Kingdom of Saudi Arabia Ministry of Education University of Jeddah College of Computer Science and Engineering Department of Computer Science and Artificial Intelligence		مالمل%ة عال)*+ة اسلعود/ة ز ارة التعل+موجامعة جدّة +ة علوم وحندسة الحاسب قسم علوم الحاسب والذ;اء الاصطنا را

Check Report (2024/2025)

CCCS323 Machine learning

Progress Check [Report Outline]

Instructor Name	Section
Dr.Hammam AL-Ghamdi	

Student Name
Jameel Rami Mahjub
Abdulrahman Sami Tibagi
Abdulaziz Hamad Nasrallah

Table of Contents

1. Introduction

The logistics domain plays a vital role in assessing the organization of infrastructure such as roads and facilities. As tourism continues to grow in Saudi Arabia, efficient logistics have become essential to support accessibility and enhance visitor experience.

Statistics:

Year	2021	2022	2023
Tourists	67.3 Million	94.9 Million	109.3 Million
Percentage of growth	-	378%	65%

Utilizing machine learning, specifically generative modelling, can significantly enhance our understanding of traffic congestion by identifying its temporal patterns and underlying causes of traffic. This approach will enable stakeholders to forecast potential traffic conditions at any given time accurately.

With the Advancement of the tourism and entertainment sector in the Kingdom of Saudi Arabia and the upcoming World Cup in 2034, we can expect an increase in the number of tourists. This will consequently lead to a rise in the number of cars and buses on the roads. Therefore, it will be essential to enhance the organization of traffic operations to accommodate these increased numbers.

To enhance the organization traffic operation to accommodate this increasing number we will use Machine learning and computer vision to solve this problem.

The expected result of this project is a future stating model that helps forecast traffic congestion trends with high accuracy. This will support the authorities in adaptation of traffic flows and resource allocation, especially during the extreme tourism period. By doing this, the solution improvement will be contributed to logistics infrastructure, better visitor experience and a more flexible transport system in the Kingdom.

2. Literature Review

Research Papers in Machine Learning:

1.Key research papers in data preprocessing and supervised learning

We have read 2 papers about preprocessing and supervised learning:

Subject	Detecting traffic incidents, address the issue of unbalanced data	Accident Severity Prediction under Unbalanced Data
Author	Dr.P.Rajesh Kanna, Dr.S.Vanithamani, Dr.P.Karunakaran, Dr.P.Pandiaraja, N.Tamilarasi, P.Nithin	Jiaxin Lu, Zhejun Huang, Lili Yang
Year	**2024**	**2023**
Aim of the project	To identify traffic incidents and deal with unbalanced data	Identify the accident severity and deal with unbalanced data elemenating zero criticality feature
Evaluate	the paper focuses on incident detection using WRF and factor analysis for Dimensionality reduction and SMOTE for balancing there data their Work is good but focusing in a specific part of road analysis lead them to instable data skewed and imbalanced , their choice of techniques and models have led them into a good accuracy of 92% which is good with this flawed data	They tried to prepare the data for classification with eliminating less wanted features using the XGBoost algorithm this helped generating good random samples to balance the data and help over sampling on the critical accidents and undersampling the less criticl ones this make identification easer this was preformed by SMOTE-NC from a prespictve of preparing the data to be procced and reduced the miss classification
Dataset	range of incident types, traffic density, and environmental variables	Traffic incident dataset, sourced from API broadcasts cameras
Models used	Weighted Random Forest (WRF) To make the model learn from both majority and minority of the classes the highest record KNN but it was the least accurate	XGBoost (Parameter elmenating unecceary features) SMOTE-NC Oversampling Undersampling
review	The paper provided a very good solutions to process the data and detect the incidents in the road using the Random Forest and dimensionality reduction with factor analysis and SMOTE for balancing the data	The paper showed another way of dealing with misleading features or less needed but generating the data after cleaning it was a good idea to help generating data with wanted features that could help identifying true critical cases
Advancements after the paper	The paper provided us with very two good ways of dealing with our data which was so relevant the RF model gave us a very good results after testing also the SMOTE for balancing the data but the paper	Our advancement was the SMOTE-NC which helped us deal with our unbalanced data this case happens when dealing with traffic data cause most of the time the traffic is normal and it can lead to

	didn’t but that much of effort to explain it but in our case we used SMOTE-NC instead based on other researchs	overfitting if we didn’t balance the data
Research Question	The gap that we solved that this paper was using a dataset that was a collected data from sensors and cameras and the focus was on road anomaly focusing on just on phenome on the road will leave out behind such an important information to help solving road problems	They deald with less wanted features but they didn’t mention the diementionality reduction because traffic data uses a lot of important features vehicle types time date etc…
Gap	How to identify majority class and how to identify a minority class? weights are added to the algorithm. In each instance, these weights make up for the data imbalance . By giving instances of the minimal value class more weights, WRF ensure that they have impact on the decision. The model can effectively learn from majority and minority	How did they deal with the unwanted classes? By Randomly excluding the unwanted class samples

1.Key research papers in data preprocessing and supervised learning

We have read 2 papers about preprocessing and supervised learning:

Generating Realistic Synthetic Traffic Data using Conditional Tabular Generative Adversarial Networks for Intelligent Transportation Systems.	**Smart Traffic Congestion Reduction System Using IoT and Machine Learning**
Archana Nigam and Sanjay Srivastava	A.Lakshna, K.Ranesh, B. Prabha, D.Sneema and K.Vijayakumar
2023	2021
To address data sparsity in ITS by generating realistic synthetic traffic data	To reduce urban traffic congestion by deploying IoT sensors on streetlight poles to collect signals (WiFi, Bluetooth) from vehicles' MAC addresses, then using ML algorithms (Logistic Regression, Random Forest, AdaBoost) for traffic prediction and route optimization.
PeMS : 5-minute interval data NYC Yellow Taxi: 1.5B+ trip records (pickup/drop-off, fare, distance)	Kaggle-sourced traffic data
Aim to improve data generation in ITS field	The future work is to improve the security level while storing the data in a cloud platform the chance of data breaches is high, so cryptography is to be used for end-to-end encryption.
How can (CTGANs) be useful to generate realistic synthetic traffic data to address the challenge of data sparsity in Intelligent Transportation Systems (ITS)? CTGANs incorporate auxiliary variables (e.g., time, location) to preserve relationships between traffic features (speed, flow, occupancy).	**How can smart traffic systems reduce congestion in urban areas?** Sensors mounted on roadside poles captured signal data, using the MAC addresses of passing devices to estimate vehicle counts. For real-time traffic prediction, Logistic Regression, Random Forest, and AdaBoost algorithms were evaluated.
the paper use CTGAN to generate accurate data but in our case we cant do that for the following problem **It needed a lot of data (which we didn't have).** **It had low accuracy without enough training samples.** **We were missing the speed feature, which is important for traffic modeling**	The paper use Logistic Regression (LR) and had 91% accuracy But when me use it in our project it has a low accuracy because the data is overlapping and Sensitive to Irrelevant or Correlated Features and not Robust to Outliers
To fix this, we used SMOTE-NC. SMOTE -NC gave me better synthetic data quality without CTGAN's limitations. It's simpler, more efficient, and works well with incomplete real-world traffic data.	In our project we used random forest (RF) because it is Uses multiple decision trees to capture non-linear interactions, Averages predictions across many trees, which together improve its Robust to Outliers and Noisy Data.

1. Key Research Papers in Deep learning:

We have read 4 papers in counting vehicles that is :

	[3]Bi-Directional Dense Traffic Counting Based on Spatio-Temporal Counting Feature and Counting-LSTM Network	[4] REAL-TIME VEHICLE COUNTING BY DEEP-LEARNING NETWORKS	[5] Vehicle Counting: Survey and Experiments	[6] Vehicle Counting based on Convolution Neural Network
Author	Shuang Li , Faliang Chang , and Chunsheng Liu	CHUN-MING TSAI1, FRANK Y. SHIH , JUN-WEI HSIEH	Hoang-Phong La, Minh-Thao Ha, Hai-Long Nguyen, Manh-Thien Nguyen	Jenna Maria Anil, Liz Mathews, Rajeswari Renji, Riya Mariya Jose
Year	2021	2022	2020	2023
Paper Use	Previous line of interest (LOI) counting methods rarely focus on dense scenarios and their performance largely relies on the accuracy of tracking. Avoiding the use of complex tracking methods, an LOI counting framework is proposed to address the bi-directional LOI counting problem in dense scenarios. For detection use (YOLOv3) without relying on a multi-target tracking process for tracking and counting each vehicle, a counting network is proposed, called the counting Long Short-Term Memory (cLSTM) network , to do analysis of the bi-directional STCF features and vehicle counting in successive video frames. an estimation model is designed for estimating traffic flow parameters including speed, volume and density, use ROI	paper used three YOLOv3 and two YOLOv2to fine tune our vehicle detectors to detect vehicle in thefHsuehshan Tunnel. In order to alleviate the traffic flow in the HST, the General Administration of Highway has set up some cameras in the HST. The driving control center staff monitors these cameras with their eyes. When the traffic volume is high, the	evaluate the viability of using Deep Learning pre-trained models include Faster R-CNN, SSD, YOLOv3 for detection-based.	YOLOv3 model is used for object detection and classification Vehicle counting; Virtual detection zone; Computer vision; YOLO; SORT; DeepSORT; The region of interest (ROI) is first identified in the work of Gabriel Oltean For object identification and counting, Thanh-Nghi Doan and Minh-Tuyen Truong [10] suggested a method that com bines YOLO with DeepSORT. YOLO is used to identify and classify objects. Tracking and counting objects is done using DeepSORT.
Dataset	UA-DETRAC dataset and the captured videos	PASCAL VOC datasets	AI CITY CHALLENGE and Vehicles Nepal dataset	UA-DETRAC dataset
FUTURE WORK or Deficiencies in work	aim to improve this LOI model to adapt to more complex scenarios, including extreme weather, dirt, and heavy congestion.	In the future, more vehicles will be collected and trained. We will also try to use the YOLOv4, YOLOR, YOLOv5, and YOLOX to train to obtain the best vehicle detection results.	In the future, we will apply the new method to our models to improve accuracy. Moreover, we will develop a traffic management system base on our experiment.	During vehicle tracking, the SORT algorithm faces several drawbacks, including multiple vehicle counting. So, in the future, to overcome these constraints, the SORT algorithm can be replaced with its extended version, DeepSORT (Deep Learning-based SORT).

Question and answer the paper	How can count bi-directional traffic flow ? They use (LOI) counting methods that are rarely focus on dense scenarios and their performance largely relies on the accuracy of tracking. without relying on a multi-target tracking process for tracking and counting each vehicle, a counting network is proposed called (cLSTM), to do analysis of the bi-directional STCF features and vehicle counting in successive video frames.	How to improve the driving safery and reduce traffic congestion during holidays and work hours ? a lane-based vehicle counting system using deeplearning networks Our method includes YOLO vehicle detection and lanebased vehicle counting.	How can get information about traffic to control the flow of transports ? The methods they use in this paper are detection-based counting, regression-based counting. they also evaluate the viability of using Deep Learning pre-trained models include Faster R-CNN, SSD, YOLO for detection-based.	How can define a good strategy to count vehicle for a traffic control to be successful, accurate and thorough traffic flow information is essential ? the zone is made by setting the coordinates in the frame. This can be done by manually plotting the points on the frame. The zone is visualised into the frame using the OpenCV library. Pre-trained YOLOv3 model is used for object detection and classification. Sort algorithm is used for vehicle tracking and counting the number of vehicles that pass through the virtual detection zone. The number of vehicles passing through the virtual detection zone in a given time can be used to estimate the traffic volume at the time.
Analysis the point of relevant and what we solve	The paper Traffic Counting Based on Spatio-Temporal Counting Feature and Counting-LSTM Network tracking-free method using Spatio-Temporal Counting Features (STCF) and an LSTM network to count vehicles in dense traffic. While effective, this approach has two key limitations: (1) the STCF feature extraction and cLSTM processing introduce computational overhead, and (2) it lacks granular outputs (that has only vehicle) and we use many class that can have different effect to traffic flow.	paper uses lane division (left/right) and temporal heuristics to track and count vehicles in a two-lane tunnel. This method relies on manually defined lane boundaries and calculates inter-vehicle time intervals to avoid duplicate counts, which may fail in dense or multi-lane scenarios. In contrast, our approach leverages DeepSORT and a Virtual Detection Zone (VDZ) to automate tracking across four lanes without manual lane partitioning. DeepSORT's association metric mitigates duplicate counts by assigning persistent IDs to vehicles, while the VDZ ensures accurate crossings regardless of lane geometry. This eliminates the need for heuristic time-based checks and scales seamlessly to complex roadways.	paper manually splits frames into a 20%/80% ROI (Region of Interest), where the top 20% handles distant/small vehicles using computationally expensive Super Resolution APIs. This rigid division fails to adapt to dynamic traffic conditions. Our solution replaces manual ROI splitting with a Virtual Detection Zone (VDZ) that automatically adjusts to traffic density. By strategically placing the VDZ where vehicles are optimally detectable (e.g., mid-frame for clarity), we eliminate the need for super-resolution preprocessing. Vehicles are counted only upon crossing the VDZ line, ensuring accuracy regardless of size or position. This approach reduces computational overhead and simplifies deployment in variable traffic scenarios	The Vehicle Counting based on Convolution Neural Network paper employs the SORT algorithm for vehicle tracking, which suffers from critical drawbacks such as frequent identity switches and duplicate counting in dense traffic due to its reliance on short-term motion cues. Our solution addresses these limitations by implementing DeepSORT, which enhances tracking robustness through a deep association metric. By integrating appearance descriptors with motion information, DeepSORT maintains consistent vehicle IDs across occlusions and minimizes counting errors—even in complex multi-lane scenarios. This upgrade eliminates SORT's dependency on heuristic motion models, ensuring accurate, real-time counts without manual intervention.

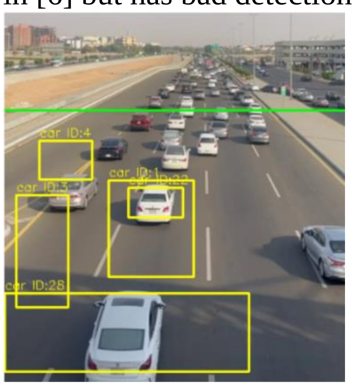
Technical Advancements and Implementation:
The reviewed papers primarily utilize YOLOv3 or YOLOv2 for vehicle detection, which suffer from limitations such as lower accuracy in small-object detection, higher false-negative rates in dense traffic, and slower inference speeds compared to modern architectures. To address these shortcomings, this project adopts YOLOv8, which offers significant improvements (Higher Accuracy, Small-Object Detection, Robustness in Dense Scenes).
The main challenge with Line of Interest (LOI) methods [3]is maintaining the relationship between consecutive frames, especially under dense traffic conditions, which often requires complex models like Spatio-Temporal Context Fusion (STCF) and ConvLSTM (cLSTM)[3] Heavier (slower than pure detection + tracking). However, in this project, we simplify the process by using DeepSORT [6] for tracking and VDZ[6] for counting, which are less complicated and more efficient. Additionally, since we are not concerned with bi-directional roads, the system design becomes even more straightforward. All vehicle counts (categorized by type: car, bus, truck, motorcycle) along with timestamps and lane information are automatically logged to structured CSV files, enabling direct integration with traffic analysis pipelines. The CSV output includes fields timestamp, vehicle_class, count, average_time_exit_frame, and traffic_density_category (low/medium/high/heavy), facilitating both real-time monitoring and long-term pattern analysis without the computational overhead of bi-directional processing models.

One of the main advantages of our project is the development of an automated data collection and organization pipelines. The system captures sequential frames from a video feed, arranges them in temporal order, and stores the corresponding vehicle detection results in a structured CSV format. CSV file allows easy access and review by both humans and machines.

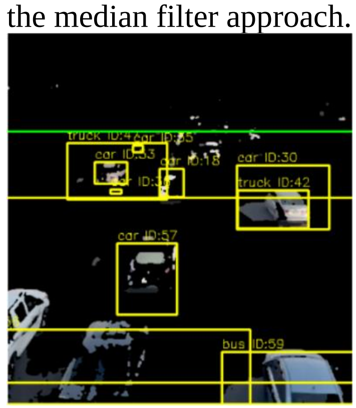
The papers focuses on counting only count. Our project adds:

- CSV logging (vehicle type, time, location).
- Traffic state classification (RF/GMM for "heavy/high/normal/low" labels).

we tried using GMM technique that in [6] but has bad detection



also tried using GMM technique and smoothing using the median filter approach. in [6] but has bad detection



3. State-of-the-art

3.1 Relevant Models

Kaggle : Traffic Prediction Random Forest

by: Hariharan

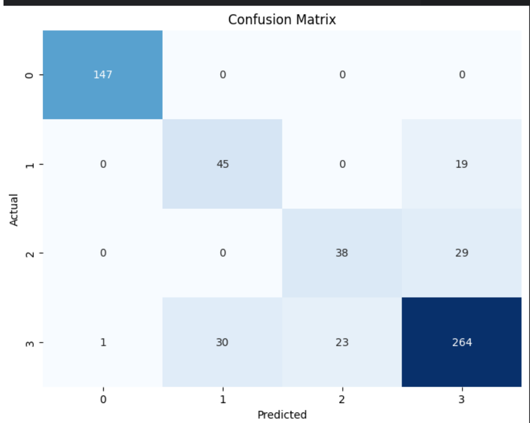
Traffic Prediction Dataset

he use random forest with Traffic Prediction Dataset this is the result :

	Time	Date	Total	Traffic Situation
0	0	10	39	low
1	900	10	55	low
2	1800	10	55	low
3	2700	10	58	low
4	3600	10	94	normal

```
rfc = RandomForestClassifier(max_depth= None,
es_split= 2, n_estimators= 10)
rfc.fit(X_train,y_train)
rfc.score(X_test,y_test)
```

```
0.8288590604026845
```

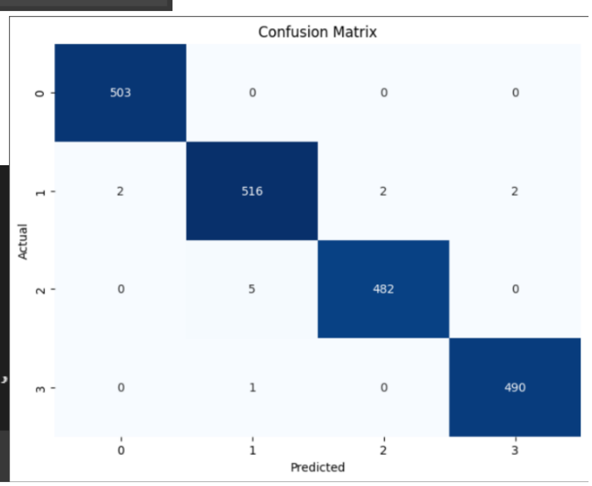


This our result :

	Time	Date	Day of the week	CarCount	BikeCount	BusCount	TruckCount	Total	Traffic Situation
0	12.00	10	2.0	31	0	4	4	39	1
1	12.15	10	2.0	49	0	3	3	55	1
2	12.30	10	2.0	46	0	3	6	55	1
3	24.45	10	2.0	51	0	2	5	58	1
4	13.00	10	2.0	57	6	15	16	94	2

```
# Train the Random Forest model
model = RandomForestClassifier(n_estimators=4, random_state=42)
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

```
Accuracy: 0.9940089865202196
```



3.2 Applicability to the Project

They used Random Forest to classify the dataset:

- the model predicts base on each sample of trees vote that this point of data is belong to a certain class and the class that's most voted will be the data point predicted class each tree is created based on bagging where all the features of the data point are grouped randomly creating multiple tree and based on the features that the tree have the tree votes for a class that it should belong to the most voted class we be the one.
- Any data preprocessing are not required for the random forest, deals with nonlinear problems which is needed in our case.
  On the other hand, it can be easily overfitted because it has no clue where to stop so it will create complex decision rules therefore most of the data points will be classified into the majority class

4. Dataset

4.1 Dataset Description

Recording Dataset:

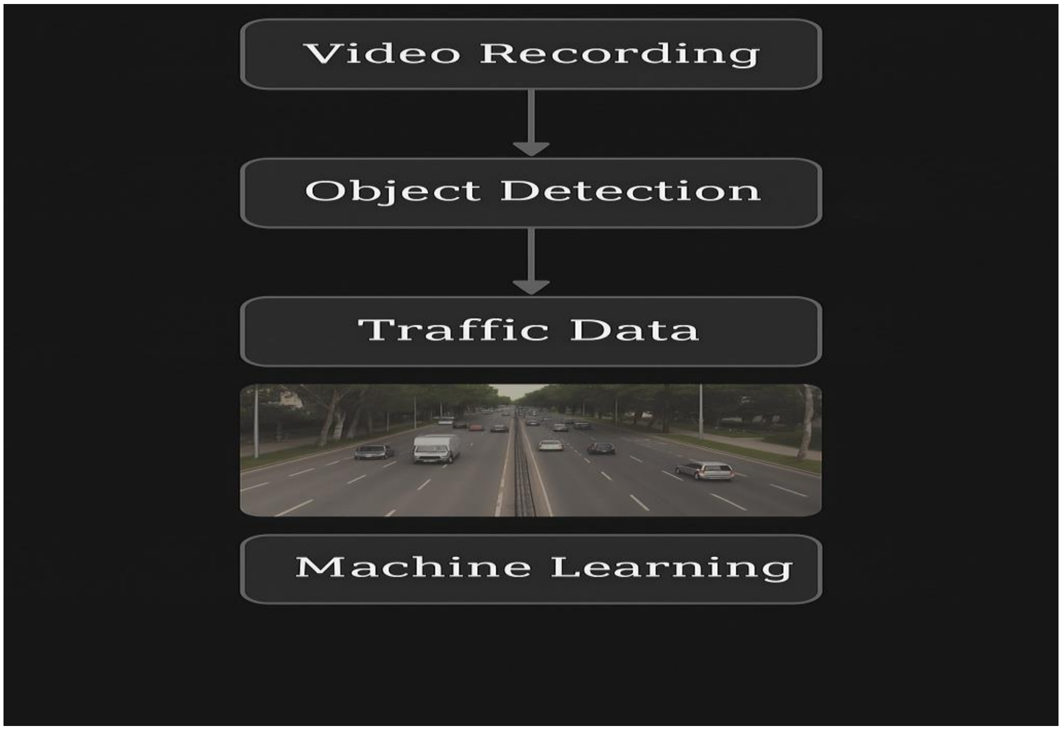
The data were collected through our recordings. The recorded videos were processed to extract detailed information, including the video's timestamp, the date of recording, the corresponding day of the week, and vehicle counts, categorizing them into cars, motorcycles, buses, and trucks. The average time each car remained within the frame was also calculated. Furthermore, the road type was classified as either a main road or a sub-road, and the overall traffic situation was assessed based on the collected data. This comprehensive approach enabled a thorough examination of traffic dynamics under varying conditions.

	time	date	day_of_week	car_count	motorcycle_count	truck_count	bus_count	total	avg_exit_time_seconds	road_status	traffic_situation
0	20:25:00	2025-04-20	Sunday	140	1	10	0	151	4.693563	m	high
1	20:26:00	2025-04-20	Sunday	74	1	7	2	84	5.676193	m	normal
2	20:27:00	2025-04-20	Sunday	69	0	2	1	72	4.196672	m	normal
3	20:28:00	2025-04-20	Sunday	75	0	4	1	80	4.595842	m	normal
4	20:29:00	2025-04-20	Sunday	133	0	5	0	138	4.605158	m	high
5	20:29:20	2025-04-20	Sunday	50	0	3	1	54	4.402627	m	high
6	20:26:00	2025-04-20	Sunday	37	2	0	0	39	6.219105	s	heavy
7	20:27:00	2025-04-20	Sunday	40	1	0	3	44	8.827520	s	heavy
8	20:28:00	2025-04-20	Sunday	34	2	0	1	37	12.453062	s	heavy
9	20:29:00	2025-04-20	Sunday	39	0	0	0	39	12.283955	s	heavy
10	20:29:44	2025-04-20	Sunday	36	1	1	0	38	9.293660	s	heavy
11	16:47:00	2025-04-21	Monday	145	1	11	1	158	4.539125	m	high
12	16:48:00	2025-04-21	Monday	103	2	11	2	118	4.286722	m	normal
13	16:49:00	2025-04-21	Monday	84	0	12	2	98	4.320115	m	normal

Kaggle dataset:

The dataset contains information collected by a computer vision model. The model detects four classes of vehicles: cars, bikes, buses, and trucks. The dataset is stored in a CSV file and includes additional columns such as time in hours, date, days of the week, and counts for each vehicle type (CarCount, BikeCount, BusCount, TruckCount). The "Total" column represents the total count of all vehicle types detected within a 15-minute duration.

	Time	Date	Day of the week	CarCount	BikeCount	BusCount	TruckCount	Total	Traffic Situation
0	12.00	10	2.0	31	0	4	4	39	1
1	12.15	10	2.0	49	0	3	3	55	1
2	12.30	10	2.0	46	0	3	6	55	1
3	24.45	10	2.0	51	0	2	5	58	1
4	13.00	10	2.0	57	6	15	16	94	2



4.2 Dataset Relevance

It is highly suitable for the purpose of our project to analyze and predict dataset traffic behavior. Unlike generic or simulated dataset, it refers to specific real world traffic conditions for a custom-made dataset targeted location, which improves the reliability of trained predictions and models on it.

The relevance of the dataset reinforces its capacity:

- Capture time-based and vehicle-type-based traffic patterns.
- Provide raw data for both descriptive analysis and future modeling.
- Support classification, clustering and forecast works using machine learning.

Facing challenges:
Manual Collection Setup: Installing a camera in a legal, safe and stable position requires planning and permission.

Environmental Factors: Light conditions, weather, and occlesions (eg, trees or poles) can affect the quality of detection.

Preprocessing Time: Important computational resources are required to extract structured data from video frames.

Model calibration: Vehicle detection model must be fine to reduce false positivity or missed detection.

Despite these challenges, the dataset provides a rich, flexible and realistic source of traffic data. This enables more accurate insight and helps develop scalable solutions for traffic forecasting and mob management

5. Replication of SOA

Kaggle

Traffic Prediction Random Forest

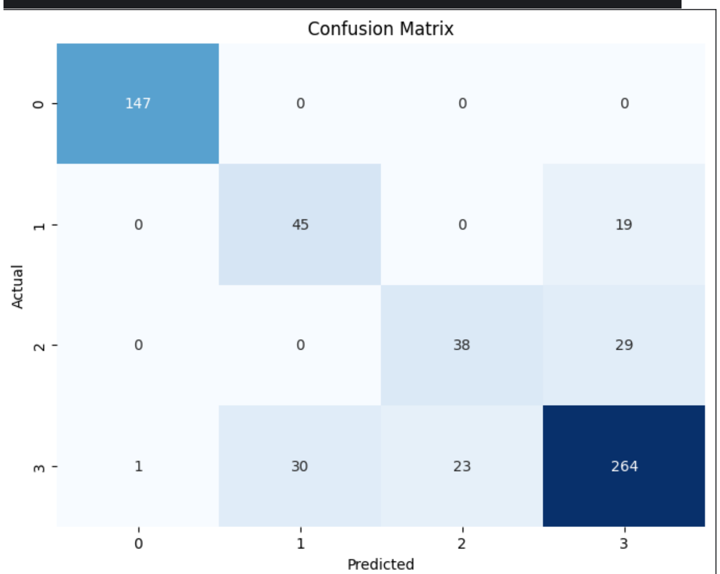
by: Hariharan

Traffic Prediction Dataset

he use random forest with Traffic Prediction Dataset this is the result :

```
rfc = RandomForestClassifier(max_depth= None,
es_split= 2, n_estimators= 10)
rfc.fit(X_train,y_train)
rfc.score(X_test,y_test)

0.8288590604026845
```

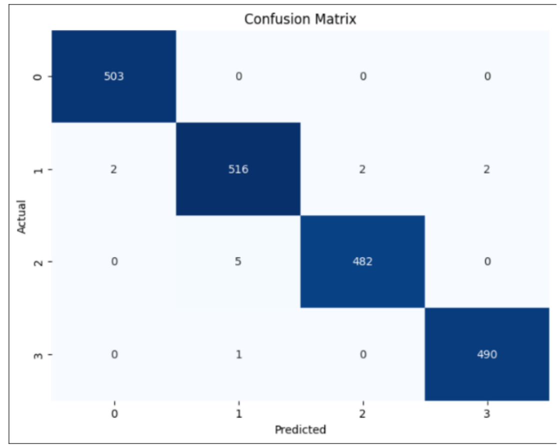
	Time	Date	Total	Traffic Situation
0	0	10	39	low
1	900	10	55	low
2	1800	10	55	low
3	2700	10	58	low
4	3600	10	94	normal



This is our code :

	Time	Date	Day of the week	CarCount	BikeCount	BusCount	TruckCount	Total	Traffic Situation
0	12.00	10	2.0	31	0	4	4	39	1
1	12.15	10	2.0	49	0	3	3	55	1
2	12.30	10	2.0	46	0	3	6	55	1
3	24.45	10	2.0	51	0	2	5	58	1
4	13.00	10	2.0	57	6	15	16	94	2

```
# Train the Random Forest model
model = RandomForestClassifier(n_estimators=4, random_state=42)
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))

Accuracy: 0.9940089865202196
```



The primary difference between his code and our code lies in the dimensionality reduction strategy. While his approach reduces the feature set from eight dimensions to just three—focusing only on Time, Date, and Total—our method retains all relevant features, ensuring that no important variables are excluded. By preserving the complete set of meaningful attributes, our model can capture more nuanced patterns and relationships in the data, leading to better accuracy and robustness. His simplification, though potentially improving computational efficiency, may sacrifice critical insights by omitting key predictors that contribute to the model's performance. Thus, our approach prioritizes comprehensive feature utilization to enhance predictive power and reliability

6. Supervised and unsupervised Models

6.1 Proposal of Machine Model Selections

1. Supervised Learning: Random Forest
   this model considers each data point as a result of a gaussian model then calculates the likelihood of the data point to decide it belongs to which cluster.
   This process can help us over come the overlapping data that we have due to multiple features

2. Unsupervised Learning: Gaussian Mixture Model (GMM)
   the model has something incommon with the gaussian mixture model that it predict is base on the maximum likelihood of each sample of trees vote that this point of data is belong to a certain class and the class that's most voted will be the data point predicted classes .
   It helps us decide the new data point class acutely with the help of GMM clustering

3. Deep Learning Yolov8 m:

   YOLO v8(medium) is a deep Learning model that uses labeled video/image data, then compresses the data for the model to use, and then goes through 4 stages:

   feature extraction - feature aggregation - prediction - post processing

   Using the YOLO model was necessary for us to be able to extract the features that we need from real data that's related to us and be able to analyze it

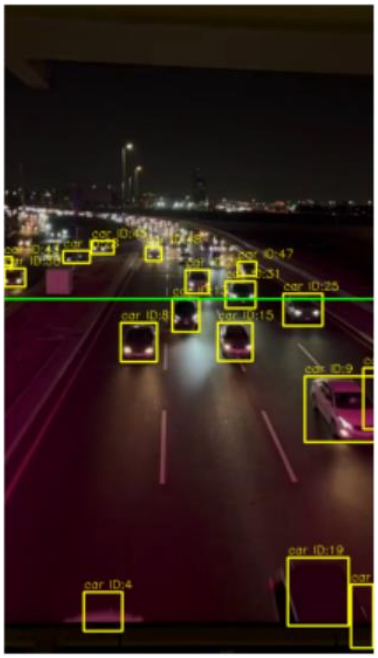
4. How do they work together:

5. Step 1: YOLO Extract features from videos dataset put them into CSV file then pass it to GMM

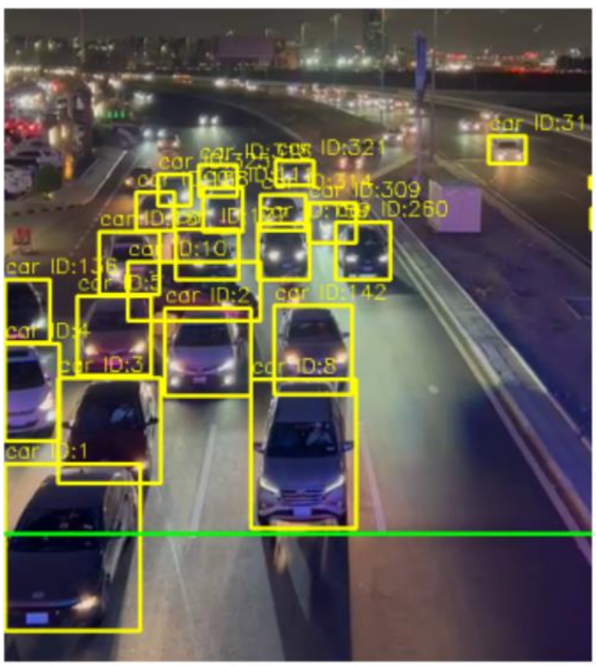
Step 1: GMM analyses raw traffic data to label the new pattern. Step 2:
RF uses these labels to improve the congestion of the crowd.
detecting the contradiction of GMM has the power of prediction of RF, which creates an adaptive system and they both use maximum likelihood-voting

6.2 Initial Results and Expectations

Before (ROI) After using (ROI)



Here we have small problem that is most of the cars are stopped due to heavy traffic that , so the number of cars will be few ,we used the average time a car was there before it lift from frame .



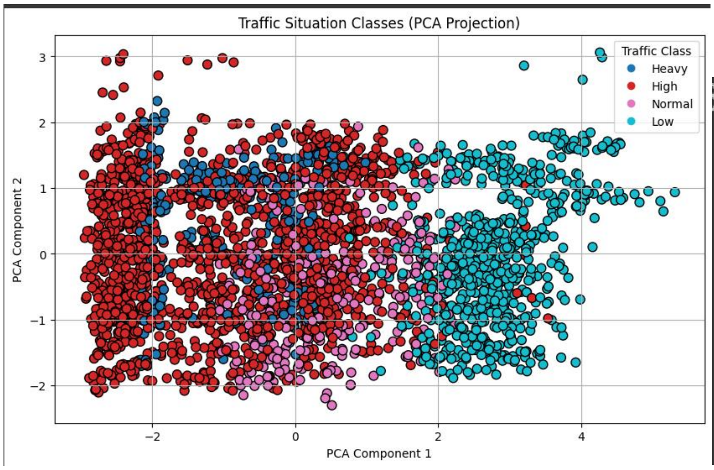
CSV file :

	A
1	time,date,day_of_week,car_count,motorcycle_count,truck_count,bus_count,total,avg_exit_time_seconds,road_status,traffic_situation
2	20:25:00,2025-04-20,Sunday,140,1,10,0,151,4.693563431924883,m,high
3	20:26:00,2025-04-20,Sunday,74,1,7,2,84,5.676192825503356,m,normal
4	20:27:00,2025-04-20,Sunday,69,0,2,1,72,4.196671946153846,m,normal
5	20:28:00,2025-04-20,Sunday,75,0,4,1,80,4.5958421933333335,m,normal
6	20:29:00,2025-04-20,Sunday,133,0,5,0,138,4.605157762557077,m,high
7	20:29:20,2025-04-20,Sunday,50,0,3,1,54,4.402627458823529,m,high
8	20:26:00,2025-04-20,Sunday,37,2,0,0,39,6.2191050882352945,s,heavy
9	20:27:00,2025-04-20,Sunday,40,1,0,3,44,8.827519966216215,s,heavy
10	20:28:00,2025-04-20,Sunday,34,2,0,1,37,12.45306194871795,s,heavy
11	20:29:00,2025-04-20,Sunday,39,0,0,0,39,12.283955096774195,s,heavy
12	20:29:44,2025-04-20,Sunday,36,1,1,0,38,9.29366031111111,s,heavy
13	16:47:00,2025-04-21,Monday,145,1,11,1,158,4.539125304964539,m,high
14	16:48:00,2025-04-21,Monday,103,2,11,2,118,4.286722041493777,m,normal
15	16:49:00,2025-04-21,Monday,84,0,12,2,98,4.320114948275862,m,normal
16	16:50:00,2025-04-21,Monday,150,3,19,1,173,4.068162429487179,m,normal
17	16:51:00,2025-04-21,Monday,178,2,15,0,195,4.031666676470588,m,high
18	16:52:00,2025-04-21,Monday,132,0,13,1,146,4.355515123636363,m,normal
19	16:52:10,2025-04-10,Monday,13,0,0,0,13,5.098095228571428,m,normal

	time	date	day_of_week	car_count	motorcycle_count	truck_count	bus_count	total	avg_exit_time_seconds	road_status	traffic_situation
0	20:25:00	2025-04-20	Sunday	140	1	10	0	151	4.693563	m	high
1	20:26:00	2025-04-20	Sunday	74	1	7	2	84	5.676193	m	normal
2	20:27:00	2025-04-20	Sunday	69	0	2	1	72	4.196672	m	normal
3	20:28:00	2025-04-20	Sunday	75	0	4	1	80	4.595842	m	normal
4	20:29:00	2025-04-20	Sunday	133	0	5	0	138	4.605158	m	high
5	20:29:20	2025-04-20	Sunday	50	0	3	1	54	4.402627	m	high
6	20:26:00	2025-04-20	Sunday	37	2	0	0	39	6.219105	s	heavy
7	20:27:00	2025-04-20	Sunday	40	1	0	3	44	8.827520	s	heavy
8	20:28:00	2025-04-20	Sunday	34	2	0	1	37	12.453062	s	heavy
9	20:29:00	2025-04-20	Sunday	39	0	0	0	39	12.283955	s	heavy
10	20:29:44	2025-04-20	Sunday	36	1	1	0	38	9.293660	s	heavy
11	16:47:00	2025-04-21	Monday	145	1	11	1	158	4.539125	m	high
12	16:48:00	2025-04-21	Monday	103	2	11	2	118	4.286722	m	normal
13	16:49:00	2025-04-21	Monday	84	0	12	2	98	4.320115	m	normal

Machine learning output :-

original class :

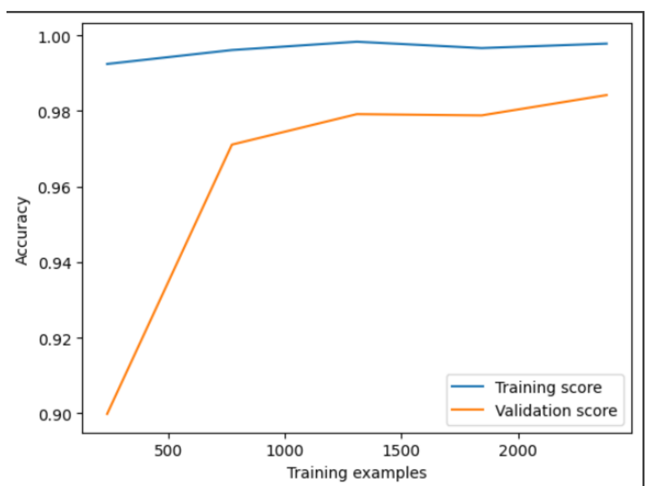


Traffic Situation	count
normal	1669
heavy	682
high	321
low	304

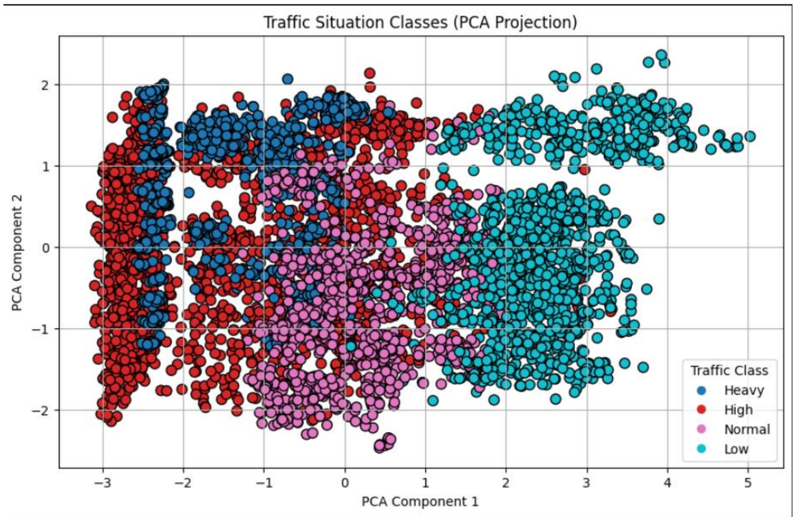
dtype: int64


```
Adjusted Rand Index: 0.266
Normalized Mutual Info: 0.378

Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       304
           1       0.73      0.97      0.83      1669
           2       0.00      0.00      0.00       321
           3       0.85      0.95      0.90       682

    accuracy                           0.76      2976
   macro avg       0.39      0.48      0.43      2976
weighted avg       0.60      0.76      0.67      2976

Weighted F1-score: 0.672
```

```
Accuracy: 0.986562150055991

Classification Report:
              precision    recall  f1-score   support

           1       0.97      1.00      0.98        93
           2       0.99      0.99      0.99       515
           3       0.98      0.94      0.96        88
           4       0.99      1.00      0.99       197

    accuracy                           0.99       893
   macro avg       0.98      0.98      0.98       893
weighted avg       0.99      0.99      0.99       893


Confusion Matrix:
 [[ 93   0   0   0]
 [  3 508   2   2]
 [  0   5  83   0]
 [  0   0   0 197]]
```

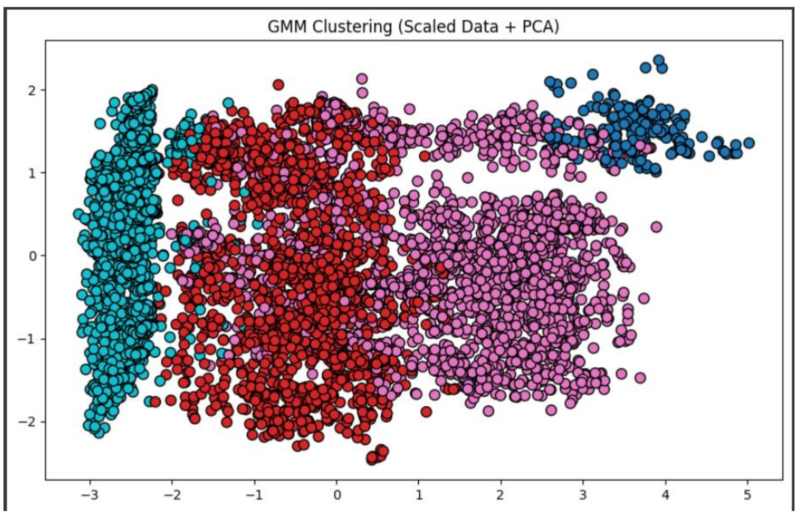


use SMOTENC with constant add:




```
sampling_strategy = {
    2: 1669 ,   # Add 7
    1: 321 + 900,    # 
    3: 304 + 900,    # 
    4: 682 + 700
}
```

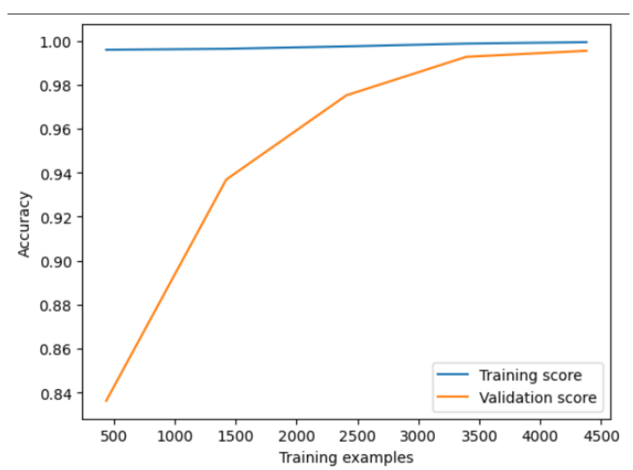
GMM:



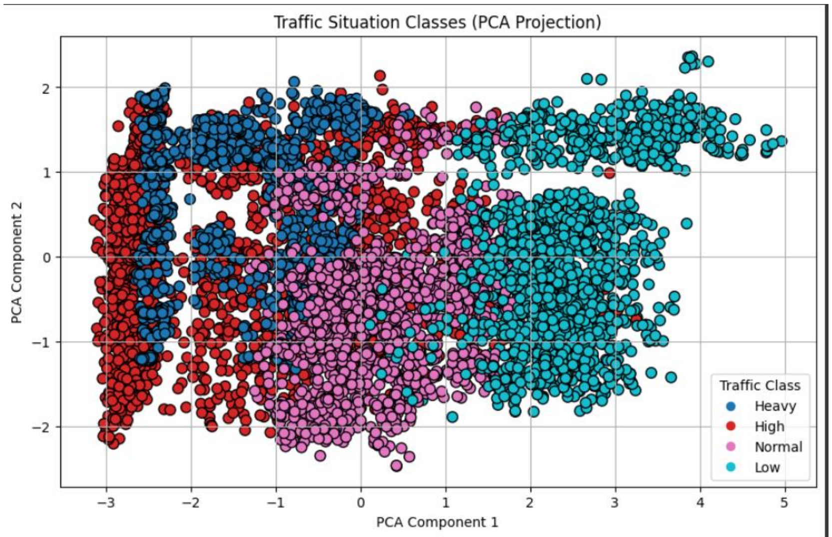

```
Adjusted Rand Index: 0.186
Normalized Mutual Info: 0.272

Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      1221
           1       0.62      0.43      0.51      1669
           2       0.42      0.66      0.51      1204
           3       0.56      0.98      0.71      1382

    accuracy                           0.52      5476
   macro avg       0.40      0.52      0.43      5476
weighted avg       0.42      0.52      0.45      5476

Weighted F1-score: 0.447
```

Random Forest:

```
Accuracy: 0.9896530736457699

Classification Report:
              precision    recall  f1-score   support

           1       0.99      1.00      0.99       352
           2       0.98      0.99      0.98       523
           3       1.00      0.98      0.99       374
           4       1.00      0.99      0.99       394

    accuracy                           0.99      1643
   macro avg       0.99      0.99      0.99      1643
weighted avg       0.99      0.99      0.99      1643


Confusion Matrix:
 [[352   0   0   0]
 [  4 519   0   0]
 [  0   9 365   0]
 [  0   4   0 390]]
```



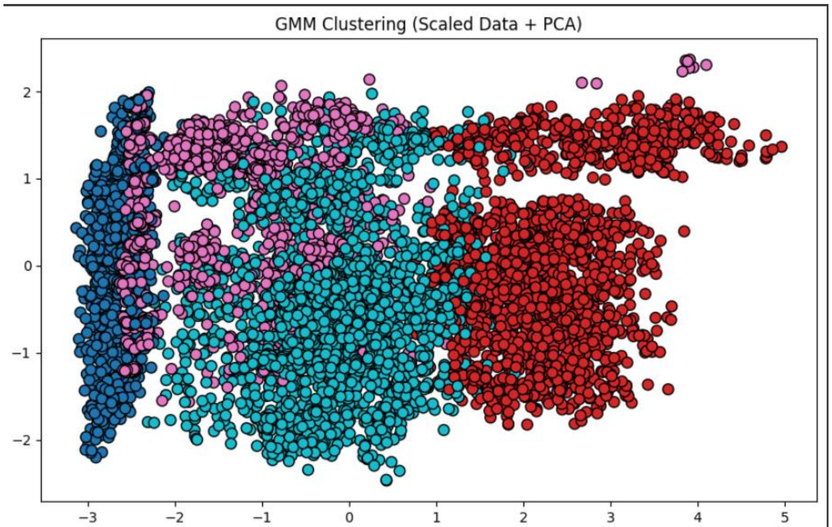
use SMOTENC with same amount of all class :



Traffic Situation	count
1	1669
2	1669
4	1669
3	1669

dtype: int64

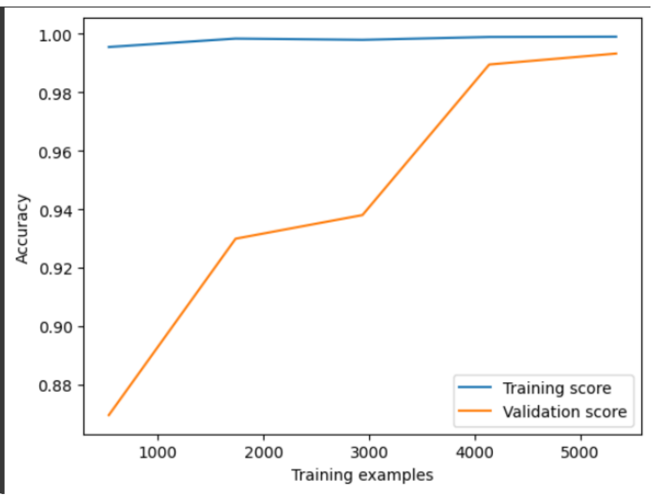
GMM:




```
Adjusted Rand Index: 0.509
Normalized Mutual Info: 0.539

Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.82      0.82      1669
           1       0.77      0.40      0.53      1669
           2       0.63      0.84      0.72      1669
           3       0.83      0.95      0.88      1669

    accuracy                           0.75      6676
   macro avg       0.76      0.75      0.74      6676
weighted avg       0.76      0.75      0.74      6676

Weighted F1-score: 0.738
```

Random Forst :

```
Accuracy: 0.9940089865202196

Classification Report:
              precision    recall  f1-score   support

           1       1.00      1.00      1.00       503
           2       0.99      0.99      0.99       522
           3       1.00      0.99      0.99       487
           4       1.00      1.00      1.00       491

    accuracy                           0.99      2003
   macro avg       0.99      0.99      0.99      2003
weighted avg       0.99      0.99      0.99      2003


Confusion Matrix:
 [[503   0   0   0]
 [  2 516   2   2]
 [  0   5 482   0]
 [  0   1   0 490]]
```



7. Conclusion

Accomplishments to this point:-

1. Data collection:

   Data collection was one of the major steps to make this project valuable and useable for

   For real-life problems because we collected video data from one of the roads on our city which helped us analyze real-life problem in our city which we can understand what's happening

2. Preprocessing:

   Preprocessing the data made us go through a related academic paper to find the best way to deal with traffic data which almost all of them has the same nature we found models to help us solve common problems that happened to other researchers like Imbalance datasets and unwanted features

3. Proposed models:

   Proposed models took us in a journey of research where each one of us searched and tried the models he found to help build the project where we found Models like (Random Forest, GMM, YOLOv8), all these models resemble the fundamental stones of our project

4. Literature Review:

   Literature Reviews where found most of the information's we need for this project we've read total of 8 papers to reach to this results

Challenges:

Data collection

Collecting the data was a hard challenge for us to find a data that suits the problem and made us satisfied that we're working on something real not just any random data

Late decisions

Our decision to go to deep learning (yolo) was late which made us not collect enough data and we could've made our project bigger and solve more problems

Next steps:

More detection:

alerts when a car crash occurs or traffic violations when the driver don't leave enough space between the cars or passing the cars in a crazy way to keep the street as safe as possible.

Traffic simulation:

Traffic simulation to make a generated video scenarios that will happen or could happen this will help the decision makers to create a plan to manage the traffic before a certain event or to make a hypothesis for hajj busses scenario