

**Artificial Intelligence**



Report

**AI in smart cities(vehicle counting)**

**Nunna Kamala 21KN1A6313**

**Mogarampalli Hema 21KN1A05A7**

**Kota Saisravani 21KN1A0590**

**Malempati Sathivika 21KN1A0590**

**INTRODUCTION:**

A smart city is a city that uses advanced technology and data-driven solutions to improve the quality of life for its citizens, enhance sustainability, and optimize urban services. These solutions are typically designed to address challenges that cities face, such as traffic congestion, pollution, energy consumption, and safety. A smart city may incorporate a range of technologies, including sensors, data analytics, automation, and artificial intelligence. For example, smart transportation systems may use sensors and data to optimize traffic flow, reduce congestion, and improve public transit. Smart energy systems may use data and automation to manage power grids, reduce energy consumption, and integrate renewable energy sources. Ultimately, the goal of a smart city is to use technology to create more efficient, sustainable, and livable urban environments.

Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think and act like humans. It involves the development of computer algorithms and systems that can perform tasks that would otherwise require human intelligence, such as learning, problem-solving, decision-making, perception, and language understanding.

AI is a multidisciplinary field that draws on computer science, mathematics, engineering, psychology, and linguistics. There are different approaches to AI, including rule-based systems, machine learning, and deep learning.

Rule-based systems rely on a set of pre-programmed rules that determine how a machine will respond to different inputs. Machine learning, on the other hand, involves the use of algorithms that enable machines to learn from data and improve their performance over time. Deep learning is a subset of machine learning that uses neural networks to process large amounts of data and make predictions based on patterns.

AI has many practical applications, including speech recognition, image recognition, natural language processing, autonomous vehicles, robotics, and healthcare. It has the potential to transform many industries and improve our lives in many ways. However, it also raises important ethical and social issues, such as privacy, bias, and job displacement, that need to be addressed.

**SCOPE:**

Vehicle counting is a common application of computer vision and AI technology that involves detecting and tracking vehicles in real-time from video feeds or images. The technology has several important use cases, including traffic management, parking management, toll collection, and security monitoring.

In traffic management, vehicle counting can help authorities to monitor traffic flow and congestion, optimize traffic signals, and plan road infrastructure improvements. In parking management, vehicle counting can help to identify available parking spaces and optimize parking utilization. Toll collection agencies can also use vehicle counting to accurately charge toll fees based on the number of vehicles passing through a toll plaza.

In security monitoring, vehicle counting can help to detect suspicious vehicles and track their movements, which can be useful in preventing crime or terrorist activities. In addition, the technology can also be used in retail analytics to monitor customer traffic and optimize store layouts.

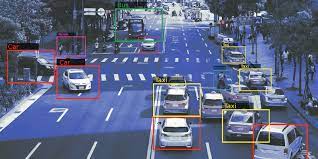
**Helps traffic police:** A vehicle detection and counting system could be beneficial for the traffic police because everything they can monitor from one place only likes how many vehicles have crossed this toll and which vehicle.

**Maintaining records:** It is challenging for some individuals to record all the vehicles with them because the cars are passing by in real-time. It’s not like that one is watching the video, and they can pause it and have a note of it, so to remove this limitation, this application can be very well-versed to attain the time-saving quality and be automated.

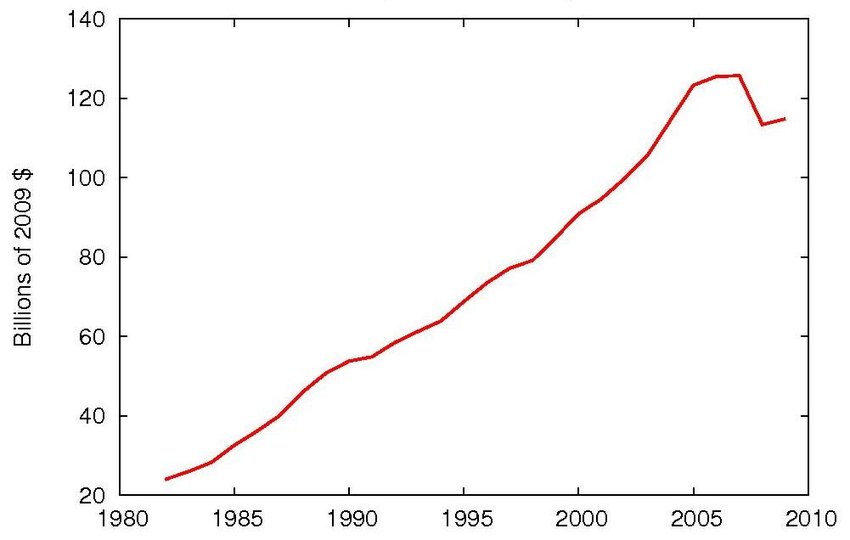
**Traffic surveillance control:** As this application can be planted anywhere as it only requires a camera or some wires (for establishing the connectivity with the central system) hence if the traffic is high at someplace, then from that area, an officer can monitor it and forward the information to next toll officer so that they could be prepared beforehand.

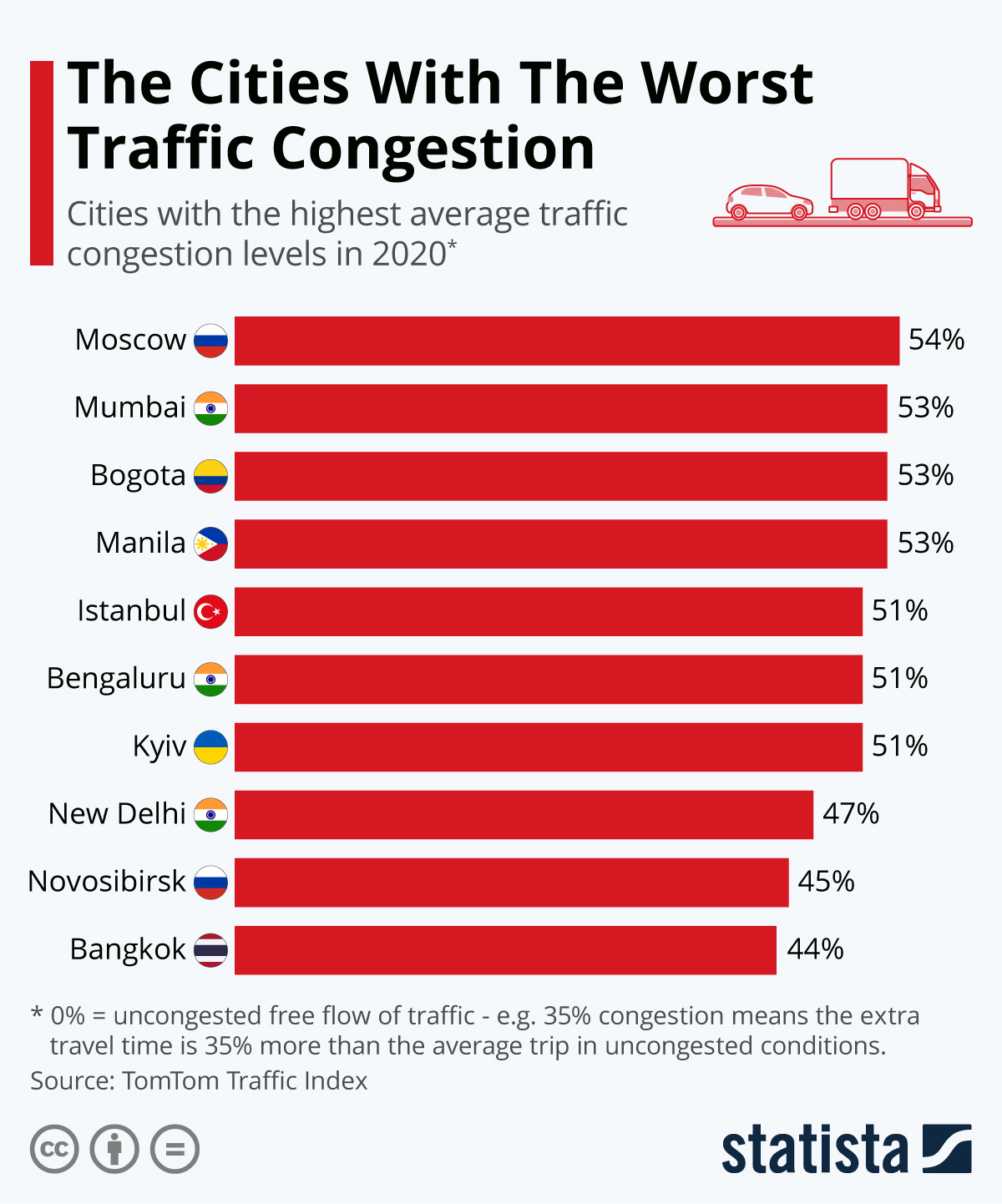
**OBJECTIVES:**

Traffic management: One objective could be to count the number of vehicles passing through a particular road or intersection to help manage traffic flow. This data can be used to optimize traffic signal timings, identify congested areas, and plan road infrastructure improvements.



Traffic management using vehicle counting in AI is a technique that involves using artificial intelligence (AI) to automatically detect and count vehicles on the road. The system works by analyzing images or videos captured by cameras installed at various locations, such as intersections or highways. By using AI algorithms, the system can accurately identify and count vehicles, including cars, trucks, and motorcycles. The data collected can then be used to monitor traffic flow, identify congested areas, and optimize traffic management strategies. One of the main advantages of using AI for traffic management is its ability to process large amounts of data in real-time. This enables traffic authorities to make quick decisions and take appropriate actions to mitigate traffic congestion and reduce accidents. Moreover, AI-based traffic management systems can also help to improve traffic safety by identifying potential hazards such as speeding vehicles or vehicles that are driving erratically. This information can then be relayed to law enforcement agencies, who can take appropriate action to prevent accidents. Overall, traffic management using vehicle counting in AI is an effective solution to manage traffic flow, improve safety, and reduce congestion on the roads.





**Solution:**

Intelligent vehicle detection and counting are becoming increasingly important in the field of highway management. However, due to the different sizes of vehicles, their detection remains a challenge that directly affects the accuracy of vehicle counts. To address this issue, this paper proposes a vision-based vehicle detection and counting system. A new high definition highway vehicle dataset with a total of 1200 days is published in this study. Compared with the existing public datasets, the proposed dataset contains number of vehicles, which provides the complete data foundation for vehicle detection based on deep learning. In the proposed vehicle detection and counting system, the highway road surface is retrieved into a frame. Several highway surveillance videos based on different scenes are used to verify the proposed methods. The experimental results verify that using the proposed segmentation method can provide higher detection accuracy, especially for the detection of small vehicle objects. Moreover, the novel strategy described in this article performs notably well in judging driving direction and counting vehicles. This paper has general practical significance for the management and control of highway scenes.

**Important libraries:**

Numpy: NumPy is a Python library used for scientific computing and data analysis. It provides a high-performance multidimensional array object, as well as tools for working with these arrays. The key feature of NumPy is its ability to perform fast mathematical operations on arrays and matrices.

cv2 : cv2, or OpenCV (Open Source Computer Vision Library), is a popular open-source computer vision library that provides a wide range of image processing and computer vision algorithms. It was originally developed by Intel in 1999 and has since been maintained by a community of developers.

Pandas: Pandas is a popular open-source data analysis and manipulation library for the Python programming language. It provides a fast and efficient way to work with structured data such as CSV files, Excel spreadsheets, SQL databases, and more.

Matplotlib: Matplotlib is a popular open-source data visualization library for the Python programming language. It provides a range of tools for creating static, animated, and interactive visualizations in Python.

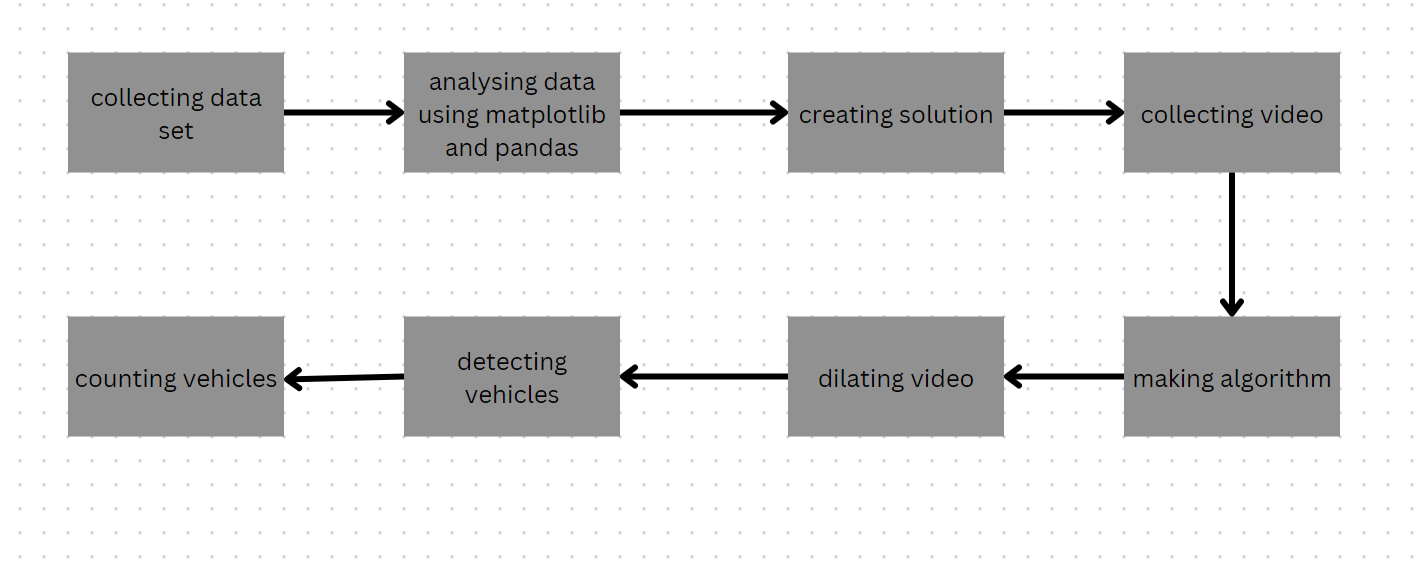
**Noteworthy contribution in the related domain:**

Naveen:analysing data set and gathering video

Anil:vehicle counting code

Vinay:vehicle counting code

**Proposed methodology:**



Steps:

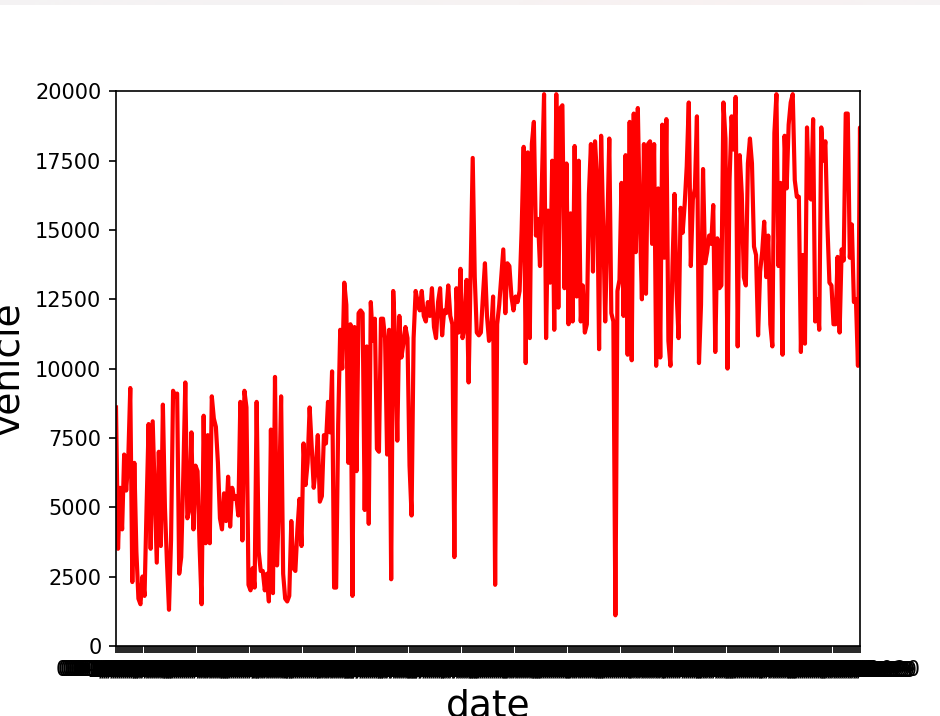
* Collect video data: The first step is we collected video data of the area where you want to count vehicles. This is done using surveillance cameras or other video sources.
* Pre-process the video data: we have collected the video data, we prepossessed it to prepare it for analysis. This may involve tasks such as converting the video to a common format, resizing the video, and filtering out noise.
* Detect vehicles: The next step is to detect vehicles in the video frames. This is done using object detection algorithms such as create background subtractor This algorithm can detect and classify objects in the video frames, including vehicles.
* Track vehicles: After detecting vehicles, we tracked them across video frames. This can be done using object tracking algorithms such as dilated filters. These algorithms can predict the position of the vehicle in the next frame and update the tracking accordingly.
* Count vehicles: Finally, we counted the number of vehicles by analysing the tracked objects over time. This may involve setting up virtual gates or lines in the video frames and counting the number of times a tracked object crosses the gate or line.
* Analyse the results: we have counted the vehicles, and analysed the results to gain insights into traffic patterns, vehicle flow, and other metrics of interest.

**Result analysis:**

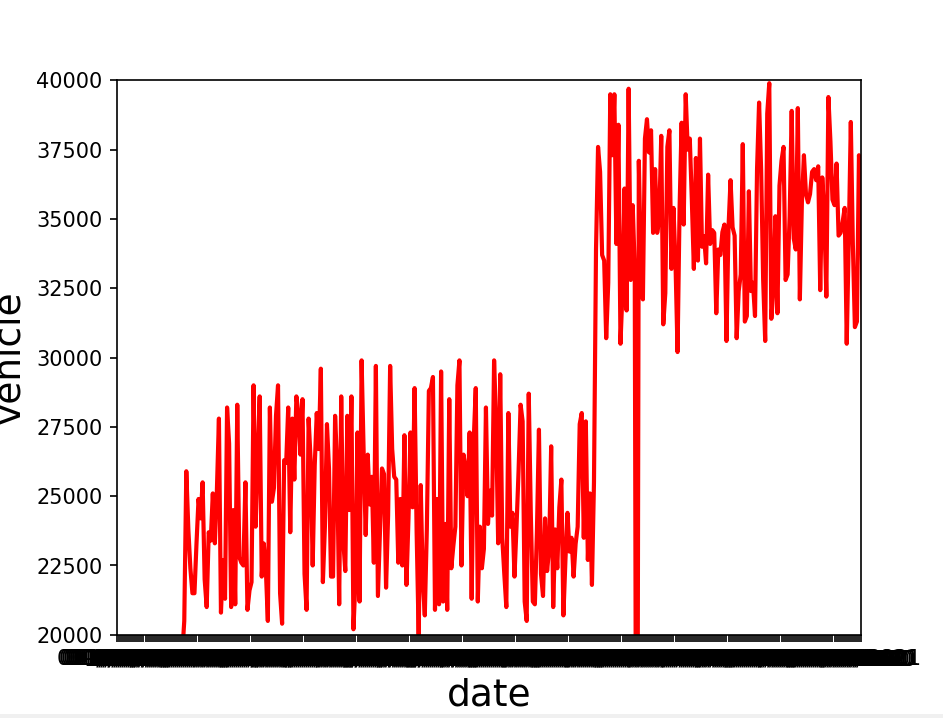
Analysing data:

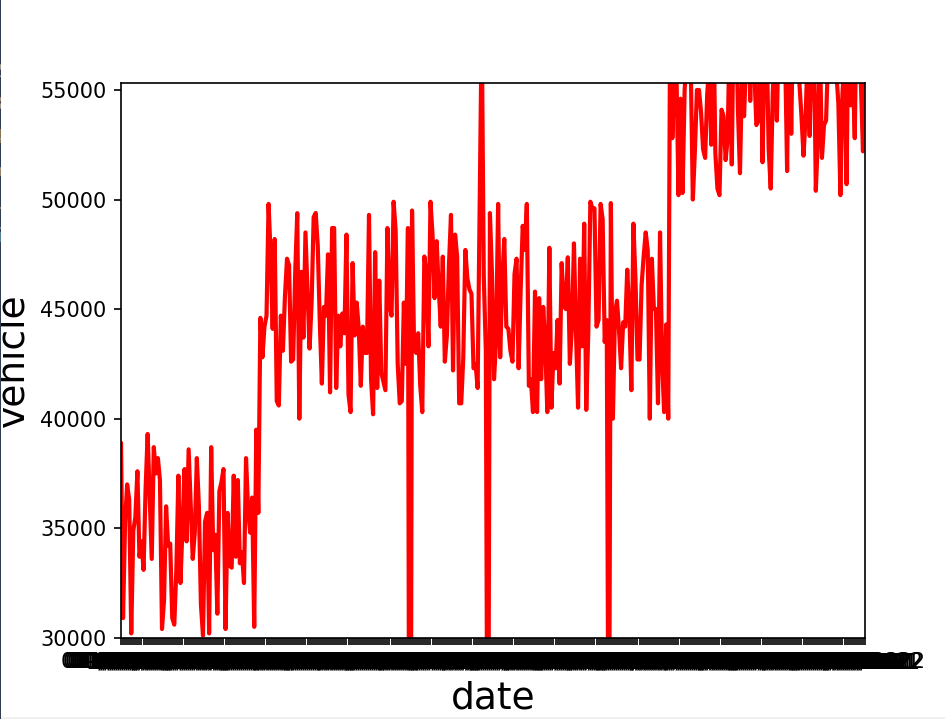
Year wise:

2019:

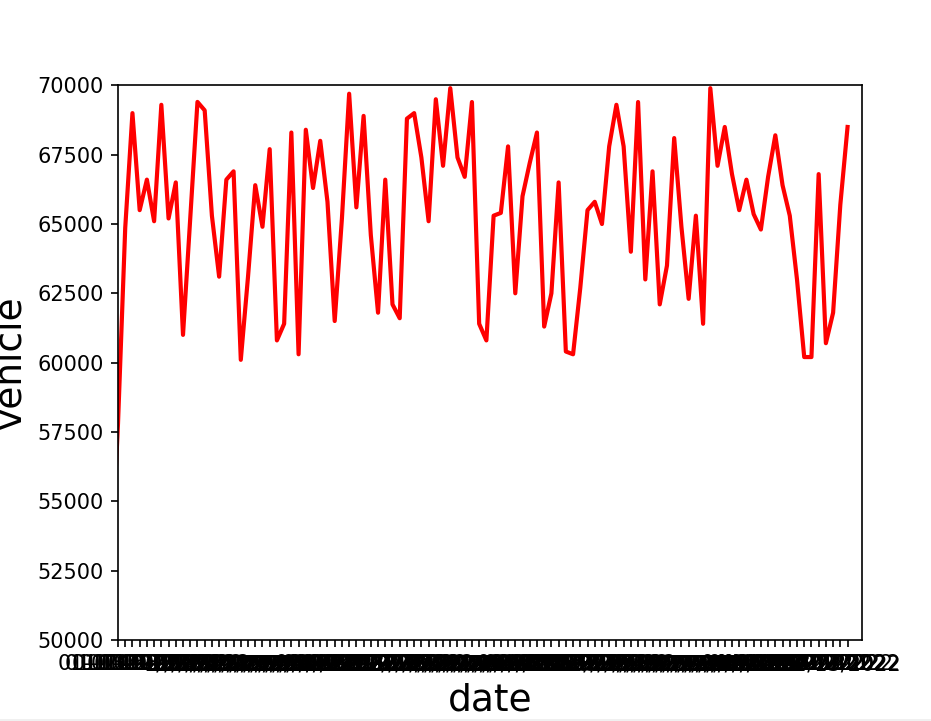


2020:

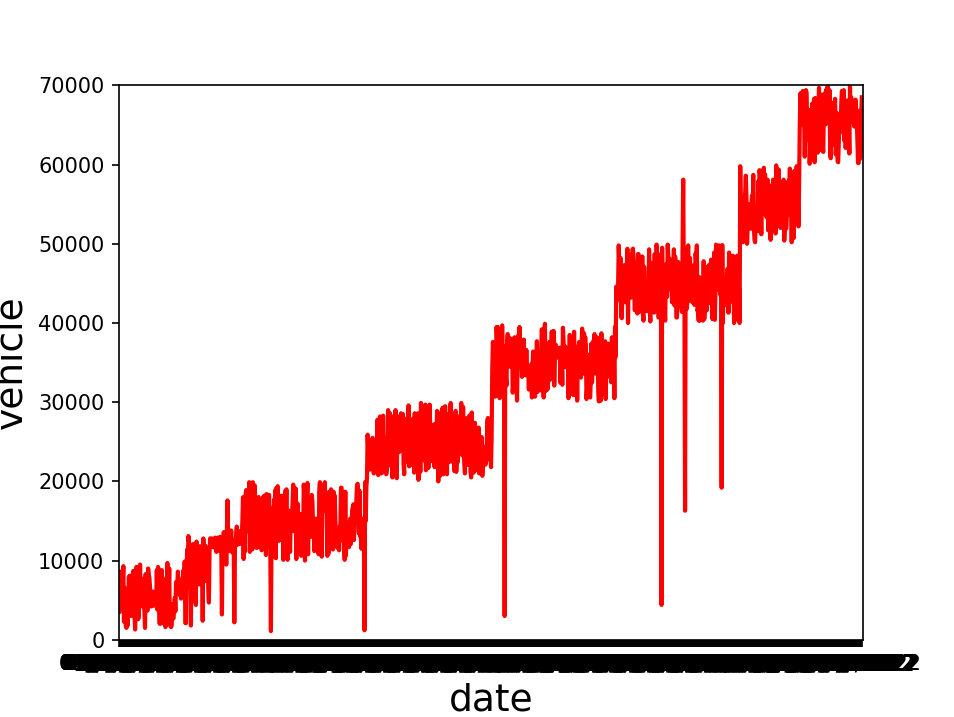


2021:

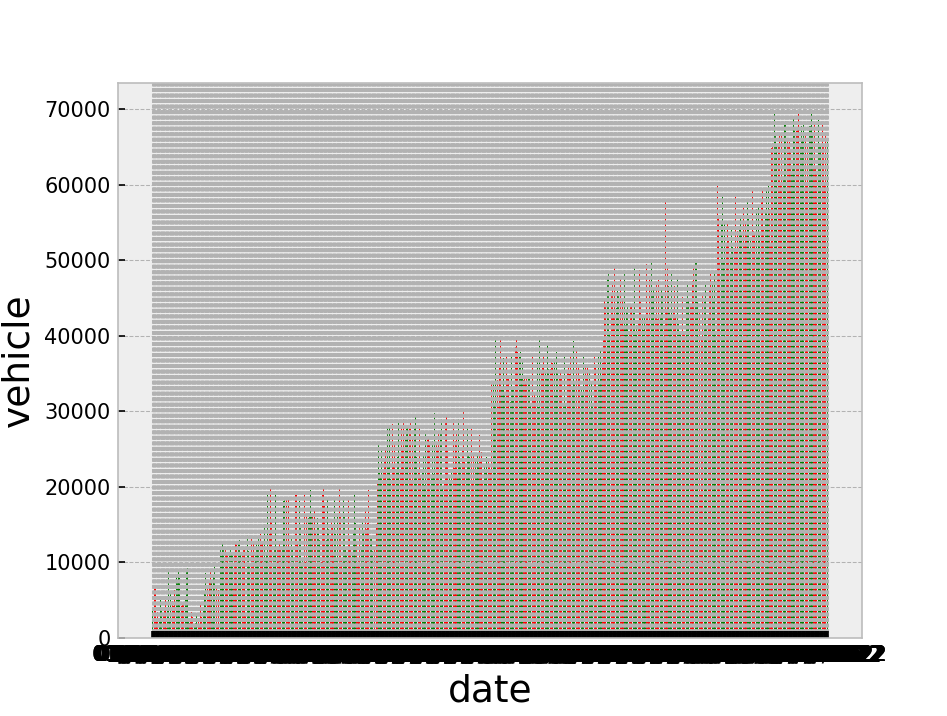
2022:



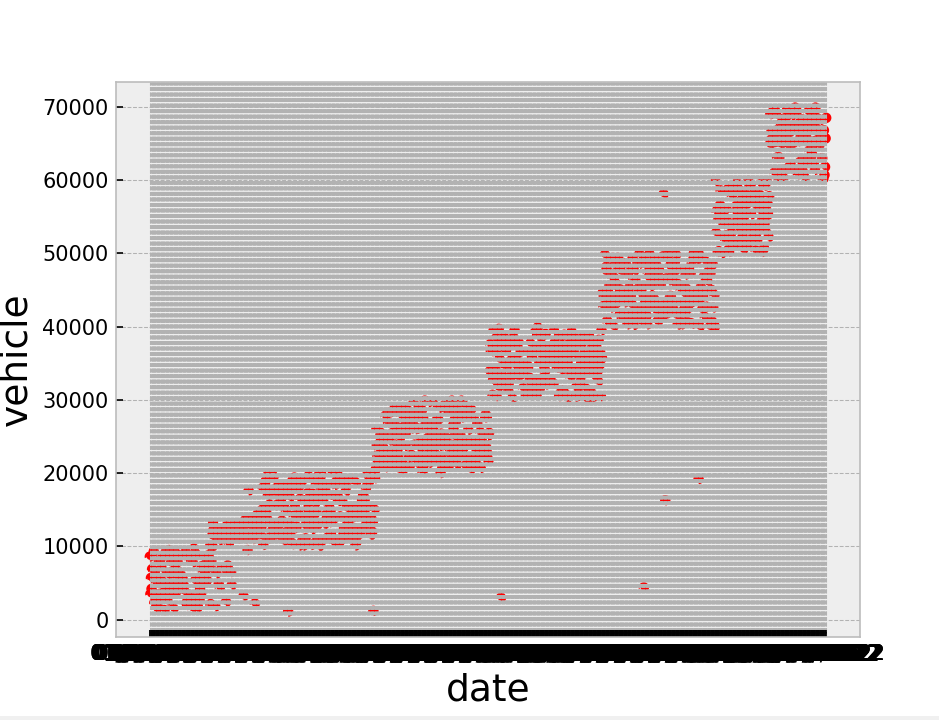
Overall:



Histogram:



Scatter:



**Outcome:**

Trend analysis: This increasing graph has trends in vehicle counts over time. By analyzing the slope and curvature of the graph, you can identify periods of high or low vehicle flow, and make predictions about future traffic patterns.

Seasonality analysis: This increasing graph has seasonal patterns in vehicle counts. For example, if the graph shows a consistent increase in vehicle counts during the summer months, this could indicate that the area experiences higher levels of tourism or recreational activity during this time.

Capacity analysis: By analysing the slope of this gradually increasing graph, we can gain insights into the capacity of the road or area being monitored. If the graph shows a steep increase in vehicle counts over time, this could indicate that the road or area is reaching its capacity and may require additional infrastructure or management.

**Vehicle counting:**

import cv2

import numpy as np

cap = cv2.VideoCapture('traffic video 3.mp4')

min\_width\_rect=40

min\_height\_rect=40

count\_line\_position=730

algo=cv2.bgsegm.createBackgroundSubtractorMOG()

def center\_handle(x,y,w,h):

    x1=int(w/2)

    y1=int(h/2)

    cx=x+x1

    cy=y+y1

    return cx,cy

detect=[]

delay=60

offset=4

counter=0

while True:

    ret,frame1=cap.read()

    grey=cv2.cvtColor(frame1, cv2.COLOR\_BGR2GRAY)

    blur=cv2.GaussianBlur(grey,(3,3),5)

    img\_sub=algo.apply(blur)

    dilate=cv2.dilate(img\_sub,np.ones((5,5)))

    kernel=cv2.getStructuringElement(cv2.MORPH\_ELLIPSE,(5,5))

    dilated=cv2.morphologyEx(dilate,cv2.MORPH\_CLOSE,kernel)

    counterShape,h=cv2.findContours(dilate,cv2.RETR\_TREE,cv2.CHAIN\_APPROX\_SIMPLE)

    cv2.line(frame1,(300,count\_line\_position),(1250,count\_line\_position),(0,0,250),2)

    for(i,c) in enumerate(counterShape):

        (x,y,w,h)=cv2.boundingRect(c)

        val\_counter=(w>=min\_width\_rect) and (h>=min\_height\_rect)

        if not val\_counter:

            continue

        cv2.rectangle(frame1,(x,y),(x+w,y+h),(0,255,0),2)

        center=center\_handle(x,y,w,h)

        detect.append(center)

        cv2.circle(frame1,center,4,(0,0,255),-1)

        for(x,y) in detect:

            if y<(count\_line\_position+offset) and y>(count\_line\_position-offset):

                counter+=1

            cv2.line(frame1,(300,count\_line\_position),(1250,count\_line\_position),(0,127,255),2)

            detect.remove((x,y))

            print("Vehicle number"+str(counter))

    cv2.putText(frame1,"VEHICLE NUMBER :"+str(counter),(450,70),cv2.FONT\_HERSHEY\_SIMPLEX,2,(0,0,255),2)

    cv2.imshow('Detector',dilated)

    cv2.imshow('video original', frame1)

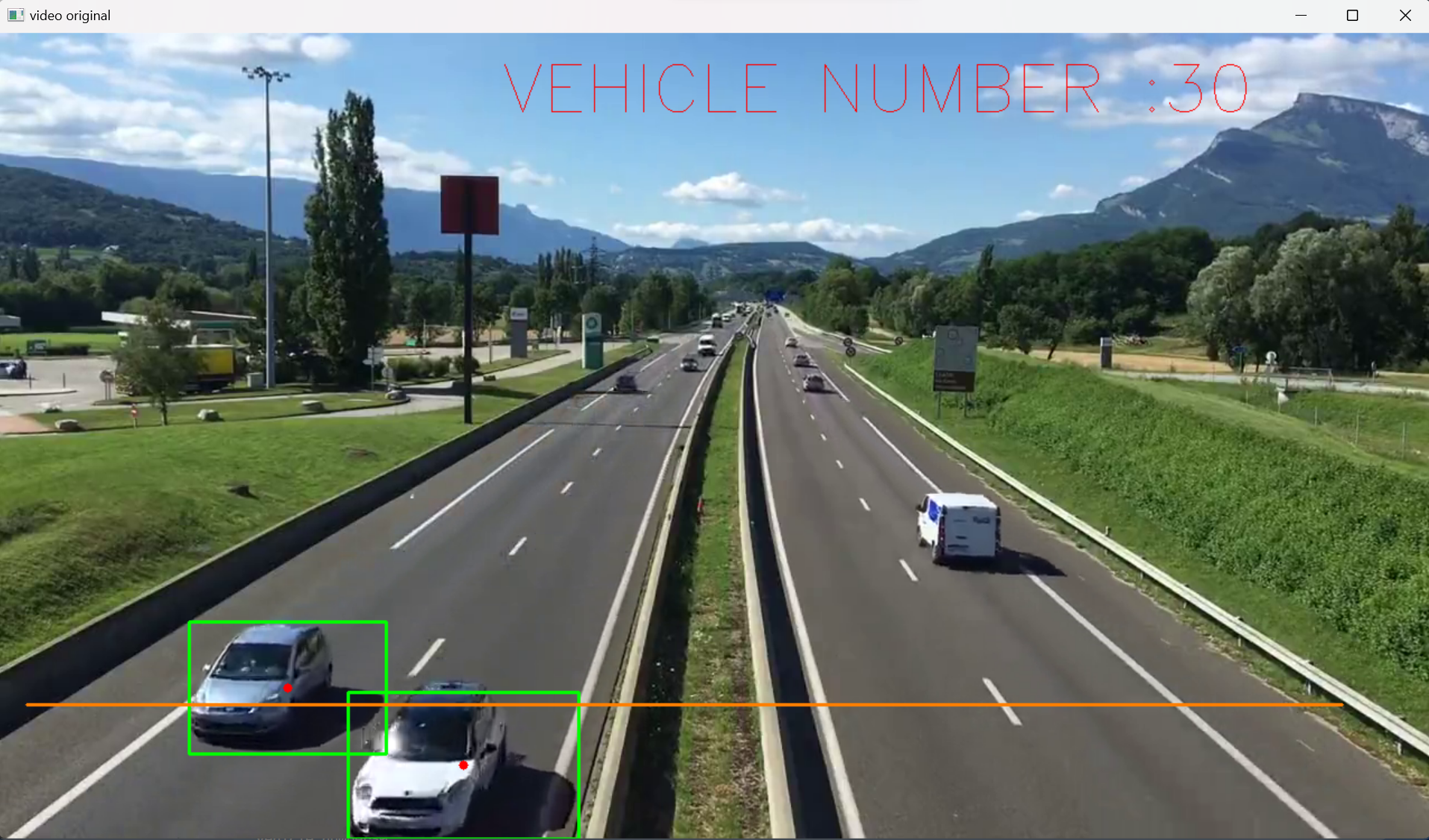
    k=cv2.waitKey(1)

    if k==ord('q'):

        break

cv2.destroyAllWindows()

cap.release()

****

**Detector:**

****

Functions used:

* createBackgroundSubtractorMOG: It is a function in OpenCV's Background Subtraction Library that can be used to subtract the background from a video sequence in real-time. It creates a Gaussian mixture model background subtractor which is used to segment out the foreground objects from the background
* Gaussian blur: It is a low-pass filter that can be used to reduce noise and blur the image, which can improve the accuracy of subsequent image processing tasks.
* Dilate: used to perform morphological dilation on an image. Dilation is a process that can be used to increase the size of objects in an image, by expanding their boundaries or thickening their edges
* Morphologyex: used to perform various morphological operations on an image. It is a general-purpose function that can be used to perform a range of operations, including dilation, erosion, opening, and closing.
* Findcontours: used to identify and extract the contours of objects in a binary image. A contour is a curve that connects points of equal intensity in an image, and can be used to represent the boundaries of objects or regions.

**Conclusion:**

vehicle counting using AI is a useful and practical application in the field of computer vision. By leveraging techniques such as image segmentation, object detection, and tracking, it is possible to accurately count the number of vehicles in a given area.OpenCV provides a rich set of tools and functions that can be used to build a vehicle counting system. For example, BackgroundSubtractorMOG and GaussianBlur can be used to pre-process the video frames and remove noise, while findContours and drawContours can be used to detect and draw the contours of vehicles.By combining these techniques, it is possible to build a robust and reliable vehicle counting system that can be deployed in a variety of settings, such as traffic monitoring, parking lot management, or security surveillance. The system can be further improved by using machine learning algorithms, such as deep neural networks, to enhance the accuracy and efficiency of the counting process.Overall, vehicle counting using AI has the potential to provide valuable insights and data for various applications, and can help to improve safety, efficiency, and productivity in a variety of settings.