A Study on Intelligent Traffic Monitoring System

Smt. K. S. Sukrutha[1], Miss. Rachana V[2]

[1]Associate Professor, Department of Computer Science, M.M.K & S.D.M Mahila Maha Vidyalaya, Mysuru, India
[2]II Semester MCA, [2]JSS Science & Technology University, Mysuru, India

Abstract:
Urban traffic has become increasingly congested in recent years due to the prevalence of personal vehicles. As a result, one of the major issues in major cities around the world is traffic. Congestion, accidents that take a lot of time and result in property damage, and environmental pollution are some of the traffic- related issues. In this research paper, an automated and intelligent traffic control system that makes use of computer vision and image processing techniques is designed and put into operation. The algorithm counts the number of vehicles on each route and provides an optimal waiting period based on the number of vehicles loaded onto each road. A dynamically managed traffic system can take the place of the traditional fixed-time, present traffic system using this completely automated approach. Low-cost image processing and traffic load balancing are the main topics of this study.

Keywords: Machine Intelligence, Traffic Control,Routing, Object Detection and Tensor Flow.

Introduction: The creation of precise machine- learning models that can locate and recognise numerous items in a single image is a major challenge in computer vision. The creation of object detection applications is now easier than ever. The open-source platform Tensor Flow's Object Identification API, built on top of Tensor flow, makes it simple to develop, train, and deploy object detection models. Item recognition from a scene has been one of the most researched subjects. Many methods have been proposed for object detection, however only a few of them have acceptable success rates. Furthermore, there are restrictions on how these technologies can be used in the real world, such as a fixed or white background. The primary objective of this study is to discuss a state-of-the-art approach for object recognition and object tracking in a hazy background. The unknown background could be as straightforward as a static white background or as complex as a scenario with several objects of varied sizes and forms. By creating a simulation website to load the object tracking results, this greatly enhances the application's usefulness in a real-world situation. The project's main objective is to compile a list of the steps in the object detection process.

Fig 1. object detection using TensorFlow

In industrialised countries, traffic management is now automated. Currently, there are two ways to manage traffic. One system uses pressure plates placed on the road to detect vehicles, and the other uses RFID tags installed on license plates and RFID readers placed nearby. Both methods are highly expensive to use when there is a significant traffic wait.

Disadvantages of Existing System:

- RFID tags cost more money
- More money
- Object detection takes longer
- Lower Accuracy

Proposed System: The newest technology makes finding things simpler and more precise. Information about moving objects is frequently virtually always necessary for surveillance videos, traffic analysis, human motion capture, etc. Applications that recognise moving objects in movies frequently employ background reduction methods.

Object Flow Diagram

A flowchart that illustrates how one activity leads to another is an activity diagram. A system operation could be used to describe the action.

Below is a diagram of our application's activities. The user must log in to the software by entering the correct username and password before continuing with the next step.

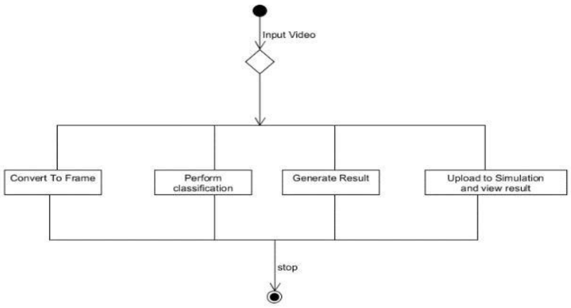


Fig 2. Object Flow diagram

System Architecture

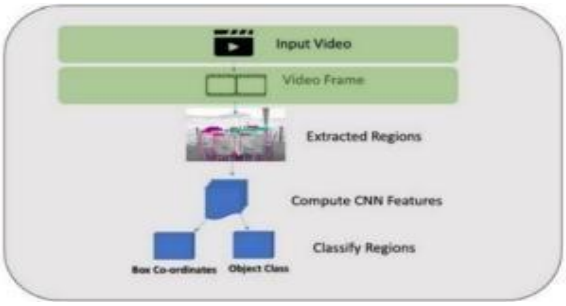


Fig 3. System Architecture

Implementation of the System

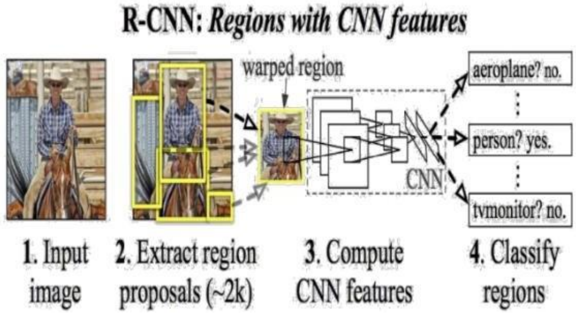


Fig 4. CNN Features

Product Functions

- Pre-Processing
- Traffic video has been live detected
- Frames are extracted from a video.
- Each frame is converted from RGB to Grayconversion

MethodologyModel CNN

- Objects are detected in images and featureextraction.
- The extracted and Google-trained datasetsare compared using TensorFlow
- The compared data is predicted using CNN.
- Traffic object identification and detected result extraction process are stored in a CSVfile.
- Created a simulation website to load the traffic object details which are stored and fetched from MySQL.
- Fetched cumulated result is provided to the user by visual view for finding the shortest and/or time-saving path

1. **Input image**: Produce and extract candidate bounding boxes, for example, category-independent region proposals
2. **Extract region proposals:** Using a deep convolutional neural network, for instance, extracting features from each potential region.
3. **Compute CNN features:** Identify features as belonging to a recognized class, such as the linear SVM classifier model.
4. **Classify regions**: Use a protocol to upload the traffic object detection results to a website and give users a visual representation of the traffic selection.

Test the reasons using optimistic scenarios

TCNO	Positive scenario	Required Input	Expected output	Actual output	Test Result
1	Upload datasets	Upload video	Should upload	successfully uploaded	Pass
2	Pre-processing	Process dataset	Remove unwanted datasets	Unwanted datasets are removed	Pass
3	train video	Video processing	Identify object	Object detected	Pass
4	Classification	Objects are classify	Identify the object and classify which type of object it is	Object classified	Pass
5	Performance analysis	Find Accuracy	Display Accuracy information	Accuracy information displayed	Pass

Table 1. System Test Case Table

Uploading File

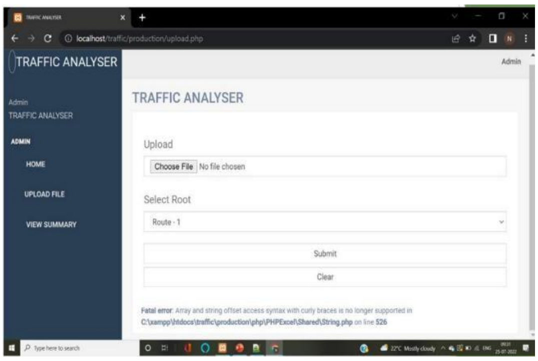


Fig 5: Uploading video

The extracted result dataset must be uploaded bychoosing the root, and we can submit files up to 4

Sample Output

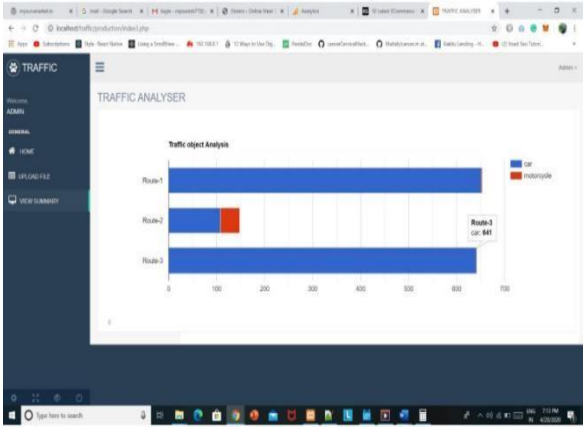


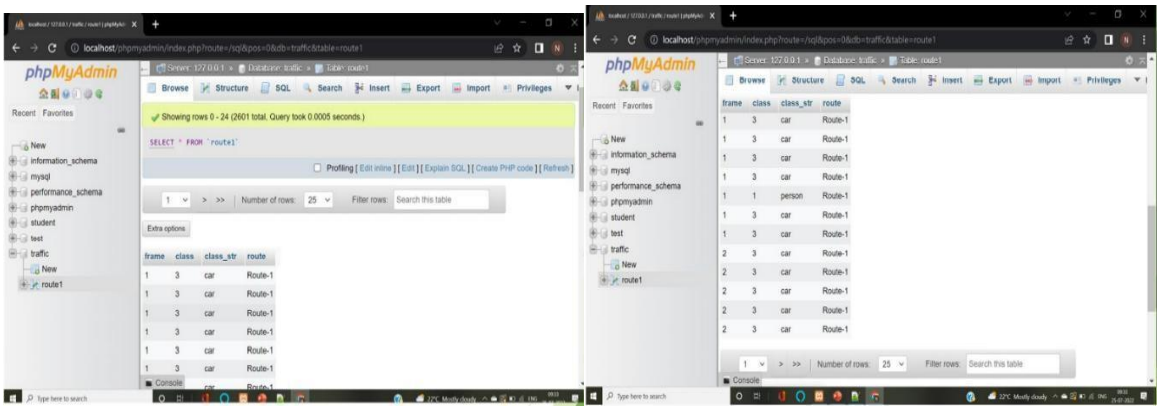
Fig 6: Analysis of Traffic

Expected Research Outcome

- PHP is used in the middle layer of a website that uses HTML, CSS, or bootstrap for the front end and provided a simulation protocol to display traffic objects graphically. We'll come to a conclusion about which route takes the least amount of time later using theframe threshold, which is utilized to determine the shortest path.
- We enter a video into the system, whichrecognizes it and converts it into frames before displaying the route with the leastamount of traffic or the shortest travel time.
- With the aid of this initiative and using the results of the studies, we are now better able to recognise things and pinpoint their locations in the environment.
- This study also provides experimental data on the effectiveness of several strategies for item detection and identification.

Following the loading of the traffic object identification findings onto the simulation website, MySQL will be used to calculate the traffic time.
routes. Based on the frame threshold, we will determine which route requires the most amount of time (high to low).

Data Base Structure



Conclusion

Based on the testing results from this research, we can identify each object separately and determine where it is located on the x and y axes in the image. The effectiveness of each method for item detection and identification is examined in this study, along with the outcomes of various experiments. The simulation website will load the traffic object detection results and use them to retrieve data from MySQL to calculate traffic time.

Future Enhancement

According to this research paper, we may be able to expand its application to other domains, such as traffic, aerospace, or any other domain to detect objects by using the Tensor Flow library. With the help of cutting-edge technology, this project provides a machine learning and video processing approach that makes it possible to recognise objects in a given movie**.**

Bibliography

Web References:

1. www.researchgate.net
2. www.sciencedirect.com
3. www.irjet.net
4. www.tensorflow.org

Article References:

1. P. F. Felzenszwalb, R. B. Girshick, D. McCallister, and D. Ramanan, "Object detection with discriminatively trained part-based models," IEEE Trans. Pattern Anal. Mach. Intel., vol. 32, no. 9, p. 1627, 2010.
2. K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection," IEEE Trans. Pattern Anal. Mach. Intel., vol. 20, no. 1, pp. 39–51, 2002
3. C. Wojek, P. Dollar, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," IEEE Trans. Pattern Anal. Mach. Intel., vol. 34, no. 4, p. 743, 2012

4. H. Kobatake and Y. Yoshinaga, "Detection of spicules on mammogram based on skeleton analysis." IEEE Trans. Med. Image., vol. 15, no. 3, pp. 235–245, 1996
5. Y.Jia, E.Shelhamer, J.Donahue, S.Karayev, J.Long, R.Girshick, S. o Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in ACM MM, 2014.
6. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in NIPS, 2012.
7. Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in CVPR, 2017
8. Z. Yang and R. Nevatia, "A multi-scale cascade fully convolutional network face detector," in ICPR, 2016
9. C. Chen, A. Seff, A. L. Kornhauser, and J. Xiao, "Deep driving: Learning affordance for direct perception in autonomous driving," in ICCV, 2015
10. X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in CVPR, 2017

11. A. Dundar, J. Jin, B. Martini, and E. Culurciello, "Embedded streaming deep neural networks accelerator with applications," IEEE Trans. Neural Netw. & o Learning Syst., vol. 28, no. 7, pp. 1572– 1583, 2017.
12. R. J. Cintra, S. Duffner, C. Garcia, and A. Leite,"Low-complexity approximate convolutional neural networks," IEEE Trans. Neural Netw. & o Learning Syst., vol. PP, no. 99, pp. 1–12, 2018.