Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2023) 31: 1099 – 1112

doi:10.55730/1300-0632.4036

YOLO and LSH-based video stream analytics landscape for short-term traffic density surveillance at road networks

K LAVANYA[1], **Stuti TIWARI**[1], **Rahul ANAND**[1], **Jude HEMANTH**[2,*]

[1]School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamilnadu, India
[2]Department of Electronics and Communication Engineering, Karunya University, Coimbatore, Tamilnadu, India

Received: 15.02.2023 • **Accepted/Published Online:** 05.06.2023 • **Final Version:** 27.10.2023

Abstract: The duty of monitoring traffic during rush hour is difficult due to the fact that modern roadways are getting more crowded every day. The automated solutions that have already been created in this area are ineffective at processing enormous amounts of data in a short amount of time, leading to ineffectiveness and inconsistent results. The YOLO (you only look once) and LSH (locality sensitive hashing) algorithms are combined with the Kafka architecture in this study to create a method for assessing traffic density in real-time scenarios. Our concept, which is specifically designed for vehicular networks, predicts the traffic density in a given location by gathering live stream data from traffic surveillance cameras and transforming it into frames (at a rate of 11 per minute) using the YOLOv3 algorithm, which is a crucial parameter for performing effective traffic diversion by suggesting alternate routes and avoiding traffic congestion. The predicted density is then projected onto Google Maps for the convenience of local clients. The comparative study's results demonstrate that our strategy consistently and accurately predicts vehicular density, with an accuracy of more than 90 percent under all conditions. It also shows a significant improvement in both precision and recall, with a 4.08 percent improvement.

Key words: Short-term traffic analysis, video streaming, Apache Kafka, image frames, you only look once, locality sensitive hashing

1. Introduction

Increasing mobility effectiveness and security has been a top priority for the automobile industry and government agencies in recent years. However, there is a serious concern about vehicle traffic jams and traffic congestion in general, especially in urban areas around the world. In addition to increasing commute times, petrol costs, and pollutants, heavy traffic in cities also increases user fatigue and dissatisfaction, which can result in psychological anguish. However, the studies [1, 7] lack a detailed description of the data set used in the experiment, making it difficult for other researchers to replicate the experiment. Some of the causes of commuter traffic include a lack of maintenance, poorly designed roads, narrow traffic lanes, and peak hour rushes. These factors have a negative effect on the state of the road network, making it possible to come across extremely congested areas where cars are sluggish or even stall. The works [9, 15] lack details about training and comparisons with other state-of-the-art methods. Quick congestion forecasting has recently received intelligent transportation system (ITS) reliability in smaller subset regions with higher accuracy [1, 7]. But the limitations of technology in traffic population capture have a major impact in accuracy of the models in traffic prediction. A precise route forecasting model can help emergency crews, including ambulances, fire engines, police cars, and military

* Correspondence: judehemanth@karunya.edu

vehicles, tremendously. The study [4] does not provide clear evidence of performance in real time situations. For effective transportation administration, a precise and real-time traffic flow forecast is vital [1–3]. There have been some unique methods published, such as the AKNN-AVL two-stage pattern classification strategy, which boosts forecast accuracy by combining the advanced k-nearest neighbour (AKNN) algorithm and the balanced binary tree (AVL) data structure [2]. Additionally, a statistics method has been created that integrates the software-defined vehicular networks (SDVN) architecture with data mining methods to produce an AI-powered model to predict traffic volume behaviour [3]. Despite being extremely effective, the techniques offered are inappropriate for some municipal roadway segments. Additionally, it is necessary to improve the functionality of the deployed models for real-world systems [2, 3, 5]. Multiple classifiers with maximum voting (MP-EVM-MCMV) techniques, for example, have 96.5 percent accuracy but are slow to handle vast amounts of data from traffic surveillance cameras [4]. Road traffic regulation is difficult as a result of problems with traffic and vehicle congestion, which are also significant barriers in the field of autonomous driving [6]. Sensor congestion prediction utilising the OWENN algorithm has been suggested in the past to address this issue; however, its energy usage and replenishment solutions have not yet been developed [7]. However, it paved the way for energy-efficient forecasting techniques to be used by smart city systems. One of the solutions makes use of GIS and VANET technology; however, they could not compete under real-time conditions [8]. The other methods used to predict short-term traffic patterns are IGDMs (inertia grey discrete models) and fusion technologies. They are successful in removing the drawbacks of the conventional DGM technique, but they are seldom utilised with real networks [9, 14]. The dynamic stochastic differential models and AUTM are utilised to provide a more realistic approach to transportation planning. It instructs researchers on the benefits of utilising dynamic features in this traffic prediction and avoidance system [10, 11]. The WAVE MAC algorithm and dynamic spatial-temporal models are offered in other works; however, they are all theoretical in character and have no evidence of practical use [12, 13, 15, 16]. The presented methods allowed channels to get information on short-term traffic congestion in their immediate area, but they were unable to estimate traffic for the rest of the image. They provide limited details on the experimental setup and the results obtained. This lack of information makes it difficult for readers to assess the performance of the proposed method and its potential for real-world applications. Also, the lack of empirical evaluation of the proposed model on real-world traffic data undermines the practicality of the approach. In this research, authors present a model that uses Kafka and deep learning model frames and near-duplicate photo capabilities to analyse traffic load on roadways. By gathering live stream data from traffic surveillance cameras and translating it into frames, which are then used to identify nearby duplicate images, our system, unlike past ones, continuously predicts traffic load. The new image is matched to database data to forecast traffic density in a specific area. The estimated density is finally projected onto Google Maps for effective use by neighbourhood customers. The rest of the paper outlines the suggested strategy for addressing the difficulties caused by traffic congestion, as well as the evaluation and implementation details of our model, comparative studies to support and assess the model's performance, a conclusion, and potential future improvements.

2. Motivation

Mobility contributes significantly to a country's social and financial expansion and profitability. If road networks are proficient, they focus on providing better living standards, along with incentives that also have a constructive stimulative effect, including elevated access to resources such as hospitals, schools, malls, work opportunities, and innumerable assets. Roadways that are inferior in regards to throughput or dependability might have

a monetary expense in the form of lost or diminished profits. As a corollary, traffic congestion issues have been extensively researched for a long time, with the primary purpose of eliminating congestion problems and increasing commuting efficacy. Modern routes become increasingly congested hour by hour, making traffic surveillance during peak hours a challenging task. In addition to adding to driving time, gas mileage, carbon emissions, and personal stress, traffic bottlenecks have major and generally negative effects. It is now one of the most important and significant contributors to pollution and global warming. According to the findings, people who live in cities spend 115 billion dollars on wasted gasoline and 4.8 billion h every year delayed in traffic. Heavy traffic causes 34 h of annual rush hour detours for the average driver, and within a few decades, the cost to the normal traveller has increased by 230 percent [1, 2, 17]. The last factor is the spillover effect from clogged main roads to subsidiary streets and back streets, which has an impact on the liveability of neighbourhoods and particularly the real estate market. As a result, since a few decades ago, increasing mobility effectiveness and security has been a top priority for both the automotive industry and governmental agencies. The number of drivers on the road is quickly rising, making governmental plans ineffective and roadway capacities inadequate. Various authorities are making efforts to develop strategies for enhancing transportation infrastructure and expanding roads and railways [21–23].

As a result, authors aim to develop a model for monitoring traffic load on the highways that utilises the frames and near-duplicate picture capabilities of Kafka and deep learning models. With the help of the OpenCV library, the LSH algorithm, the YOLOv3 algorithm, and the Kafka framework,a method for predicting short-term traffic density in real-time settings is integrated in this paper. Our method effectively predicts the automotive congestion in a certain region and then directs vehicles to areas with lower traffic densities to avoid traffic bottlenecks by superimposing real-time traffic concentrations onto Google Maps.

3. Background study

3.1. Apache Kafka ecosystem

Data can be handled as an uninterrupted and secure stream when it is transmitted using a technique called data streaming. Four high-level phases make up the dataflow of Spark streaming. The first is to transmit information from various sources. Sources for batch or static streaming include HBase, MongoDB, PostgreSQL, MySQL, and Cassandra. Spark SQL or NoSQL is used to run processes on this data, and Spark can then apply deep learning to the data via its MLlib API. The streaming output can also be sent to local file systems and other data storage platforms. For effective real-time streaming and traffic density prediction, the authors have combined technologies like Apache Kafka, YOLOv3, and LSH in the current architecture (Figure 1).

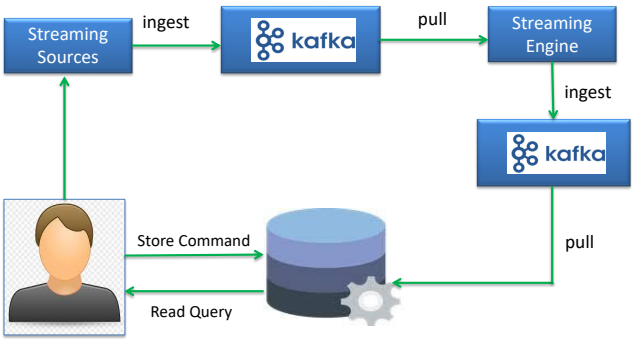

Figure 1. Apache Spark ecosystem.

3.2. YOLOv3

The traffic image is divided into S × S blocks by YOLO, and each block is in charge of identifying targets whose centre points fall within the grid. After recognition, the repeated bounding boxes are removed using nonmaximum suppression. The third generation of YOLO, known as YOLOv3, incorporates a variety of state-of-the-art technologies, such as a residual block-based backbone, a feature pyramid network that acts as the network head for multiscale prediction, batch normalisation, anchor box prediction, etc. For a 256 MP image, it is not possible to distinguish the backdrop from each individual object using the memory trail on standard hardware (NVIDIA Titan X GPUs with 12 GB RAM). Instead, the network interprets contextual background information for each object [24, 25]. The goal of creating a feature map using convolutional and pooling layers with this input image is to enable downsampling. Each cell in the output feature map predicts numerous bounding boxes. In order to reduce model unevenness and reliably recognise dense objects in traffic, authors implement and trial a network architecture that uses 22 custom layers and down samples by a factor of 16 rather than the YOLO standard of 32 down samples. Therefore, a 416 pixel input traffic image outputs a 26 × 26 forecast grid. Our suggested model is inspired by the 28-layer YOLO network, although this new design is primarily enhanced for small and tightly packed objects (Table 1). The loss function defined in YOLO is as follows:

$$\lambda_{coord}\sum_{i=0}^{S^4}\sum_{j=0}^{B}1_{ij}^{obj}\left[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2\right]$$

$$+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left[(sqrt(w_i)-sqrt(\hat{w}_i))^2+(sqrt(h_i)-sqrt(\hat{h}_i))^2\right]$$

$$+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left[(C_i-\hat{C}_i)^2\right]+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{noobj}\left[(C_i-\hat{C}_i)^2\right]+\sum_{i=0}^{S^2}1_i^{obj}\Sigma_{C\in classes}\left(p_i(c)+\hat{p}_i(c)\right)^2(Eq.1), \tag{1}$$

where l_i^{obj} indicates if the object is present in cell i.

l_{ij}^{obj} indicates j^{th} bounding box liable for the estimation of object in the cell i.

λ_{coord} and λ_{noobj} are regularization parameters essential to stabilize the loss function. In this model we take, $\lambda_{coord}=5$ and $\lambda_{noobj}=5$.

Table 1. The architecture of YOLOv3.

Layer	Form	Filters	Size/stride	Size of the output
0	Convolutional	32	3 × 3/1	416 × 416 × 32
1	Maxpool		2 × 2/2	208 × 208 × 32
2	Convolutional	64	3 × 3 /1	208 × 208 × 64
3	Maxpool		2 × 2/ 2	104 × 104 × 64
4	Convolutional	128	3 × 3/1	104 × 104 × 128
5	Convolutional	64	1 × 1/1	104 × 104 × 64
6	Convolutional	128	3 × 3/1	104 × 104 × 128
7	Maxpool		2 × 2/2	52 × 52 × 64
8	Convolutional	256	3 × 3/1	52 × 52 ×256
9	Convolutional	128	1 × 1/1	52 × 52 × 128
10	Convolutional	256	3 × 3/1	52 × 52 × 256
11	Maxpool		2 × 2/2	26 × 26 × 256
12	Convolutional	512	3 × 3/1	26 × 26 × 512
13	Convolutional	256	1 × 1/1	26 × 26 × 256
14	Convolutional	512	3 × 3/1	26 × 26 × 512
15	Convolutional	256	1 × 1/1	26 × 26 × 256
16	Convolutional	512	3 × 3/1	26 × 26 × 512
17	Convolutional	1024	3 × 3/1	26 × 26 × 1024
18	Convolutional	1024	3 × 3/1	26 × 26 × 1024
19	Passthrough		10 -> 20	26 × 26 × 1024
20	Convolutional	1024	3 × 3/1	26 × 26 × 1024
21	Convolutional	Nf	1 × 1/1	26 × 26 × Nf

3.3. Locality sensitive hashing (LSH)

It is a set of elements (known as LSH categories) for hashing data points into different groups in such a way that data points that are close to one another are probably in the same pool, while data points that are far apart are probably in different pools. This speeds up the process of recognising occurrences that resemble one another in different ways. In the realm of computer vision, finding proximal neighbours or nearly identical images is indeed a popular topic of research.

During the first stage of the shingling process, authors combine input surveillance camera frames into a set of n-length glyphs, sometimes referred to as n-shingles or n-grammes. Every frame in the database is represented as an n-shingle combination. Similar compositions are significantly more likely to have shingles overlap, although moving pixels around in a video frame barely affects shingles. In practise, a ratio of 8 to 10 is often used because a low figure could produce a lot of shingles that are present in many video frames, making it challenging to discern between similar pairings (Figure 2).

The equation of the Jaccard index between images A and B is represented as follows:

$J(A,B) = |A \cup B|/|A \cup B|$. $(Eq.2)$

According to the observation, a higher percentage of identical shingles means a higher Jaccard index, demonstrating how much more likely it is for the images to be comparable. Even if the process might end there, the

next step helps handle runtime complexity and spatial complexity, two significant problems that are frequently not addressed in conventional methodologies. The content vector is scarce, and continuing with it will result in a significant storage footprint as well as a lack of scalability. The solution to this conundrum is hashing. The min-hashing function's equation is as follows:

$P[h_n(C_1) = h_n(C_2)] = Sm(C_1, C_2)$ $(Eq.3)$

Min-hashing appears to be the appropriate cryptographic procedure using the Jaccard similarity measure. The statement (Eq. 2) claims that the expected proximity of two indicators is equivalent to the Jaccard similarity of the cells in a video frame. The length of the signatures increases as the divergence decreases. Min-hashing eliminates ambiguity and maintains consistency, thus omitting the worry about computing needs. The technique then eliminates time complexity via locality sensitive hashing. The fundamental idea behind this phase is to create a method that tells if two video frames make up a feasible combination or not based on whether their mutual proximity exceeds a predetermined threshold.

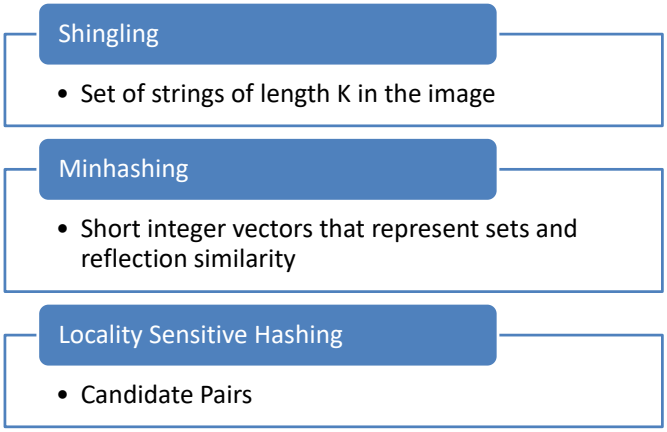

Figure 2. Work flow of locality sensitive hashing algorithm.

4. Methodology

The suggested framework uses a near-duplicate picture identification methodology and video analytics tools in the Kafka framework to develop a traffic surveillance video stream analytics and traffic prediction application. The result will be a system that can handle and process massive amounts of video stream data from security cameras and is scalable, fault-tolerant, efficient, and dependable. A decentralised programming environment called Apache Kafka enables publishing and subscription to data streams. Clients may process data while these entries are maintained in a fault-tolerant way. In the context of our research, it enables us to store and use live feeds or camera recordings in chronological order.

The Python OpenCV library contains a number of algorithms or modules that are useful for image processing or video transformations, enabling the use of live stream films for motion detection, object tracking, and background subtraction [18–20]. YOLOv3 is a method of real-time image processing and analysis that locates certain objects in recordings, live streams, or graphics. It uses traits that a deep convolutional neural network has learned to identify items to detect them. Our concept, which is specifically designed for vehicular networks, enables transportation systems to continuously estimate traffic load by gathering live stream data from traffic surveillance cameras and translating it into frames using the YOLOv3 algorithm.

The frames will be saved and checked in order to prevent data loss. The LSH algorithm, which effectively compares nearly identical frames with the stored dataset and forecasts whether traffic in a region is low or high, is then used to search for images that are nearly identical. Low traffic means there are no more than 10 to 20 vehicles on the route, whereas huge traffic means there is a lot of traffic and is probably indicative of traffic congestion. The predicted traffic density is also visualised on Google Maps for the convenience of local users. This section begins with a description of the key distinguishing features of urban roadmaps and traffic scenarios (Figure 3).

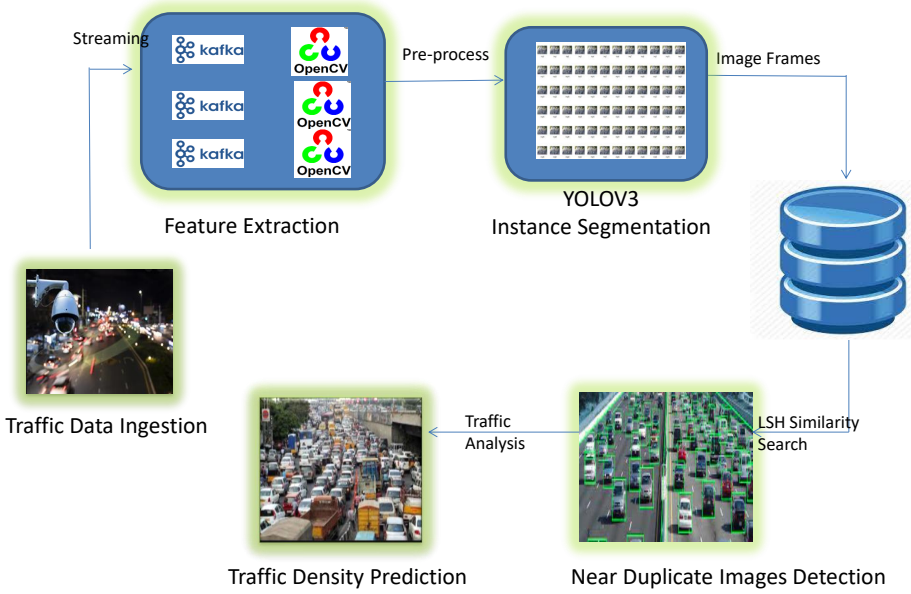

Figure 3. Short-term traffic monitoring framework for road networks.

4.1. Roadmaps used for experimentation

Vehicle traffic statistics were gathered for this study's examination using traffic surveillance cameras at several crossings along the Vellore-Katpadi route. The Chittor transit terminus station, which is adjacent to Vellore Institute of Technology, is one of the most significant ones. All surveillance video data was then processed and analysed, and a dataset was made utilising the information obtained in order to obtain the necessary information. The dataset, which was gathered from the traffic surveillance cameras over a period of three months from April 2022 to June 2022, included every conceivable scenario of traffic that may have been present on the roads at various times (Figure 4). To forecast the amount of traffic on the road, it is helpful to compare the newly acquired video frame with the learned dataset.

4.2. Data processing and frames extraction

In this context, the term "Kafka servers" refers to a decentralised programming environment that aids in enabling, publishing, and subscribing to data streams relevant to our work. It enables us to retroactively retain and use live feeds and video footage from traffic surveillance cameras. The dataset is then produced using recordings made by in-motion traffic monitoring equipment. Additionally, a Kafka topic called "kafka-video-topic" is created to publish videos here, whether they come from the camera or the system's video. The Kafka

server is currently running on localhost. Then authors run the producer code to publish the video using either the recorded video or the live traffic camera feed after creating localhost and running the model for publishing the video. Then, authors employ the YOLOv3 algorithm, a real-time image analysis and processing method that quickly locates particular elements in recordings, live streams, or images. YOLO uses traits that a deep convolutional neural network has learned to identify items and detect them. Authors run the model to split the video into frames and save them locally, as shown in Figure 5, using the YOLOv3 technique from the OpenCV library.

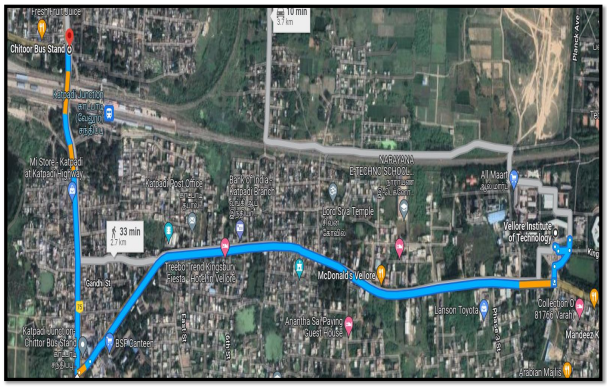

Figure 4. The study area: Vellore-Chitoor transit terminus.

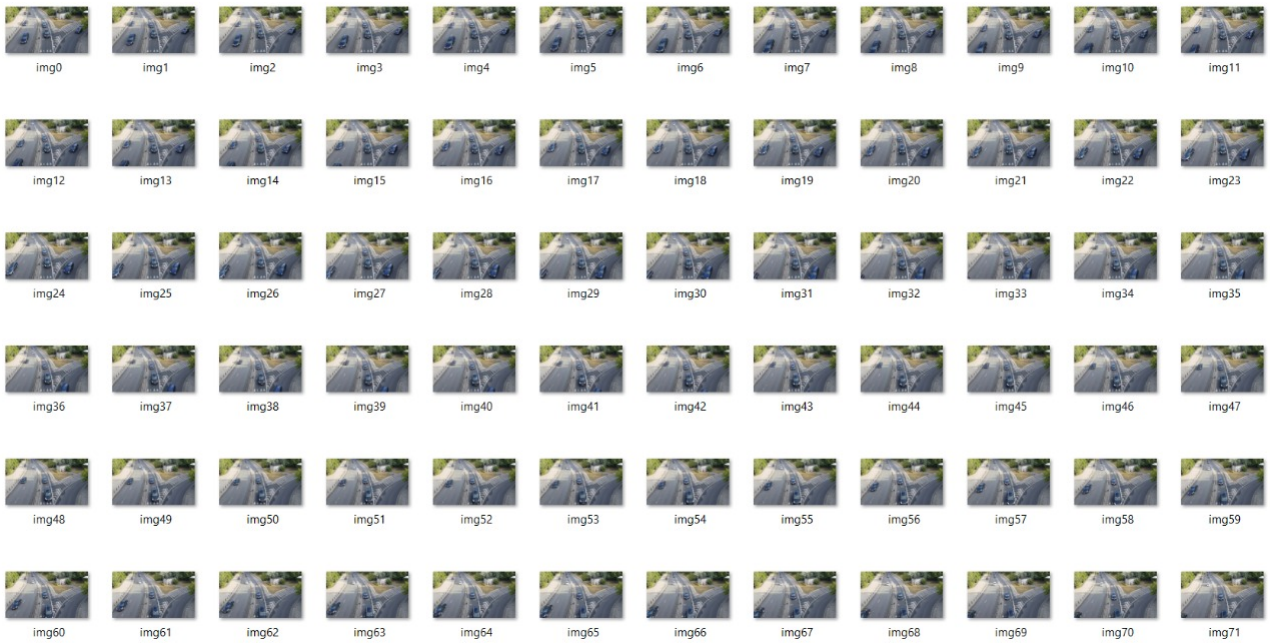

Figure 5. Traffic dataset contains 3 modes of traffic jam: low, medium and heavy.

Once the frames are stored in a local database, an appropriate image identification algorithm is deployed on the stored frames to generate a dataset of near-duplicate images. It is done using a locality sensitive hash

algorithm and random projection. The random projection methodology aids in the reduction of the complexity of a collection of Cartesian coordinates. The issue of organising as well as handling huge information volumes is typically alleviated by reducing multiplicity. In most density-lowering strategies, linear modifications are used to not only determine the multitudinous' innate dimensionality but also identify its primary orientations.

Input: dissect image into q-grams (shingles)
Sketching of an image
From each shingle-set
Build a "sketch vector" (200 size)
Postulate: image that share $\geq t$ components of their sketch-vectors are claimed to be near-duplicates
Sketching by min-hashing
Consider
Pixel matrix of images A, B as S_A, S_B $\in P = \{0, ..., p-1\}$
Pick a random permutation of the whole set P (such as ax+b mod p)
Pick the minimal element of S_A : $\alpha = \min \pi (S_A)$
Pick the minimal element of S_B : $\beta = \min \pi (S_B)$
Lemma:
$\mathrm{p}[\alpha = \beta] = \frac{S_A and S_B}{S_A or S_B}$
Similarity sketch sk(A)
d minimal elements under π S_A
Or take d permutations and the min of each
Note: The variance can be reduced by using a larger d. Typically d is few hundred minutes, approximately 200.

When processing a video frame, the model creates a feature list of every image in the dataset and keeps this list whenever a new image is presented so that it can compare and analyse comparable frames and create images that are almost identical to the original. Additionally, the system compares the density of newly generated frames or images with previously generated, analogous images that have been subjected to established density parameters, and it predicts whether the density of the traffic in that particular frame will be label 0 or label 1 (Figure 6). Label 0 in this case denotes busy traffic, while Label 1 denotes low traffic. In this manner, the model is able to accurately anticipate traffic in a certain location. For effective use by local clients, the anticipated density in the area is projected onto Google Maps.

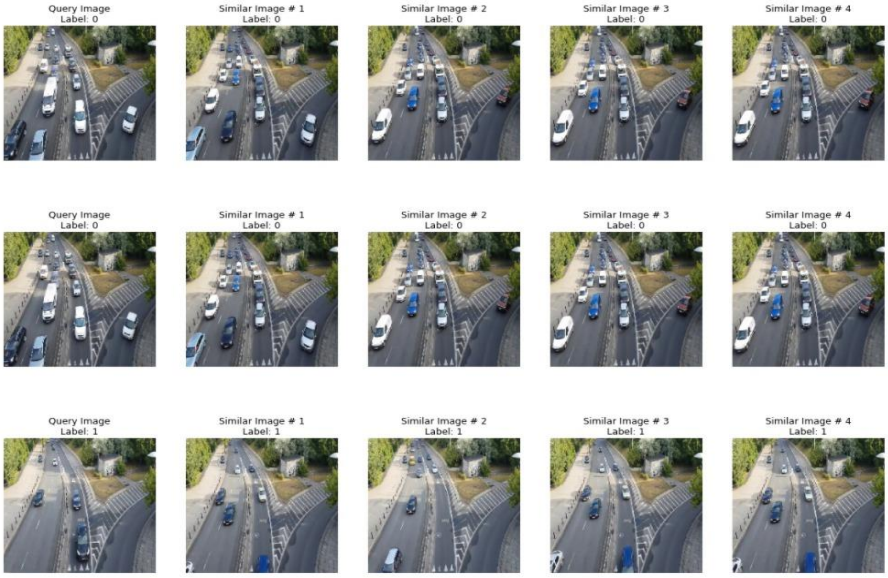

Figure 6. Similarity image detection using LSH.

5. Comparative analysis with existing approaches

During the course of our research, a number of different theories for estimating vehicle congestion were investigated, but they were not quite as successful as LSH's using the random projection paradigm. Most of the methods examined for similarity frame authentication used the K-Nearest Neighbour (KNN) algorithm or clustering techniques. Finding the k-nearest neighbours for each entity in another collection T from another collection D seems to be the cognitively demanding approach of KNN. On a dedicated server, the KNN algorithm cannot produce results quickly due to the expansion of the database parameters and density. In order to solve these scaling issues, researchers describe the LSH as employing a random projection method built into the RT model as a strategy. LSH is frequently used to group pertinent things into the same category, which may also aid in narrowing the focus of record searches. In the TensorFlow framework, TRT is a TensorFlow-TensorRT collaboration that takes inferential improvement on NVIDIA GPUs. Authors investigated LSH as two approaches, i.e. with the TRT and non-TRT model.

Table 2. Time comparison between KNN, LSH with TRT, LSH without TRT models.

Algorithm	Average CPU utilization (percent)	Average GPU utilization (percent)	Time taken (in millisecond)
KNN	70.3	20.1	352.176
YOLO-LSH with TRT	28.3	51.5	151.466
YOLO-LSH without TRT	31.4	40.5	268.673

Evaluation of our YOLO-LSH with TRT and YOLO-LSH without TRT gives insights to learn the characteristics of image frames and to predict the traffic density (Table 2). First, the dataset is split into a training set and a test set. Second, the classification models are trained using 70 percent of the sampled training data, and the remaining 30 percent of test data is used to test the accuracy of each model.

Table 3. Performance metrics of KNN, LSH with TRT, LSH without TRT models.

Algorithm	Precison	Recall	F1-score	Support	Latency	Cumulative gain (percent)	Frames per second	Accuracy
KNN	1.00	0.5	0.67	2	28.38	80.42	6	92.9
YOLO-LSH with TRT	0.67	1.00	0.80	2	30.25	95.90	11	96.98
YOLO-LSH without TRT	1.0	0.80	0.87	2	24.19	92.01	9	94.57

In terms of optimization, Mini Batch is chosen to be our optimizer technique to improve detector performance. Table 3 illustrates the training process using KNN, YOLO-LSH with TRT and YOLO-LSH without TRT. YOLO-LSH with TRT achieves less time consumption in training than other models. From Table 4, it is concluded that YOLO-LSH with TRT is better than other models on validation time, root mean square error (RMSE) and validation loss.

Table 4. Training and validation process results.

Algorithm	Epoch	Iterations	RMSE	Log-loss (percent)	Validation loss
KNN	10	150	87.86	31.51	0.49
	20	300	66.76	29.88	0.47
	30	450	45.84	28.69	0.46
YOLO-LSH with TRT	10	150	86.17	31.59	0.30
	20	300	65.51	29.96	0.32
	30	450	45.04	28.77	0.34
YOLO-LSH without TRT	10	150	89.33	31.44	0.52
	20	300	67.93	29.80	0.53
	30	450	46.56	28.62	0.65

The best accuracies of our traffic density prediction using the deep learning model are 92.9, 96.98, and 94.57, respectively. The accuracy and loss is shown in Figure 7, respectively. Further, to deal with the class imbalance problem, the cross validation method is applied. The mean area under the curve (AUC) score over each label is 97.07. AUC is used to aggregate the performance across all classes of opinions.

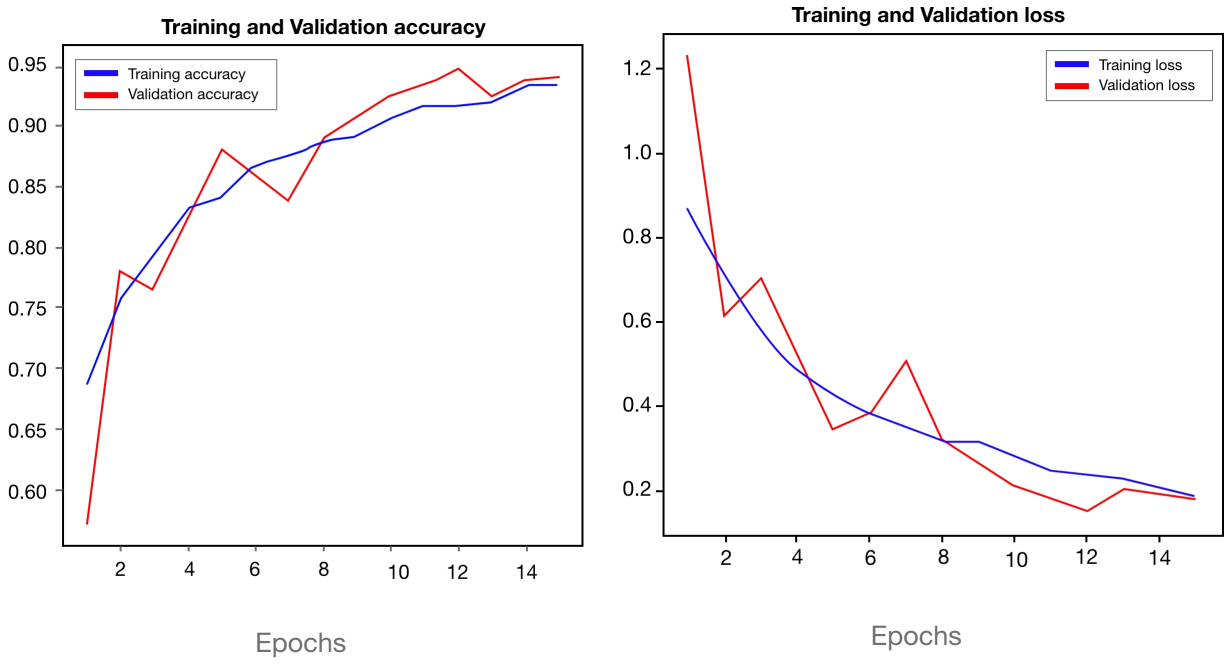

Figure 7. Accuracy and loss graph for training and validation.

6. Conclusion

In Asian countries, residents increasingly relocate to metropolitan hubs in greater numbers, resulting in increased automobile traffic in the metropolis. Video analytics and traffic monitoring systems for traffic prediction may assist in a variety of ways, including public safety on the roads, business intelligence, healthcare, law enforcement, industries, and traffic law violations. Above all, it will suggest the best possible routes that take less time to reach the destination by predicting real-time traffic in an area and projecting it onto Google Maps. In addition to predicting real-time traffic, the developed application can also assist in emergency response situations. By detecting and predicting traffic flow, emergency responders can be directed to the most efficient routes, reducing response times and potentially saving lives. The developed video stream analytics and object detection application for traffic surveillance will recognise and predict traffic using an optimum detection model and near-duplicate images. The built application will be a scalable, fault-tolerant, effective, and dependable system that can handle and process massive amounts of video stream data acquired from security cameras. It will not only help an individual avoid traffic congestion and delays, but it also helps to save money by reducing gasoline use. The rapid forecast allows congestion managers to take proactive measures to regulate traffic flow and avoid bottlenecks. It also addresses the delays and inconsistencies brought on by manual applications and other technical methods. The system can also be integrated with other smart city technologies, such as connected vehicles and intelligent transportation systems, to create a comprehensive traffic management solution.

Acknowledgement

Conceptualization, K Lavanya; methodology, K Lavanya; software, Stuti Tiwari, Rahul Anand; validation, K Lavanya, Stuti Tiwari and Rahul Anand; writing—original draft preparation, K Lavanya, Stuti Tiwari and Rahul Anand; writing—review and editing, Jude Hemanth; visualization, Jude Hemanth. All authors have read and agreed to the published version of the manuscript.

References

[1] Barrachina J, Garrido P, Fogue M, Martinez FJ, Cano JC et al. A V2I-based real-time traffic density estimation system in urban scenarios. Wireless Personal Communications. 2015;83 (1):259-280.https://doi.org/10.1007/s11277-015-2392-4

[2] Bhatia J, Dave R, Bhayani H, Tanwar S, Nayyar A. SDN-based real-time urban traffic analysis in VANET environment. Computer Communications.2020;149: 162-175.https://doi.org/10.1016/j.comcom.2019.10.011

[3] Jose C, Grace KV. A New approach for accurate path Prediction using multiple Prediction system. Materials Today: Proceedings.2020 ;24:1749-1757. https://doi.org/1749-1757.10.1016/j.matpr.2020.03.599

[4] Panovski , Zaharia T Long . Short-term bus arrival time prediction with traffic density matrix. IEEE Access.2020;8:226267-226284 https://doi.org/10.1109/ACCESS.2020.3044173

[5] Alhussain T. Density-scaling traffic management for autonomous vehicle environment—predictive learning-based technique. Soft Computing 2021;25 (18): 12043-12057.https://doi.org/10.1007/s00500-021-05722-4

[6] Neelakandan S, Berlin MA, Tripathi S, Devi VB, Bhardwaj I et al. IoT-based traffic prediction and traffic signal control system for smart city.Soft Computing,2021;25 (18):12241-12248. https://doi.org/10.21203/rs.3.rs-306500/v1

[7] Bhavani MM, Valarmathi A. Smart city routing using GIS and VANET system. Journal of Ambient Intelligence and Humanized Computing,2021; 12 (5):5679-5685. https://doi.org/10.1007/s12652-020-02148-y

[8] Duan H, Xiao X, Xiao Q. An inertia grey discrete model and its application in short-term traffic flow prediction and state determination. Neural Computing and Applications.2020;32(12):8617-8633. https://doi.org/8617-8633.10.1155/2017/3515272

[9] Su Y,Sun W. Dynamic differential models for studying traffic flow and density.Journal of Ambient Intelligence and Humanized Computing 2019; 10 (1):315-320. https://doi.org/315-320.10.1007/s12652-017-0506-4

[10] Hu W, Wang H, Qiu Z, Yan L, Nie C et al. An urban traffic simulation model for traffic congestion predicting and avoiding. Neural Computing and Applications 2018;30 (6):1769-1781. https://doi.org/10.1007/s00521-016-2785-7

[11] Lee YS, Kim SH. A proposal of novel design on the WAVE MAC algorithm with low-latency for seamless video surveillance in V2X environment. Multimedia Tools and Applications.2020; 79 (23):16035-16049. https://doi.org/16035-16049.10.1007/s11042-019-7151-1

[12] Zheng L, Yang J, Chen L, Sun D, Liu W.Dynamic spatial-temporal feature optimization with ERI big data for Short-term traffic flow prediction. Neurocomputing.2020; 412:339-350.https://doi.org/10.1016/j.neucom.2020.05.038

[13] Yinglei H, Dexin Q, Shengyuan Z. Smart transportation travel model based on multiple data sources fusion for defense systems. Soft Computing 2022;26 (7);3247-3259. https://doi.org/3247-3259.10.1007/s00500-022-06825-2

[14] Kong X, Zhang J, Wei X, Xing W, Lu W. Adaptive spatial-temporal graph attention networks for traffic flow forecasting.Applied Intelligence 2022; 52 (4):4300-4316.https://doi.org/10.1007/s10489-021-02648-0

[15] Mounica B, Lavanya K. Real Time Traffic Prediction Based On Social Media Text Data Using Deep Learning. Journal of Mobile Multimedia 2021;18 (2):373–392. https://doi.org/10.13052/jmm1550-4646.18211

[16] Alqaness MA, Abbasi AA, Fan H, Ibrahim RA, Alsamhi SH et al. An improved YOLO-based road traffic monitoring system. Computing 2021;103 (2):211-230. https://doi.org/211-230.10.1007/s00607-020-00869-8

[17] Zhou S, Bi Y, Wei X, Liu J, Ye , Li F et al. Automated detection and classification of spilled loads on freeways based on improved YOLO network. Machine Vision and Applications.2021; 32 (2):1-12.https://doi.org/1-12.10.1007/s00138-021-01171-z

[18] Ihueze CC, Onwurah UO. Road traffic accidents prediction modelling: An analysis of Anambra State, Nigeria. Accident Analysis and Prevention 2018; 112:21-29.https://doi.org/10.1016/j.aap.2017.12.016

[19] Barros J, Araujo M, Rossetti RJ. Short-term real-time traffic prediction methods: A survey. In 2015 International Conference on Models and Technologies for Intelligent Transportation Systems.IEEE (MT-ITS).2015. https://doi.org/132-139.10.1109/MTITS.2015.7223248

[20] Ke R, Li Z, Tang J, Pan Z, Wang Y. Real-time traffic flow parameter estimation from UAV video based on ensemble classifier and optical flow. IEEE Transactions on Intelligent Transportation Systems.2018; 20 (1):54-64. https://doi.org/54-64.10.1109/TITS.2018.2797697

[21] Abbasi M, Shahraki A, Taherkordi A. Deep learning for network traffic monitoring and analysis (NTMA): A survey. Computer Communications 2021:170; 19-41.https://doi.org/10.1016/j.comcom.2021.01.021

[22] Sundareswaran A, Lavanya K. Real-time vehicle traffic prediction in apache spark using ensemble learning for deep neural networks. International Journal of Intelligent Information Technologies.2020;16 (4): 19-36.https://doi.org/10.4018/IJIIT.2020100102

[23] Adoni WYH, Nahhal T, Aoun NB, Krichen M, Alzahrani M. A Scalable Big Data Framework for Real-Time Traffic Monitoring System.Journal of Computer Science.2022;18 (9):801-810. https://doi.org/10.3844/jcssp.2022.801.810

[24] Hashemi H, Abdelghany K. End-to-end deep learning methodology for real-time traffic network management. Computer-Aided Civil and Infrastructure Engineering.2018;33 (10):849-863.https://doi.org/10.1111/mice.12376

[25] Chen C, Liu B, Wan S, Qiao P, Pei Q. An edge traffic flow detection scheme based on deep learning in an intelligent transportation system. IEEE Transactions on Intelligent Transportation Systems.2020;22 (3):1840-1852.https://doi.org/10.1109/TITS.2020.3025687