SYNOPSIS

Report on

IMAGE PROCESSING BASED TRACKING AND COUNTING VECHILE

by

Ankit Gangwar (2100290140028)

Anshul Yadav (2100290140030)

Session:2022-2023 (4th Semester)

Under the supervision of

Dr. Akash Rajak
Professor

KIET Group of Institutions, Delhi-NCR, Ghaziabad

DEPARTMENT OF COMPUTER APPLICATIONS
KIET GROUP OF INSTITUTIONS, DELHI-NCR, GHAZIABAD-201206

(MARCH- 2023)

ABSTRACT

This project involves the development of an image processing-based tracking and counting system for vehicles using Python. The system utilizes a video feed from a surveillance camera to detect and track vehicles as they move through the camera's field of view. The tracking algorithm relies on computer vision techniques such as image segmentation, feature extraction, and object tracking.

The system also includes a counting module that keeps track of the number of vehicles passing through the camera's field of view. The counting algorithm uses the vehicle tracking data to determine when a vehicle enters or exits the field of view.

Python is used as the programming language for this project due to its robust libraries for image processing and computer vision, such as OpenCV and NumPy. The system is designed to be highly scalable and can be easily integrated into existing traffic management systems to improve traffic flow and reduce congestion.

TABLE OF CONTENTS

INTRODUCTION

The rapid increase in the number of vehicles on roads and highways has led to a significant increase in traffic congestion and accidents, making effective traffic management a critical challenge for cities and countries worldwide. To address this challenge, there is a growing need for automated vehicle tracking and counting systems that can provide accurate and real-time traffic data.

Image processing-based tracking and counting systems have emerged as a promising solution for traffic surveillance, offering a non-intrusive and cost-effective way to monitor traffic flow. These systems use computer vision techniques to analyse video feeds from surveillance cameras, detect and track vehicles, and count the number of vehicles passing through a particular area.

Python has become a popular choice for implementing image processing-based tracking and counting systems due to its rich collection of libraries for computer vision, such as OpenCV and NumPy. Python's simplicity and readability make it easy to develop and maintain complex image processing algorithms.

In this project, we propose the development of an image processing-based tracking and counting system for vehicles using Python. The system will utilize a video feed from a surveillance camera to detect and track vehicles as they move through the camera's field of view. The tracking algorithm will rely on computer vision techniques such as image segmentation, feature extraction, and object tracking. Additionally, the system will include a counting module that keeps track of the number of vehicles passing through the camera's field of view.

The proposed system offers several advantages over traditional manual traffic counting methods, such as increased accuracy and real-time data collection. The system is designed to be scalable, easily integrated into existing traffic management systems, and can be deployed in a variety of environments, including highways, parking lots, and intersections.

LITERATURE REVIEW

Image processing-based tracking and counting systems have been the focus of research in the field of traffic surveillance and management for several years. These systems offer a non-intrusive and cost-effective way to monitor traffic flow, providing real-time data for traffic analysis and planning.

One of the earliest and most widely used image processing-based tracking and counting systems is the Automatic Traffic Recording and Analysis System (ATRAS) developed by the US Federal Highway Administration (FHWA) in the early 1970s. ATRAS used image processing techniques to detect and count vehicles passing through a camera's field of view, providing valuable data for traffic planning and analysis.

Since then, several researchers have proposed various methods for vehicle tracking and counting using image processing techniques. For instance, researchers have explored the use of object detection algorithms, such as the YOLO algorithm, to detect and track vehicles in real-time. Others have used machine learning techniques, such as support vector machines (SVMs), to classify vehicles and track their movements.

Python has emerged as a popular choice for implementing image processing-based tracking and counting systems due to its rich collection of libraries for computer vision, such as OpenCV and NumPy. Python's simplicity and readability make it easy to develop and maintain complex image processing algorithms.

In recent years, researchers have also explored the use of deep learning techniques for vehicle tracking and counting. Deep learning algorithms, such as convolutional neural networks (CNNs), have been used to detect and classify vehicles in real-time, achieving high accuracy rates.

Overall, image processing-based tracking and counting systems have demonstrated significant potential for improving traffic management and safety. This paper proposes the development of a tracking and counting system for vehicles using Python, building on existing research in this field. The proposed system will utilize advanced computer vision techniques to detect, track, and count vehicles passing through a camera's field of view, providing accurate and real-time data for traffic analysis and planning.

TECHNOLOGIES USED

The technologies used for developing this project are:

PYTHON: That offers rich libraries for image processing, computer vision, and machine learning, such as OpenCV and NumPy. Python's simplicity and readability make it easy to develop and maintain complex image processing algorithms.

OpenCV: OpenCV is an open-source computer vision library that provides a variety of functions for image and video analysis, including image segmentation, feature extraction, and object detection and tracking.

NumPy: NumPy is a Python library that provides a powerful array manipulation capability. It allows efficient numerical operations on multi-dimensional arrays and matrices, making it an ideal choice for handling image data.

Object Detection Algorithms: Object detection algorithms, such as the YOLO algorithm, will be used to detect and track vehicles in real-time. These algorithms utilize deep learning techniques to classify objects and track their movements.

SOFTWARE REQUIREMENTS

The software environment used for developing the application is:

- Operating System - Windows
- Code editor – Microsoft Visual Studio Code
- PYTHON- Latest Version
- NumPy- The lateat version of NumPy 1.x should be installed.
- OpenCV- The latest version of OpenCV 4.x should be installed. It is recommended to install OpenCV with the contrib modules to access additional features.

HARDWARE REQUIREMENTS

Hardware components required for installing all the required software environment and tools are:

Minimum Hardware Requirements:

1. Processor: Intel Core i5 or equivalent.
2. RAM: 8 GB.
3. Storage: 50 GB of free disk space.
4. Graphics Card: Intel integrated graphics or equivalent.
5. Monitor: 1024x768 resolution.
6. Camera: Minimum resolution of 720p.

Maximum Hardware Requirements:

1. Processor: Intel Core i7 or equivalent.
2. RAM: 32 GB.
3. Storage: 100 GB of free disk space or more.
4. Graphics Card: NVIDIA GeForce RTX 3090 or equivalent.
5. Monitor: 4K resolution or higher.
6. Camera: High-quality camera with 1080p or higher resolution.

MODULES IN PROJECT

The following are the modules that may be included in the development of an image processing-based tracking and counting system for vehicles using Python:

Video Input Module: This module captures the video feed from the surveillance camera.

Image Preprocessing Module: This module applies various image preprocessing techniques, such as resizing, cropping, and normalization, to enhance the quality of the input image.

Object Detection Module: This module detects and localizes objects in the input image, using advanced computer vision techniques such as Haar cascades, HOG features, or deep learning-based algorithms like YOLO.

Object Tracking Module: This module tracks the objects detected by the object detection module, using techniques such as optical flow, Kalman filters, or correlation trackers.

Vehicle Classification Module: This module classifies the objects detected and tracked as vehicles, using machine learning techniques such as SVM or CNN.

Vehicle Counting Module: This module counts the number of vehicles passing through the camera's field of view, using various techniques such as background subtraction, frame differencing, or line detection.

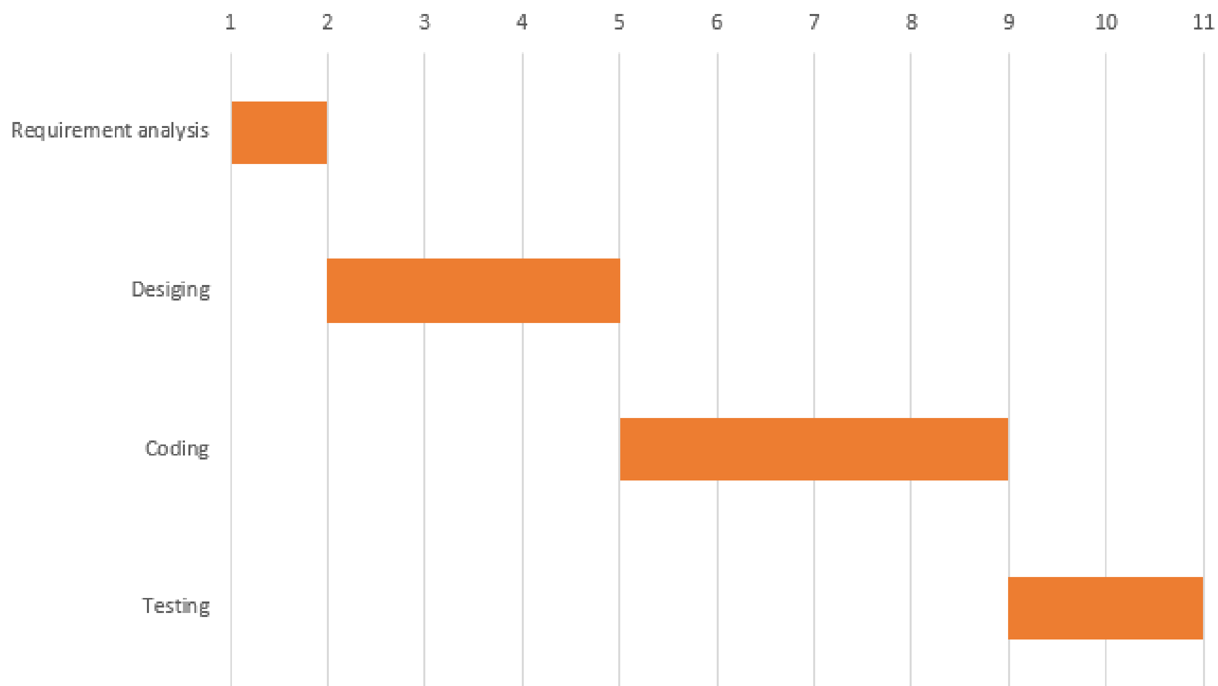
Database Management Module: This module stores the data generated by the system, such as vehicle count, vehicle classification, and traffic flow information, in a database.

User Interface Module: This module provides a user interface to interact with the system, displaying the output generated by the system and allowing the user to configure the system's settings.

Integration Module: This module integrates the system with other traffic management systems or devices, such as traffic lights, sensors, or message boards.

Proposed Time Duration

Gantt Chart:



.

Conclusion

In conclusion, the proposed image processing-based tracking and counting system for vehicles using Python is a promising solution for traffic management and monitoring. The system leverages advanced computer vision techniques, such as object detection, tracking, and classification, to accurately count and track vehicles passing through a camera's field of view. The system's modular design allows for flexibility in customization and integration with other traffic management systems and devices.

The proposed system's software requirements are minimal, with most of the required software being open-source and freely available. The system's hardware requirements are also flexible, with a range of specifications that can be customized based on the project's specific needs.

Overall, the proposed system has the potential to improve traffic management, reduce congestion, and enhance safety on roads. The system can be applied to various scenarios, including highways, toll booths, parking lots, and other traffic-intensive areas. As such, the proposed system is a valuable contribution to the field of computer vision and traffic management, with potential applications in smart cities and intelligent transportation systems.