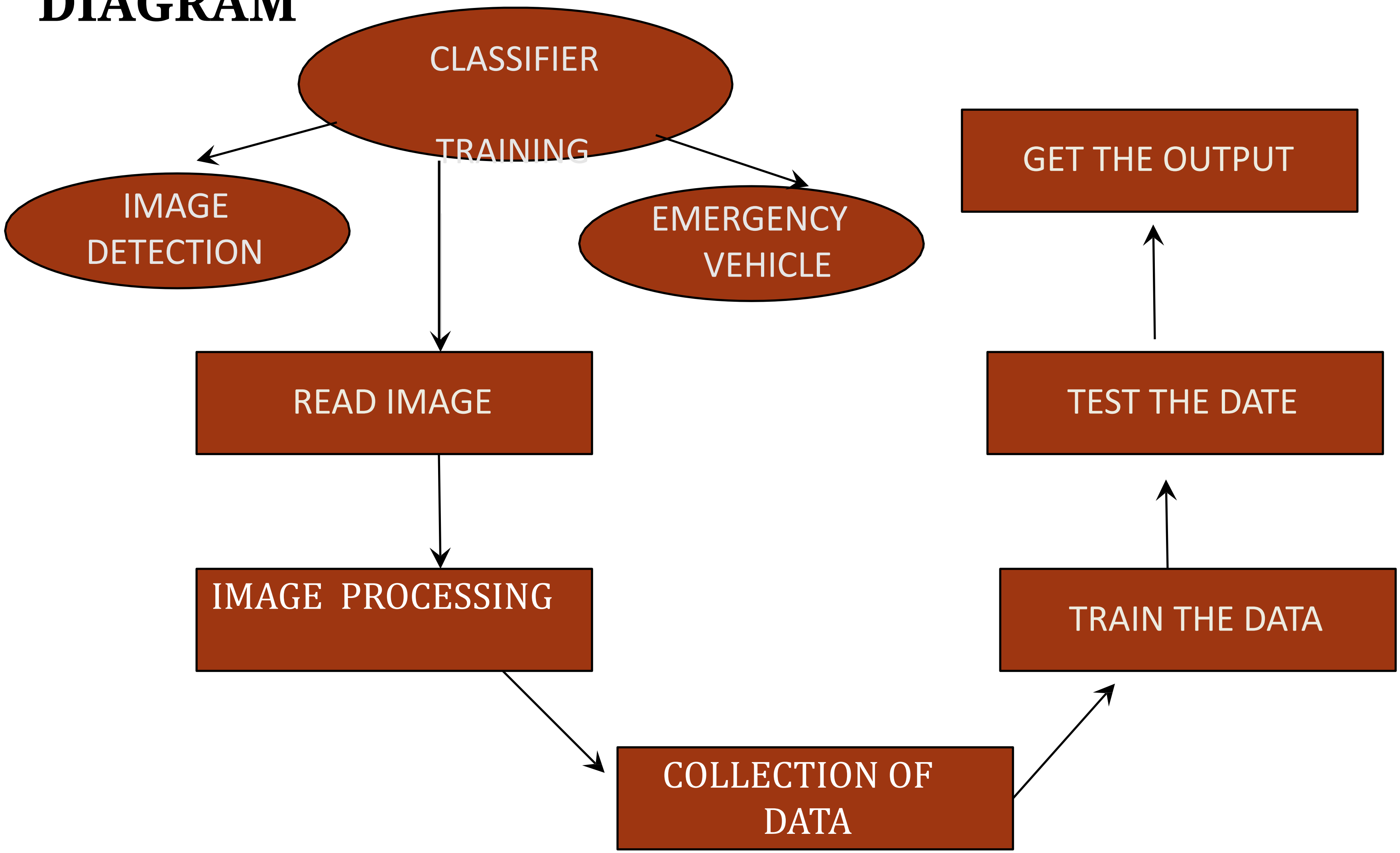
IMPLEMENTATION

ARCHITECTURE DIAGRAM

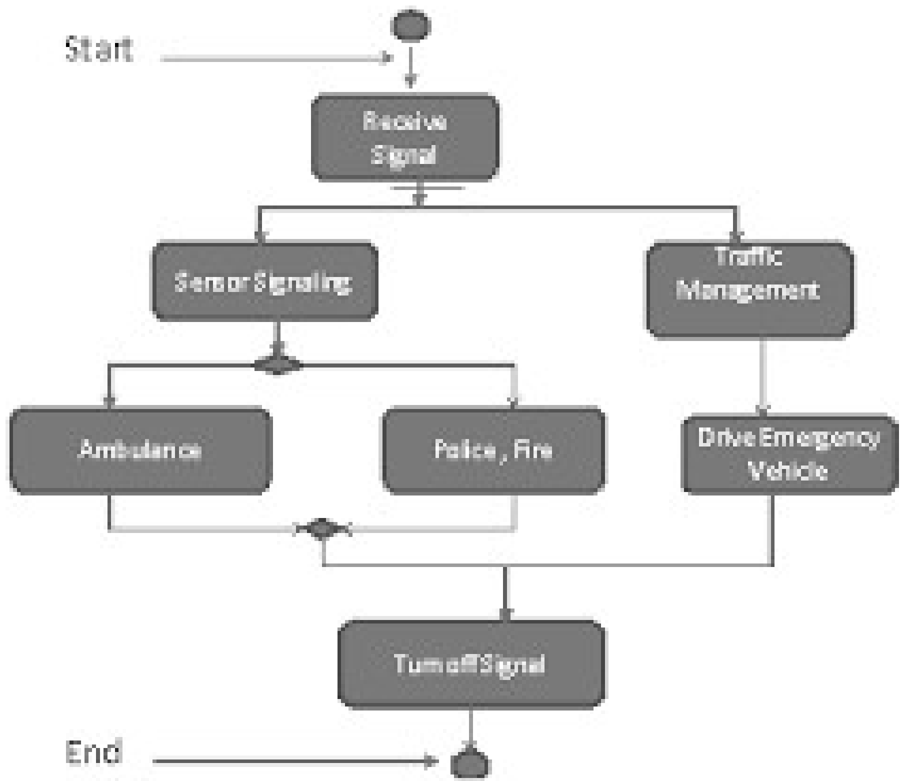


DATA FLOW DIAGRAM

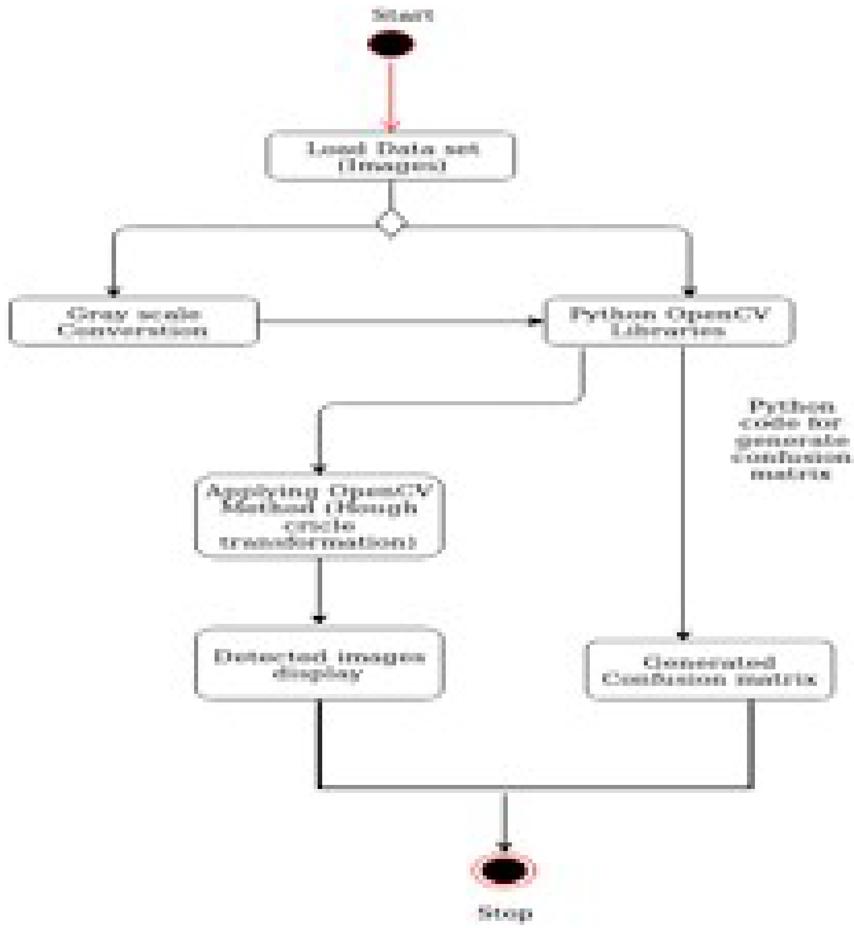
ER DIAGRAM


CLASSIFIER
TRAINING
IMAGE DETECTION
EMERGENCY VEHICLE
GET THE OUTPUT
READ IMAGE
TEST THE DATE
IMAGE PROCESSING
TRAIN THE DATA
COLLECTION OF DATA

- SEQUENCE DIAGRAM



- COLLABORATION DIAGRAM



TESTING

- UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid

outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the

completion of an individual unit before integration. This is a structural testing,
that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit

- INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

- FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested
are available as specified by the business and technical

requirements, system documentation, and user manuals.

Functional testing is centered on the following items:
Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

- WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

- BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge

of the inner workings, structure or language of the module being

tested. Black box

tests, as most other kinds of tests, must be written from a

definitive source document, such as specification or

requirements document, such as

specification or requirements document. It is a testing in which the software

under test is treated, as a black box. cannot “see” into it. The test provides inputs and responds to outputs without

considering how the software works.

Test Results: All the test cases mentioned above passed successfully. No

INPUT AND OUTPUT

SCREENSHOTS

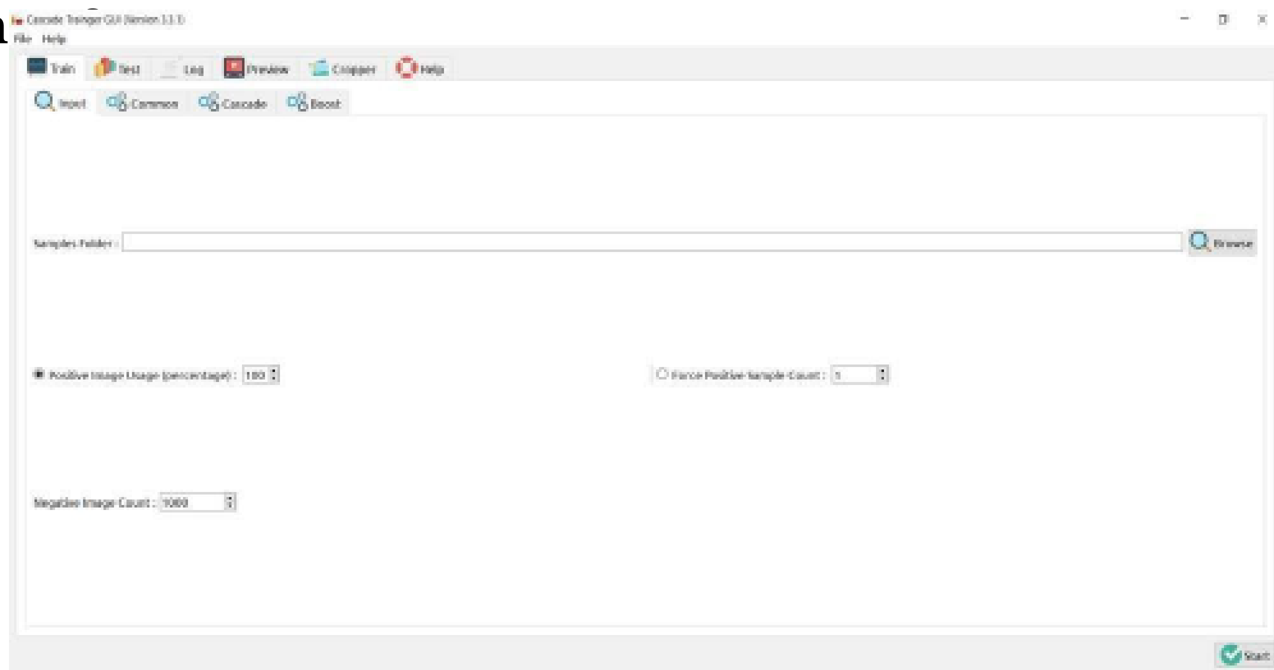
Traffic Intensity Analysis.py - D:\Project\Project Final\Project Demo\Traffic Intensity Analysis...

File Edit Format Run Options Window Help

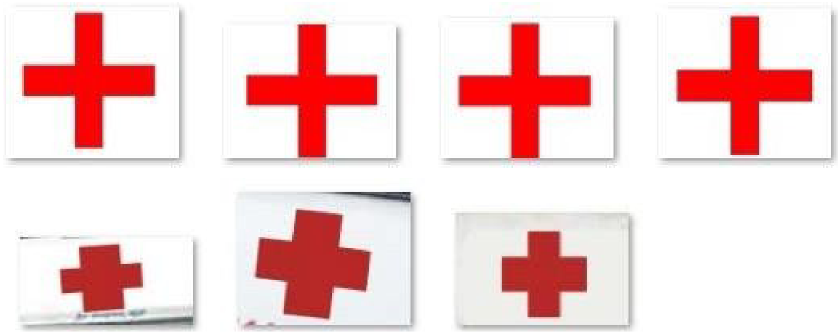
```
import time
import cv2
import urllib.request
import numpy as np
import imutils


cascade_src = 'cars.xml'
car_cascade = cv2.CascadeClassifier(cascade_src)

cascade_src1 = 'cascade 1.xml'
car_cascade1 = cv2.CascadeClassifier(cascade_src1)

ip_cam=["192.168.0.146:8080"]


try:
    while True:
        detected = [0]
        for ip in range(len(ip_cam)):
            url="http://"+ip_cam[ip]+"/shot.jpg"
            imgPath=urllib.request.urlopen(url)
            imgNp=np.array(bytearray(imgPath.read()),dtype=np.uint8)
            img=cv2.imdecode(imgNp, 1)
            img=imutils.resize(img,width=300)
            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            cars = car_cascade.detectMultiScale(gray, 1.1, 1)
            amb = car_cascade1.detectMultiScale(gray, 1.1, 1)

            for (x,y,w,h) in cars:
                cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),2)

            for (x1,y1,w1,h1) in amb:
                cv2.rectangle(img,(x1,y1),(x1+w1,y1+h1),(0,0,255),2)
                print("Ambulance detected..!!")
            cv2.imshow(str(ip), img)
            b=str(len(cars))
            a= int(b)
            n=a
```

NumPy is a library for the Python Programming Language adding support for large, multi-dimensional arrays and matrices, along with a large collection of high- level mathematical functions to operate on these arrays Imutils are a series

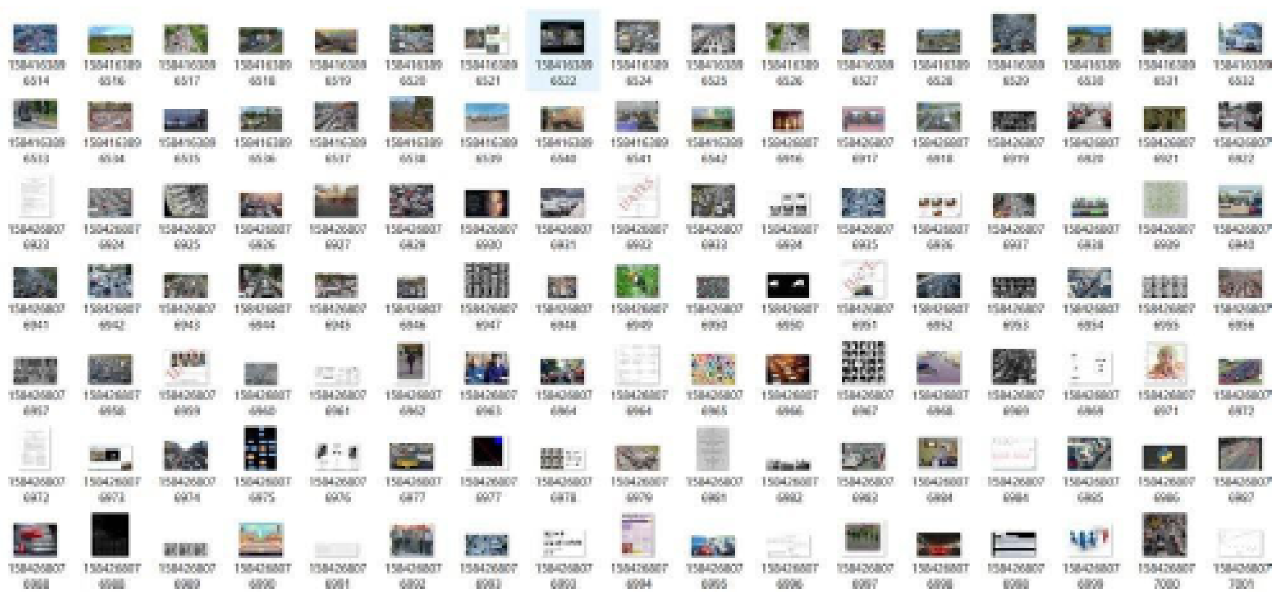
of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV and both Python 2.7 and Pyth



POSITIVE IMAGES



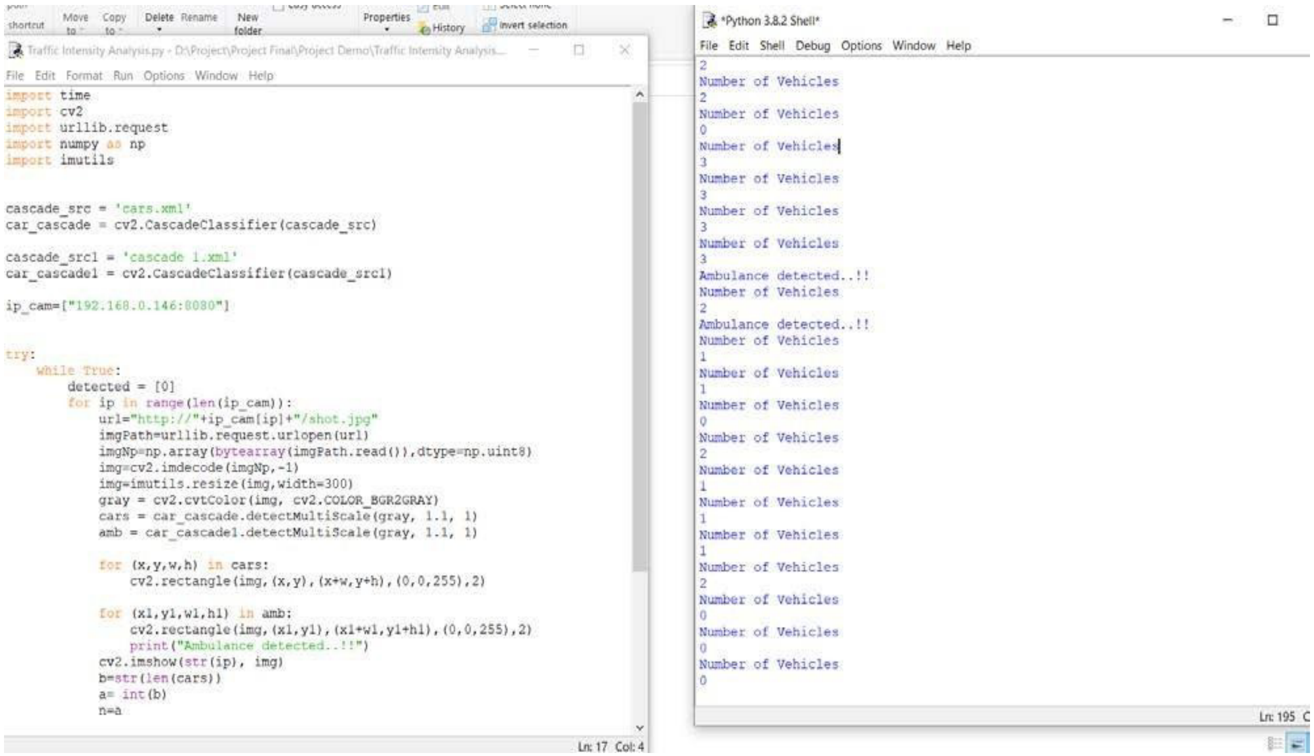
NEGATIVE IMAGES



GENERATED XML DOCUMENT (DATASET)

```xml
<?xml version="1.0"?>
<opencv_storage>
<cascade>
  <stageType>BOOST</stageType>
  <featureType>HAAR</featureType>
  <height>24</height>
  <width>24</width>
  <stageParams>
    <boostType>GAB</boostType>
    <minHitRate>9.9500000476837158e-01</minHitRate>
    <maxFalseAlarm>5.0000000000000000e-01</maxFalseAlarm>
    <weightTrimRate>9.4999998807907104e-01</weightTrimRate>
    <maxDepth>1</maxDepth>
    <maxWeakCount>100</maxWeakCount></stageParams>
  <featureParams>
    <maxCatCount>0</maxCatCount>
    <featSize>1</featSize>
    <mode>BASIC</mode></featureParams>
  <stageNum>3</stageNum>
  <stages>
```

- This is the XML file that contains the data
- It is also called the dataset
- The edge detection requires the data present in this file



- Number of vehicles is the number of vehicles detected on the road

- Ambulance detected..!! appears when the ambulance is detected

CONCLUSION

Proposed system works efficiently over the present traffic controlling system with respect to no waiting time, efficient operation during emergency mode. In this paper, we proposed a model which demonstrates the feasibility of using deep learning and neural networks to detect the objects.Overall concept is to make the ambulance cross the signal without any delay. Although the generated data set was small and did not represent the real-world scenario of the datasets about the vehicles, it was very helpful throughout the experiment. The process of detection of vehicles and ambulance is quick and easy under the trained model. By this we can also assure that under the real and large data set, cascade haar algorithm can be well-trained and also provides accurate results, which can help the people in recognizing the Emergency Vehicles, objects they come across easily.

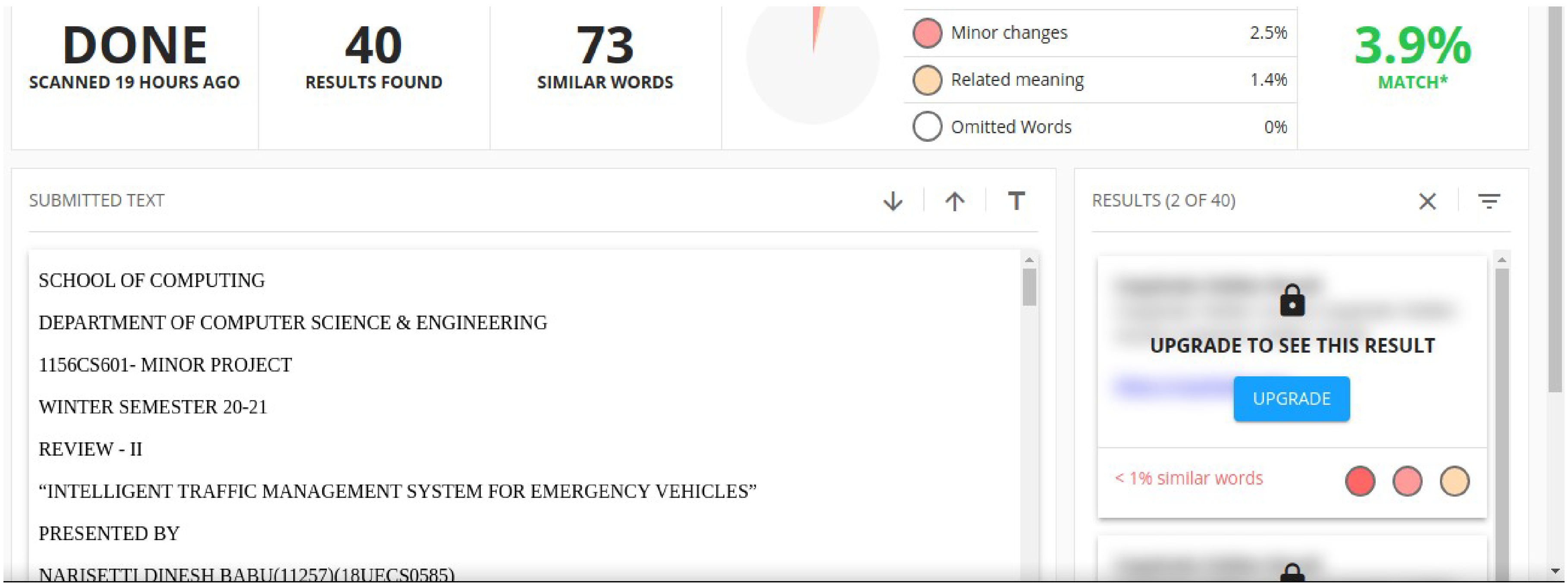
Web references/video links

Web link: https://ieeexplore.ieee.org/abstract/document/906 8695

https://towardsdatascience.com/emergency-vs-non-emergency-vehicle-classification-f0153c4f87f8

Video link: https://www.youtube.com/watch?v=DjQnVzp4SYo

Plagiarism Report of PPT



REFERENCES(as per IEEE format only)

- P. Soille, Morphological Image Analysis: Principles and Applications,Springer-Verlag, 2018, pp. 173-174.
- N. Otsu, "A Threshold Selection Method from Gray- Level Histograms,"IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp.62-66.
- Gopal Manne, Neetesh Raghuwanshi, "Vehicle Detection from Video using

  a Morphological Operations", International Journal of Science, Engineering and Technology Research (IJSETR) Volume 6, Issue 4, April 2097, ISSN: 2278 -7798.
- H. Jun-Wei, Y. Shih-Hao, C. Yung-Sheng, H. Wen- Fong, "Automatic traffic surveillance system for vehicle tracking and classification", IEEE Trans. Intell.Transp. Syst, vol. 7, no. 2, pp. 175-187, June 2020