IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 9 November 2023, accepted 26 December 2023, date of publication 1 January 2024,
date of current version 10 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3349034

RESEARCH ARTICLE

A Graph-Based Framework for Traffic Forecasting and Congestion Detection Using Online Images From Multiple Cameras

**BOWIE LIU[1], CHAN-TONG LAM[1], (Senior Member, IEEE),
BENJAMIN K. NG[1], (Senior Member, IEEE),
XIAOCHEN YUAN[1], (Senior Member, IEEE),
AND SIO KEI IM[2], (Member, IEEE)**

[1]MPU-UC Joint Research Laboratory in Advanced Technologies for Smart Cities, Faculty of Applied Sciences, Macao Polytechnic University, Macau, China
[2]Engineering Research Centre of Applied Technology on Machine Translation and Artificial Intelligence, Macao Polytechnic University, Macau, China

Corresponding author: Chan-Tong Lam (ctlam@mpu.edu.mo)

This work was supported by The Science and Technology Development Fund, Macau SAR under Grant 0044/2022/A1.

ABSTRACT Many countries across the globe face the serious issue of traffic congestion. This paper presents a low-cost graph-based traffic forecasting and congestion detection framework using online images from multiple cameras. The advantage of using a graph neural network (GNN) for traffic forecasting and detection is that it represents the traffic network in a natural way. This framework requires only images from surveillance cameras without any other sensors. It converts the online images into two types of data: traffic volume and image-based traffic occupancy. A clustering-based graph construction method is proposed to build a graph based on the traffic network. For traffic forecasting, multiple models, including statistical models and deep graph convolutional neural networks (GCNs), are used and compared using the extracted data. The framework uses logistic regression to determine the threshold of traffic congestion. In the experiment, we found that the Decoupled Dynamic Spatial-Temporal Graph Neural Network (D2STGNN) model achieved the best performance on the collected dataset. We also propose a threshold-based method for detecting traffic congestion using traffic volume and image-based traffic occupancy. This framework provides a low-cost solution for traffic forecasting and congestion detection when only surveillance images are available.

INDEX TERMS Traffic forecasting, traffic congestion detection, online images, graph convolutional neural networks, logistic regression.

I. INTRODUCTION

Many countries across the globe face the serious issue of traffic jams. They result in huge delays in transportation and excessive consumption of fuel and money [1]. In the context of a smart city, transportation produces a huge amount of data that reflects the status of the city. In recent years, artificial intelligence has rapidly developed and accomplished a lot of tasks that could not be accomplished by traditional methods. For example, the traffic flow can be predicted using the data produced previously, thereby notifying people of the future traffic status and improving traffic efficiency. In addition, based on the collected data, a method for detecting traffic congestion can be developed, which can provide useful information for people.

Currently, various types of data and approaches are used for traffic forecasting and congestion detection. The types of data include speed, travel time, delay, etc [2]. The data is usually collected from various sensors or produced from video image processors [3]. Different methods based on neural networks have been developed for traffic forecasting and replaced statistical methods [4], [5], [6]. In addition to determining from sensor data, different approaches based on images were developed for congestion detection [7], [8], [9], [10], [11].

The associate editor coordinating the review of this manuscript and approving it for publication was Frederico Guimarães.

For the public to view traffic status conveniently, the Macao government provides a website that offers instant traffic stats in images or videos from approximately 100 cameras in different traffic nodes [12]. These images or videos show the real-time traffic status of their corresponding traffic node, including the shape of the road and vehicles on the road. Based on this website, useful information can be collected and extracted from the images and videos provided by the website. For example, the number of vehicles can be obtained using off-the-shelf object detection techniques.

This paper proposes a graph-based framework to extract useful information from real-time online traffic images for predicting traffic status and detecting traffic congestion. This framework is low-cost and off-the-shelf as it can be directly deployed on the system with surveillance cameras. Compared with our previous works on congestion detection [7], [8], [9], [11], we propose to consider multiple camera sites for congestion detection. Moreover, we consider traffic forecasting in this paper. The contributions of this paper can be summarized as follows:

- We propose a graph-based framework for traffic forecasting and congestion detection using online images from multiple cameras, converting images to numerical values;
- We build a dataset extracted from a period of collected images from multiple cameras;
- We design an improved method that builds a graph network from the traffic network;
- We evaluate and compare a number of models in traffic forecasting using the extracted data from the online images;
- We determine the numerical thresholds for detecting traffic congestion for different traffic nodes.

The remaining of this paper is organized as follows. Section II reviews the related works on traffic flow prediction and congestion detection. Section III describes the process of collecting and processing the data from the DSAT website and a method for graph construction. Section IV introduces the statistical and deep learning methods used for traffic forecasting and an approach for congestion detection. Section V shows the experimental results for both traffic forecasting and congestion detection, and a discussion of the results. Section VI concludes the paper.

II. LITERATURE REVIEW

The measures for traffic flow are necessary to estimate traffic status and define traffic congestion for congestion detection. There are many commonly used measures mentioned in [13], such as traffic flow or volume, speed, and occupancy. Afrin et al. [2] provided more measures with corresponding traffic status. For example, the volume-to-capacity ratio (V/C), which can be calculated as:

$$\mathrm{V/C} = N_v / N_{max} \tag{1}$$

where N_v indicates the spatial mean volume, and N_{max} denotes the maximum number of vehicles of the road segment. The N_{max} can be further expanded as:

$$N_{max} = (L_s / L_v) \times N_l \tag{2}$$

where L_s is the length of the road segment, L_v is the average length occupied by a vehicle which includes the safety distance between vehicles and vehicle length, and N_l denotes the number of lanes. $\mathrm{V/C} < 0.6$ indicates the traffic flow is smooth and free. $0.6 < \mathrm{V/C} < 1.0$ indicates the speed of traffic flow is affected. When $\mathrm{V/C} > 1.0$, it indicates a breakdown traffic flow.

Different approaches have been developed for traffic congestion detection. Sun et al. [14] proposed an approach to identify traffic congestion based on threshold values. The threshold values are determined with mutual information maximization theory between discrete traffic flow parameters and traffic state. Then a decision tree is used to extract traffic congestion identification rules. Wang et al. [15] proposed an approach using surveillance images. It extracts texture features as low-level features from images. Then, it uses the proposed Locality Constraint Metric Learning to produce a distance metric. Finally, it uses Kernel Regression to predict the congestion level based on the learned metric. Lam et al. [8] proposed a multiple IoU (mIOU) method to evaluate traffic level, which is calculated by applying intersection over union (IoU) to all vehicles on the image. It also provides a set of thresholds for estimating traffic congestion levels.

Neural networks are widely used in traffic congestion detection as well. Kurniawan et al. [16] proposed a congestion detection method using an image classification approach. It uses a set of CCTV monitoring images and processes them through a sequence of operations, including resizing, gray-scaling, and normalization. Then, it builds a convolutional neural network (CNN) with a simple, basic structure to detect traffic congestion. Ke et al. [17] also proposed a CNN-based method. It extracts multiple features from traffic images and then fuses them into multidimensional features. Then it uses a CNN classifier to predict the congestion level. Chakraborty et al. [10] compared the performance of detecting traffic congestion from traffic images for YOLO, deep convolution neural network (DCNN), and support vector machine (SVM). As a result, YOLO achieved the best accurancy. Cho et al. [18] proposed a method to classify the density of road networks using an image generation approach. The nodes in the traffic network are converted to polygons whose shapes represent traffic conditions in different directions.

There are various algorithms used for traffic flow forecasting. Auto-regressive integrated moving average (ARIMA) is a time series analysis model that can be used for traffic flow forecasting [19], [20]. Kalman filtering is an algorithm that predicts future states using a series of historical data with noise, which were used in traffic flow forecasting as well [21].

After deep learning techniques are widely used, many related works based on this have been done for predicting traffic information. Those works can be basically classified

into two types: CNN-based and GNN-based. CNN-based models usually convert the information into images for further convolution operation. Ma et al. [22] proposed a CNN-based method that converts spatio-temporal traffic dynamics to two-dimensional time-space images. The space dimension describes the detected values on a consecutive section of road and the time dimension describes the detected values of a certain section at different times. Then the convolutional layer can extract features in coherent areas. Chen et al. [23] proposed PCNN to process periodic data using convolution operaton. PCNN first folds the time series data to a two-dimensional matrix as the input, then a series of convolutions is used to extract features.

CNN-based models require the data modeled in Euclidean space. Considering the structure of the traffic network, the graph is better at representing the network because the traffic nodes and roads can be considered as nodes and connections in a graph, respectively. Li et al. [24] introduced Diffusion Convolutional Recurrent Neural Network (DCRNN) for traffic forecasting considering diffusion inside traffic networks. DCRNN models the traffic flow using random walks in two directions. It uses the encoder-decoder structure to extract temporal features. Yu et al. [25] proposed a Spatio-Temporal Graph Convolutional Network (STGCN) to apply convolution in both spatial and temporal dependency. Using graphs to represent the traffic data, this model enables much faster training speed with fewer parameters. Li et al. [26] proposed a Dynamic Graph Convolutional Recurrent Network (DGCRN). It contains a dynamic graph generator to produce a dynamic graph based on node embeddings, and integrates the dynamic graph with the input static graph. Then it uses graph convolutions and temporal decoder to generate predictions. Jin et al. [27] proposed a method that uses Wasserstein Generative Adversarial Nets (WGAN) for traffic forecasting based on a generative adversarial network (GAN). The generator predicts the link speeds. The model consists of GCN, RNN, and attention mechanism to capture the spatial-temporal relations. Shao et al. [28] proposed Decoupled Dynamic Spatial-Temporal Graph Neural Network (D2STGNN) for traffic prediction. It uses a decoupled spatial-temporal framework to decouple the traffic signals into diffusion signals and inherent signals. Then it uses two networks to handle these two types of signals. It also contains a dynamic graph learning model.

The related works mainly focus on improving performance on a particular task. Our work in this paper focuses on proposing a framework for traffic prediction and congestion detection using real-time online images.

III. DATA COLLECTION AND PROCESSING

In this section, the whole process of data collection will be introduced, including image collection, traffic information extraction, and graph construction. Firstly, we will introduce the process of collecting images from the DSAT official website, followed by a description of the methods to convert images into numeric data that reflects traffic information. Finally, the method for constructing the graph among traffic nodes will be proposed. In this paper, we use the real-time images collected from the DSAT official website to illustrate the effectiveness of the proposed approach. Note that the proposed framework is applicable to the scenarios where online or CCTV images for traffic monitoring are available.

A. IMAGE COLLECTION

The traffic instant images are collected from the DSAT official website. Figure 1 shows an example of typical traffic instant images. The DSAT provides instant traffic surveillance images or videos of about 100 cameras. The cameras are spread in traffic nodes in Macao Peninsula and Taipa. Each camera is assigned a number which appears in the link of this camera on the DSAT website. We use the number as the identity of each camera. Before starting to collect images, the image quality of each camera needs to be verified to ensure that the extracted information in the next step is useful. Therefore, cameras that produce blurry images frequently or have a bad angle of view are filtered. Similarly, the cameras on traffic nodes that have very little traffic flow are also filtered because the traffic forecasting for them is not useful. Finally, we select a set of cameras that can produce quality images and cover most of the area in Macao. As a result, 16 cameras were chosen as the sources of image collection. The collection script is implemented in Python. Considering network and server conditions and data quality, the time interval between two collections is set to 2 minutes. This is reasonable for congestion detection and traffic prediction because significant change of traffic flow requires longer time interval.

FIGURE 1. Example of traffic instant images.

B. TRAFFIC INFORMATION EXTRACTION

Extracting useful information from images from different cameras is a crucial and difficult task. The cameras may have different angles and points of view. In addition, the collected images may vary in size and resolution. The

traditional method may not be able to produce stable results. Fortunately, the deep learning-based method can detect vehicles in the images directly, which provides a reasonable foundation for extracting more commonly used measures in traffic congestion estimation. Two types of relatively static measures are extracted, which are instant traffic volume and image-based traffic occupancy.

1) INSTANT TRAFFIC VOLUME

Traffic volume is usually defined as the number of vehicles on a particular length of road. However, due to the variety in properties of different cameras, it is difficult to estimate unit length in images. In this work, we extract instant traffic volume as a measure of traffic status. Instant traffic volume is defined as the total number of vehicles on an image. Compared to normal traffic volume, the instant traffic volume is easier to extract as it only needs to know the number of vehicles on a single frame without other prior knowledge. For exchange, it cannot be used to estimate traffic congestion directly because the lengths or areas of roads captured by different cameras may vary.

Figure 2 illustrates the process of calculating instant traffic volume for an image. YOLOv5 [29] is used as the object detector because it is accurate, fast, and easy to deploy. However, YOLOv5 may give one object two different labels. To avoid duplicate labels, the intersection over union (IoU) can be used to detect if any duplicates exist. Then the instant traffic volume can be calculated by counting the number of valid bounding boxes.

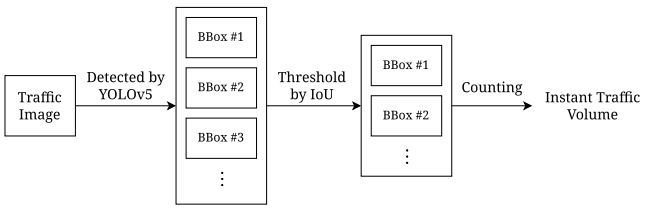


FIGURE 2. The process of calculating instant traffic volume.

2) IMAGE-BASED TRAFFIC OCCUPANCY

Similar to instant traffic volume, image-based traffic occupancy is a variant of traffic occupancy. Traffic occupancy is defined as the percentage of time that the detection zone has been occupied by vehicles. With only one frame, we can use the ratio of the area occupied by vehicles and the total area of the road to estimate traffic occupancy. In this paper, image-based traffic occupancy, which is defined as the ratio of the total area of detected vehicle bounding boxes and the total area of the road, is extracted as the second measure. It has a limitation which is relatively unstable because it is affected by the angle of the camera and the distance of the detected vehicle. For example, a vehicle may cover more area of the road when the angle between the camera and the ground is closer to 90 degrees, or when the vehicle is closer to the camera. We can use perspective transformation to transform the image to an aerial view, which reduces the impact of the inconsistent distance for vehicles. After perspective transformation, vehicles far from the camera will appear much larger. To avoid this, the transformation should not include areas too far from the camera. An advantage of the occupancy is that it is naturally between 0 and 1, which is potentially capable of providing information for estimating traffic congestion.

Figure 3 illustrates the process of calculating image-based traffic occupancy. For each camera, we manually labeled a mask for the region of interest which represents the area of the road segment, and a set of corresponding points for 4-point perspective transformation. First, when processing a traffic image, it is detected by YOLOv5 to produce the bounding boxes of objects. Objects other than cars, trucks, buses, and motorcycles are removed. Secondly, the bounding boxes are plotted on the region of interest (ROI), and the parts of the boxes where not inside are cropped by the mask. Then, the bounding boxes and the mask are projected onto a plane by 4-point perspective transformation. Finally, the image-based traffic occupancy of this image can be calculated as the ratio of the total valid area of bounding boxes and the total area of the ROI.

C. DATA PROCESSING

The data generated from the previous steps are two sequences of data where each value corresponds to an image. However, due to network or website problems, the images at some timestamps are lost. In addition, an instant value is not stable for reflecting the traffic status over a time period. Therefore, the raw data is processed by the following operations.

For better reflecting the traffic status of a period, more samplings are needed for representing it, i.e., the traffic status of a period should be extracted from a short sequence of data. We use three consecutive data to represent the traffic status over a time period. For each of the three consecutive instant traffic volume and image-based traffic occupancy data, we calculate their mean as the newly generated dataset. If one data is missing, the mean of the left two data will be used. After data aggregation, the size of the data is approximately reduced to one-third. Note that in the newly generated datasets, the time interval is increased to 6 minutes, which can be changed by varying the original sampling interval. For the newly generated dataset from instant traffic volume, we will call it traffic volume directly in the following sections.

D. GRAPH CONSTRUCTION

For graph neural networks, a graph of data is necessary for capturing relations between nodes. A graph is input as an adjacency matrix which is a square matrix including the information of the graph. The size of an adjacency matrix is $N \times N$ where N is the number of nodes. The value located at (i, j) represents the weight or distance between the i-th and j-th nodes in the graph.

The graph construction is based on the locations of the cameras and the distances between them. In the related works,

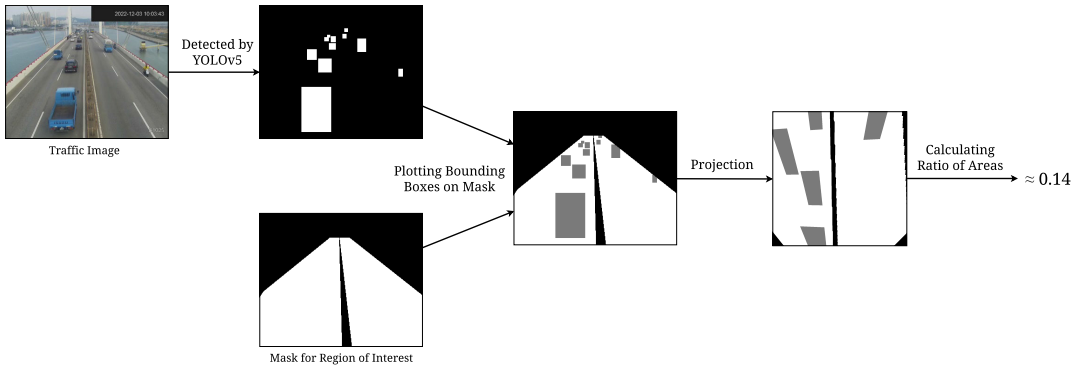


FIGURE 3. The process of calculating image-based traffic occupancy.

an edge between two nodes is determined as existing if their distance is less or equal to a threshold [24], [30]. However, unlike sensors, the densities of cameras are not equal in Macao Peninsula and Taipa. Only a threshold to determine the existence of edges will cause too many edges in the area with a high density of cameras and too few edges in the area with a low density of cameras. On the other hand, the threshold for distance cannot fully reflect the connections among the traffic network. The higher density of cameras usually means more intersections or traffic nodes in this area, which causes a lower average speed. Therefore, the node in the low-density area should be allowed to connect to other nodes at a farther distance.

In real-world traffic networks, the density of traffic nodes in a region changes moderately, i.e., the distances between nodes are usually close. Based on this assumption, we consider the blue node in Figure 4. Intuitively, the node should connect to the green nodes because the distances between the blue node and green nodes are similar and closer, which indicates they may be directly connected in the real world. The orange nodes are farther from the blue nodes but close to the green nodes, which indicates they are directly connected with the green nodes instead of the blue node. The right part of Figure 4 shows the distances between the blue node and other nodes. It is easy to find that the distances of the green nodes form a cluster, and it is the same for the orange nodes.

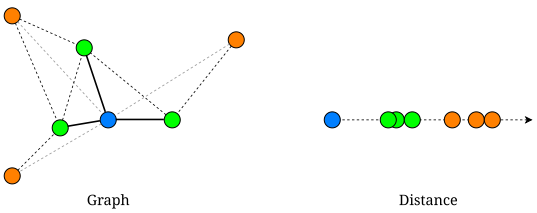


FIGURE 4. An example of a node in graph.

Therefore, a clustering algorithm can be used to group the nodes. The K-means algorithm is a clustering algorithm that divides data into K groups. It initially selects centroids for K clusters and assigns data to the closest cluster. Then it keeps updating the new centroids to minimize the distance between data and centroids until they converge. Based on the K-means algorithm, we design an algorithm to connect nodes to their closest cluster, which is shown in Algorithm 1. This algorithm takes 4 inputs: v_0 is the node to be connected with others, $\mathbf{v}$ is a list that contains all nodes except v_0, σ_{max} is a value that provides a base maximum dispersion of clusters by comparing with the standard deviation, n_t is a value of the expected number of edges, b is a value that controls the effect of n_t. The Kmeans function inside the algorithm applies the K-means algorithm to the distances between v_0 and other nodes and divides other nodes into i groups. The return of the Kmeans function is a nested list that contains lists of the result clusters.

Algorithm 1 Algorithm for Connecting Node to the Closest Cluster

```
procedure ConnectNodes(v_0, v, σ_max, n_t, b)
    for i in 1 to length of v do
        c ← Kmeans (v_0, v, i)
        for c_j in c do
            σ_j ← standard deviation of distances between v_0 and each v_k in c_j
            n_j ← size of c_j
            if σ_j > σ_max / b^(n_j − n_t) then
                Skip to the next iteration of i
            end if
        end for
        c_closest ← the c_j with the smallest mean of distances in c
        Connect v_0 with each v_close in c_closest
        return
    end for
end procedure
```

The idea of Algorithm 1 is to increase the number of clusters for the K-means algorithm until the dispersion of each cluster is less than a threshold. However, for the area with a high density of cameras, the acceptable dispersion should be smaller as the lengths of roads are shorter. Therefore, n_t and b are used to punish the node with too many edges and allow more dispersion for the nodes

with fewer edges by adjusting the acceptable dispersion. In this algorithm, the dispersion limitation is defined by the following equation,

$$\sigma_{lim} = \frac{\sigma_{max}}{b^{n_j - n_t}} \tag{3}$$

where n_j is the size of the current cluster. When b is larger, its effect on σ_{lim} is bigger. When $b = 1$, we have $\sigma_{lim} = \sigma_{max}$, which means the n_t is disabled.

For example, we manually labeled the locations of 49 cameras on the DSAT website. We set $\sigma_{max} = 1.0$, which means the base standard deviation of clusters is 1 kilometer. That allows the node to connect to farther nodes to ensure the connections on the bridges do not break. We set $n_t = 3, b = 1.15$ to reduce the number of connections. For comparison, we constructed another graph based on the locations of the cameras by thresholding the distance. We set the threshold distance to the minimum distance which ensures every node has at least one connection, which is 1.7 kilometers. Figure 5 shows the comparison of the two constructed graphs. The nodes are represented by red dots and the edges are represented by blue dashed lines. The graph threshold by distance shows a mess in the high-density area, whereas the graph constructed by the proposed method is much clearer. Figure 6 shows the comparison of the graph constructed by the proposed method and the real map. The edges are basically reasonable and follow the roads.

After the edges are determined, the weights of the edges should be calculated. We use a similar method as [24] which uses a Gaussian kernel. The weights of edges can be calculated as,

$$W_{ij} = \exp(-\frac{\text{dist}(v_i, v_j)^2}{2\sigma^2}) \tag{4}$$

where W_{ij} is the weight of node v_i and node v_j in the adjacency matrix, $\text{dist}(v_i, v_j)$ calculates the distance between v_i and v_j, σ is the standard deviation of distances.

IV. TRAFFIC FORECASTING AND CONGESTION DETECTION

In this section, the approaches for traffic forecasting and traffic congestion detection will be presented. Firstly, the models which will be evaluated for forecasting traffic volume are introduced. Then the method for detecting traffic congestion based on traffic volume and image-based traffic occupancy will be proposed. Lastly, the performance metrics for evaluating the models and the method will be introduced.

A. TRAFFIC FORECASTING

Time series forecasting uses the historical data sequence to forecast the future data sequence. For traffic forecasting, the graph structure of the traffic network can be also considered as the input. The traffic forecasting problem using the graph of the traffic network can be defined as,

$$[X_{(t-P)}, \cdots, X_t; \mathcal{G}] \xrightarrow{f} [X^*_{(t+1)}, \cdots, X^*_{(t+Q)}] \tag{5}$$

where X_t denotes the traffic signal at timestamp t, X^* is the predicted traffic signal, $\mathcal{G}$ is the graph structure, P is the length of the input sequence, and Q is the length of the predicted sequence. In this paper, the extracted dataset for traffic volume is used as the traffic signal for forecasting. We will evaluate several statistical models and state-of-the-art deep learning models on this dataset

B. TRAFFIC CONGESTION DETECTION

The traffic volume is a good measure for detecting congestion because it is relatively stable and accurate. However, the disadvantage is that the maximum capacities are different for different road segments. That is caused by the inconsistent lengths of the road segment captured by the camera and the angles of view of the camera. Therefore, each node needs a specific threshold to detect traffic congestion.

To determine the threshold that distinguishes congestion and non-congestion, image-based traffic occupancy can be used. One of the characteristics of the occupancy is that it is naturally between 0 and 1, which means it can be used as a consistent standard for all nodes. As we mentioned in Section II, a related measure V/C and its corresponding traffic status have been introduced by [2], where a V/C ratio over 0.80 indicates a high-density flow. Based on this standard, we use 0.80 of V/C as the threshold to distinguish between congestion and non-congestion.

The traffic occupancy can be modeled by,

$$\text{Occupancy} = \frac{Area_v}{Area_r} \tag{6}$$

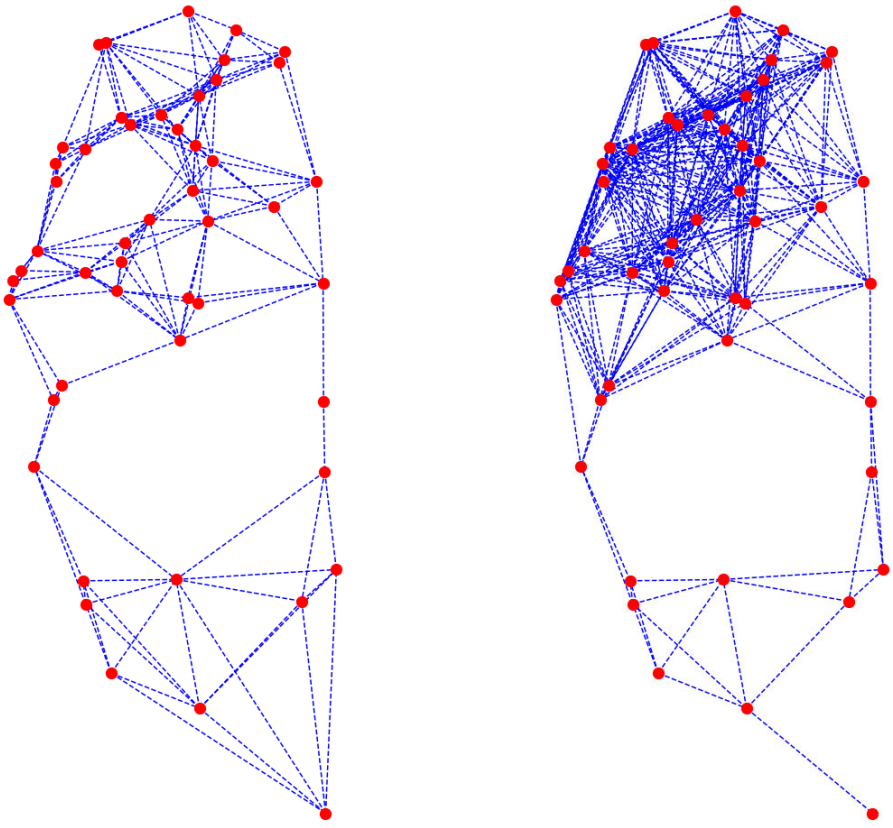
where $Area_v$ is the total area of vehicles, $Area_r$ is the area of road segment. Equation (6) can be expanded as,

$$\begin{aligned}\text{Occupancy} &= \frac{N_v \times L_v \times W_v}{N_l \times L_r \times W_l} \\ &= \frac{N_v \times (L_o - L_s) \times W_v}{N_l \times L_r \times W_l}\end{aligned} \tag{7}$$

where N_v is the number of vehicles on the road segment, L_v is the average length of vehicles, W_v is the average width of vehicles, N_l is the number of lanes, L_r is the length of the road segment, W_l is the width of a lane, L_o is the length occupied by vehicles including vehicle length and safety distance, L_s is the safety distance. According to [2], the vehicle length can be assumed as 4.27 m, and the safety distance is about 4.57 m. By observation, the ratio of the width of a vehicle and a road lane $\frac{W_v}{W_l}$ is approximate $\frac{4}{7}$. Based on (1) and (2), the occupancy can be further calculated as follows,

$$\begin{aligned}\text{Occupancy} &= \frac{14}{14+15} \times \frac{N_v \times L_o}{N_l \times L_r} \times \frac{4}{7} \\ &\approx 0.28 \text{ V/C}.\end{aligned} \tag{8}$$

When V/C > 0.8, we have Occupancy > 0.22. Hence, the threshold determining congestion is 0.22. Although image-based traffic occupancy is relatively unstable, it can reflect the approximate distributions of ground truth. Based on this, the occupancy can be used as a label that indicates



(a) Constructed graph for 49 cameras using proposed method

(b) Graph for 49 cameras threshold by distance

FIGURE 5. Comparing the graphs constructed by proposed method and distance threshold.

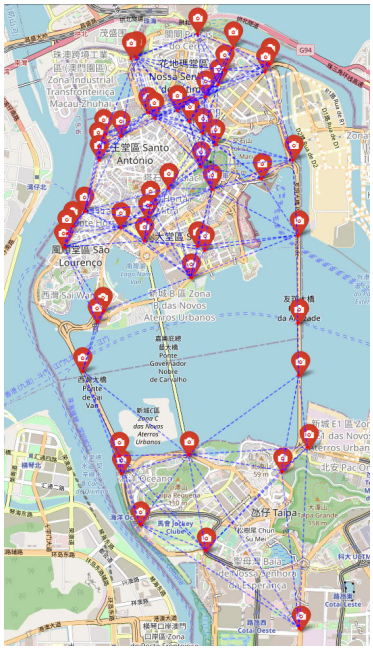


FIGURE 6. Comparing constructed graph with the real map.

whether the current timestamp is encountering traffic congestion or not.

With the labeled timestamp, the model for determining congestion through traffic volume can be developed. We use logistic regression to model the relationship between traffic congestion and traffic volume. The probability of encountering traffic congestion can be represented by,

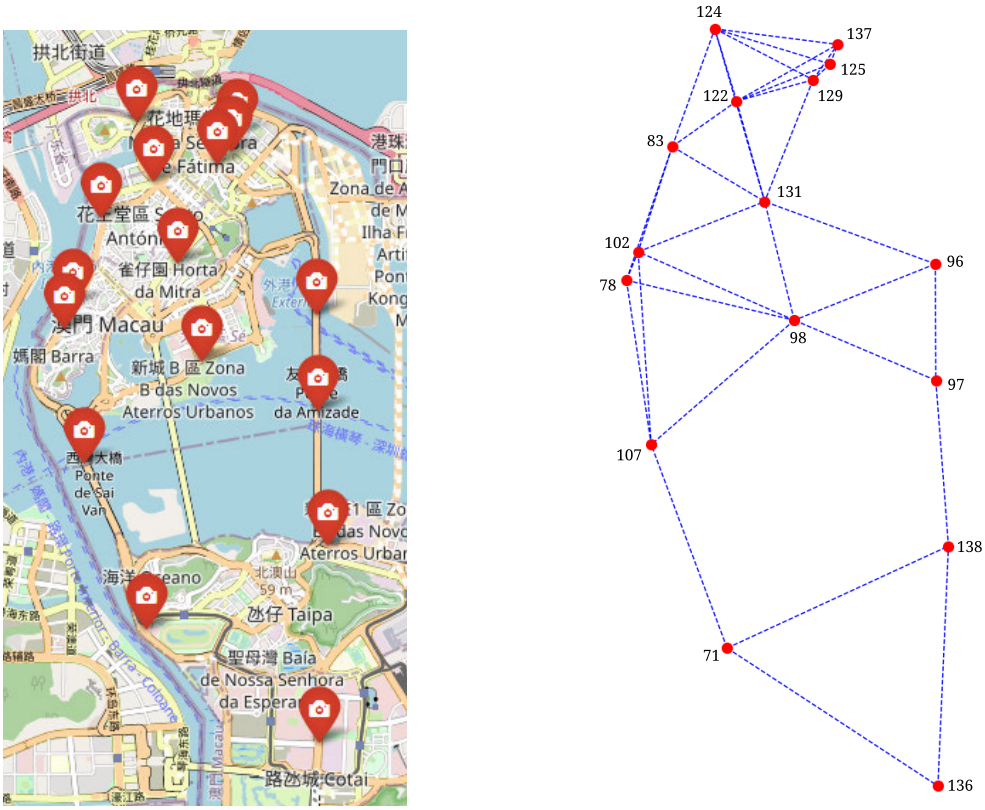
$$p(v) = P(\text{Congested}|v) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 v)}} \tag{9}$$

where v denotes the traffic volume, θ_0 and θ_1 are the trainable parameters. Because the number of congested samples is much less than not congested samples, the weight of the congested class needs to be balanced. Therefore, the weighted loss function can be calculated as follows,

$$C(p(\boldsymbol{v}), \boldsymbol{y}) = -\sum_{i=1}^{n} (\frac{n_1}{n_0} y_i \ln p(v_i) + (1 - y_i) \ln(1 - p(v_i))) \tag{10}$$

where n_1 is the number of congested samples, n_0 is the number of not congested samples, y_i is the label of the i -th sample.

Due to the instability of the image-based traffic occupancy, the timestamp labeled as congested should contain several false positive samples that are not congested but labeled as congested. To determine traffic congestion more accurately,



(a) Locations of the 16 cameras

(b) Graph of the 16 cameras

FIGURE 7. The locations of the 16 cameras and the constructed graph.

we set a threshold that is greater than 0.5 as the probability threshold of the congestion, which can be expressed as,

$$\text{Congested}(v) = \begin{cases} 1 & p(v) \geq \alpha \\ 0 & p(v) < \alpha \end{cases} \tag{11}$$

where α denotes the threshold, 1 indicates the traffic status is congested, and 0 indicates the traffic status is not congested.

After the model is fitted to the data, a threshold for traffic volume that can determine the traffic congestion directly can be extracted. When the probability of congestion is α, it can be expressed as,

$$\frac{1}{1 + e^{-(\theta_0 + \theta_1 v^*)}} = \alpha \tag{12}$$

which can be transformed to,

$$v^* = -\frac{\ln(\frac{1}{\alpha} - 1) + \theta_0}{\theta_1}. \tag{13}$$

Therefore, v^* can be used as the threshold.

C. PERFORMANCE METRICS

In the following metrics, y_i denotes the i -th ground truth, y_i^* denotes the i -th predicted value, and n is the number of forecasting time steps.

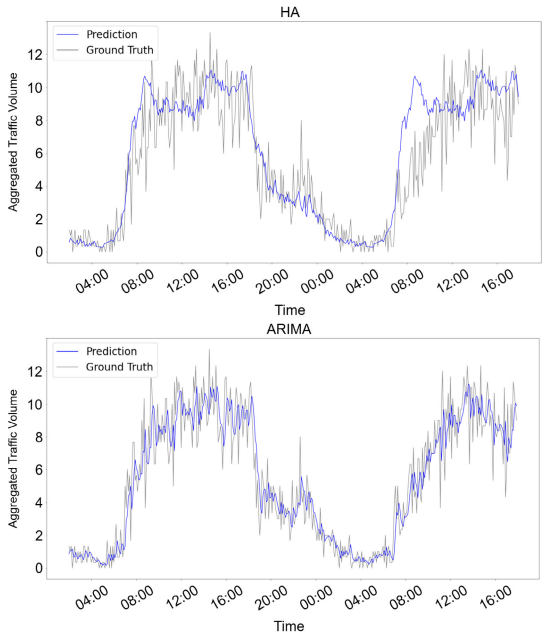


FIGURE 8. Examples of prediction of statistical models for Cam#96.

Mean Absolute Error (MAE) measures the average absolute error in a set of predictions. Its formula can be

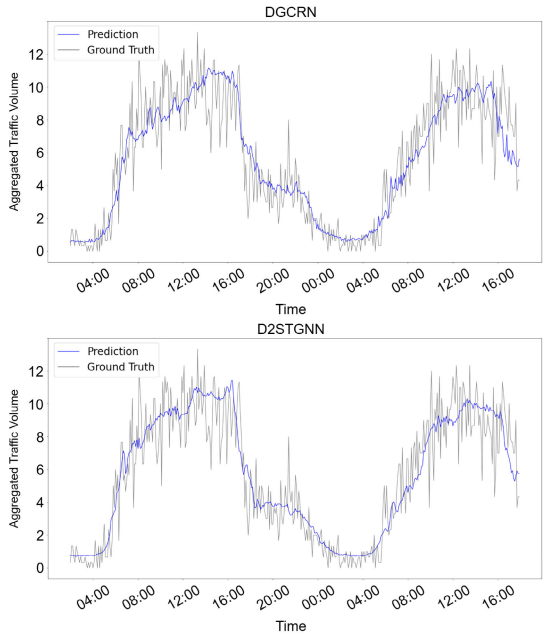


FIGURE 9. Examples of prediction of deep learning models for Cam#96.

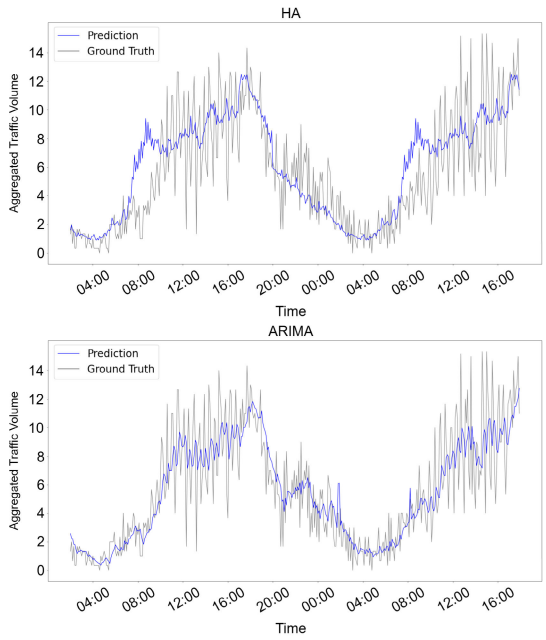


FIGURE 10. Examples of prediction of statistical models for Cam#98.

expressed as,

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i^* - y_i|. \tag{14}$$

Root Mean Square Error (RMSE) measures the average square root error in the predictions. Compared to MAE, it gives more weight to large errors. It can be expressed as,

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i^* - y_i)^2}. \tag{15}$$

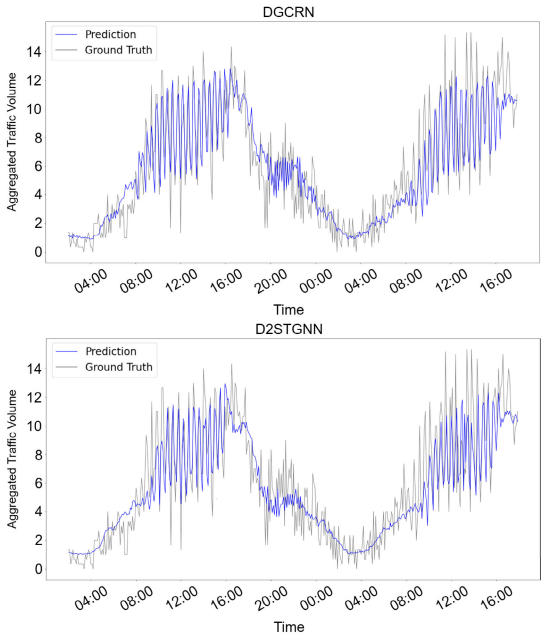


FIGURE 11. Examples of prediction of deep learning models for Cam#98.

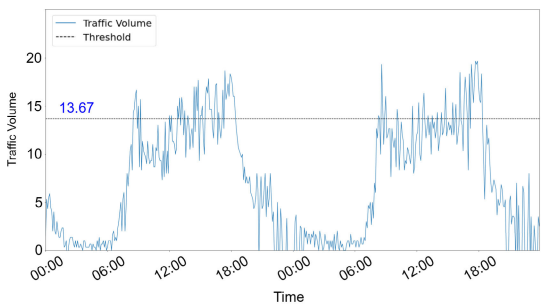


FIGURE 12. An example of the calculated threshold for Cam#107.

Mean Absolute Percentage Error (MAPE) measures the average percentage error in the predictions. It is a relative measure and is useful when the magnitude of the errors is important. It can be expressed as,

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i^* - y_i}{y_i}\right|. \tag{16}$$

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

We use the data set collected in Macao to demonstrate the effectiveness of the proposed framework, which is applicable to images collected from multiple monitoring cameras where traffic is correlated.

A. TRAFFIC FORECASTING

1) GRAPH CONSTRUCTION

Based on the image quality and camera pose, the locations of 16 chosen cameras are manually labeled and the graph is constructed by the proposed method. The locations of the cameras are shown in Figure 7. The cameras are basically

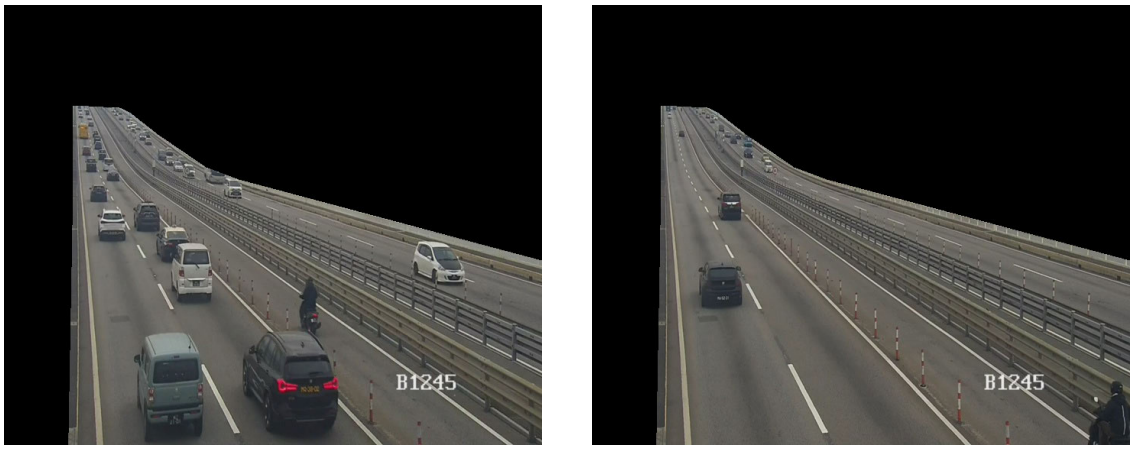


FIGURE 13. A comparison of congested and not congested images.

spread throughout most of the area in Macao. To construct the graph, we set $\sigma_{max} = 0.5, n_t = 4, b = 1.3$ as we expect that a node has around 4 neighbors, and their distances from the node differ no more than 0.5 kilometers. The constructed graph and their identification numbers are shown in Figure 7b. The red points indicate the locations of the cameras and the blue dashed lines are the edges.

2) EXPERIMENTAL SETTINGS

We conducted the experiments through 2 statistical models Historical Average (HA), ARIMA, and 2 state-of-the-art deep learning models: DGCRN [26] and D2STGNN [28]. All experiments are implemented through Python. The ARIMA model is implemented based on the *Statsmodels* [31] library. The deep learning models are implemented based on *PyTorch* [32] and the *BasicTS* [33] project. The collection script is implemented in Python. The forecasting time step is set to 12. The dataset is divided into training, validation, and testing sets, with a ratio of 7:1:2. For the deep learning models, the input is the previous 12 steps of data and the adjacency matrix of the graph of the traffic network. The time information, such as the time of day and the day of the week, is also used as input. The collection started on November 30th, 2022, and ended on February 1st, 2023. Totally 567,012 images were collected from the 16 cameras. After removing the data sequences containing unavailable data, the lengths of training, validation, and testing sets are 6661, 951, and 1904, respectively, where each element is a 16×12 matrix.

3) PERFORMANCE COMPARISON

Table 1 shows the performance of the models. The D2STGNN model achieves the best performance in almost all performance metrics. The performance of deep learning models is better than the performance of statistical models.

Figure 8 and Figure 9 show the illustrations of sequences of prediction from statistical models and deep learning models, respectively. In these two figures, the forecasting time step is 6, and the identification number of the camera holding this sequence is 96. The blue line in each figure is the predicted sequence and the gray line is the ground truth. The deep learning models can basically produce reasonable predictions. The ARIMA model shows a slight trend of replicating the past value, so the result seems to lag a little behind the ground truth. The HA model has a significant difference from the ground truth at 08:00 on January 22nd, which is because it does not consider the short-term data.

Figure 10 and Figure 11 show the illustrations of predictions for another camera with an identification number of 98. The difference between camera 98 and camera 96 is that camera 98 is monitoring a road intersection. It is obvious that strong but regular shaking occurred from 8:00 to 16:00 on January 21st, which was caused by the traffic light. The statistical models cannot handle this sequence properly, but the deep learning models basically made successful predictions.

TABLE 1. The performance of evaluated models.

Methods	MAE	RMSE	MAPE	MAE	RMSE	MAPE
	Horizon 3 (18 minutes)			Horizon 6 (36 minutes)		
HA	1.92	2.67	36.44%	1.92	2.67	36.44%
ARIMA	1.41	1.94	30.83%	1.56	2.17	33.99%
DGCRN	1.35	**1.82**	**28.48%**	1.47	2.00	30.74%
D2STGNN	**1.34**	**1.82**	29.59%	**1.44**	**1.97**	**30.68%**
	Horizon 12 (72 minutes)			Overall		
HA	1.92	2.67	36.44%	1.92	2.67	36.44%
ARIMA	1.83	2.55	41.03%	1.57	2.19	34.51%
DGCRN	1.68	2.28	36.28%	1.47	2.00	31.24%
D2STGNN	**1.57**	**2.14**	**32.56%**	**1.44**	**1.96**	**31.13%**

B. TRAFFIC CONGESTION DETECTION

1) EXPERIMENTAL SETTINGS

The experiment is implemented through Python. The logistic regression model is implemented using *scikit-learn* [34] library. The L_2 regularization is used in the model. The L-BFGS [35] is used to optimize the model. For each node, the probability threshold is set to 0.7.

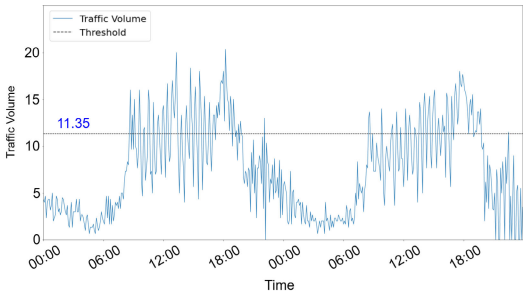


FIGURE 14. An example of the calculated threshold for Cam#98.

2) EXPERIMENTAL RESULTS

Figure 12 shows the calculated threshold for the camera monitoring one of the bridges in Macao. and an example of the traffic volume sequence from January 26, 2023, to January 27, 2023. The blue curve is the traffic volume and the black dashed line is the threshold of 13.67. The threshold reasonably divides the curve into two parts.

Figure 13 shows two masked images on January 26, 2023, where the image on the left is captured at 17:26 which is around the peak before 18:00 on January 26, and the image on the right is captured at 11:02 which is after the peak at 10:00 on January 26. It is obvious that the traffic congestion happened in the first image, and the traffic flow is free in the second image. The threshold can successfully distinguish congested and not congested traffic status.

However, for the camera capturing road segments with a traffic light, the traffic volume cannot distinguish the real traffic congestion and congested traffic flow caused by red light. Figure 14 illustrates an example of the data sequence of Cam#98 with its calculated threshold. In the illustration, the threshold crosses from a sequence of continuous peaks and valleys of the curve which is caused by the traffic light turning. Due to the static data and the relatively long interval, it is difficult to handle this situation.

VI. CONCLUSION

This paper has proposed a low-cost graph-based framework for traffic forecasting and congestion detection by converting online images to numerical values. A set of images was collected from the DSAT website to construct a dataset to demonstrate the effectiveness of the proposed framework. Two types of data: traffic volume and image-based traffic occupancy are introduced and extracted from the images. A graph construction method based on distance clustering was proposed and used to build a graph from the locations of nodes in the traffic network. To evaluate the effectiveness of the proposed framework, we evaluated 4 models, including historical average, ARIMA, DGCRN, and D2STGNN, on the extracted traffic volume. It was found that the D2STGNN achieved the best performance in traffic forecasting. A threshold-based method for detecting traffic congestion using the two types of extracted data was proposed. The proposed unique framework provides a solution for traffic forecasting and congestion detection when only surveillance images from multiple cameras are available.

REFERENCES

[1] V. Jain, A. Sharma, and L. Subramanian, "Road traffic congestion in the developing world," in *Proc. 2nd ACM Symp. Comput. Develop.*, Mar. 2012, pp. 1–10.

[2] T. Afrin and N. Yodo, "A survey of road traffic congestion measures towards a sustainable and resilient transportation system," *Sustainability*, vol. 12, no. 11, p. 4660, Jun. 2020.

[3] A. M. Nagy and V. Simon, "Survey on traffic prediction in smart cities," *Pervas. Mobile Comput.*, vol. 50, pp. 148–163, Oct. 2018.

[4] W. Jiang and J. Luo, "Graph neural network for traffic forecasting: A survey," *Expert Syst. Appl.*, vol. 207, Nov. 2022, Art. no. 117921.

[5] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep learning: A generic approach for extreme condition traffic forecasting," in *Proc. SIAM Int. Conf. Data Mining*, Jun. 2017, pp. 777–785.

[6] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: A deep learning approach for short-term traffic forecast," *IET Intell. Transp. Syst.*, vol. 11, no. 2, pp. 68–75, Mar. 2017.

[7] C.-T. Lam, H. Gao, and B. Ng, "A real-time traffic congestion detection system using on-line images," in *Proc. IEEE 17th Int. Conf. Commun. Technol. (ICCT)*, Oct. 2017, pp. 1548–1552.

[8] C.-T. Lam, B. Ng, and C.-W. Chan, "Real-time traffic status detection from on-line images using generic object detection system with deep learning," in *Proc. IEEE 19th Int. Conf. Commun. Technol. (ICCT)*, Oct. 2019, pp. 1506–1510.

[9] B. Liu, C.-T. Lam, and B. K. Ng, "Improved real-time traffic congestion detection with automatic image cropping using online camera images," in *Proc. IEEE 21st Int. Conf. Commun. Technol. (ICCT)*, Oct. 2021, pp. 1117–1122.

[10] P. Chakraborty, Y. O. Adu-Gyamfi, S. Poddar, V. Ahsani, A. Sharma, and S. Sarkar, "Traffic congestion detection from camera images using deep convolution neural networks," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2672, no. 45, pp. 222–231, Dec. 2018.

[11] L. Wang, C. T. Lam, K. E. Law, B. Ng, W. Ke, and M. Im, "Real-time traffic monitoring and status detection with a multi-vehicle tracking system," in *Proc. Int. Conf. Intell. Transp. Syst.* Cham, Switzerland: Springer, Nov. 2021, pp. 13–25.

[12] DSAT. *Instant Traffic Status*. Accessed: Mar. 20, 2023. [Online]. Available: http://www.dsat.gov.mo/dsat/realtime.aspx

[13] F. L. Hall, "*Traffic Stream Characteristics*. vol. 36, Washington, DC, USA: Traffic Flow Theory U.S. Federal Highway Administration, 1996.

[14] Z. Sun, W. Gu, J. Feng, and X. Zhu, "Threshold value based traffic congestion identification method," in *Proc. Int. Conf. Data Sci. (ICDATA) Steering Committee World Congr. Comput. Sci. Comput.*, Jul. 2011, p. 1–7.

[15] Q. Wang, J. Wan, and Y. Yuan, "Locality constraint distance metric learning for traffic congestion detection," *Pattern Recognit.*, vol. 75, pp. 272–281, Mar. 2018.

[16] J. Kurniawan, S. G. S. Syahra, C. K. Dewa, and Afiahayati, "Traffic congestion detection: Learning from CCTV monitoring images using convolutional neural network," *Proc. Comput. Sci.*, vol. 144, pp. 291–297, Jan. 2018.

[17] X. Ke, L. Shi, W. Guo, and D. Chen, "Multi-dimensional traffic congestion detection based on fusion of visual features and convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2157–2170, Jun. 2019.

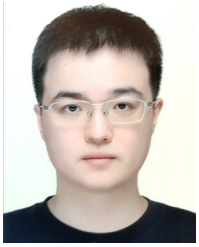
[18] J. Cho, H. Yi, H. Jung, and K.-H.-N. Bui, "An image generation approach for traffic density classification at large-scale road network," *J. Inf. Telecommun.*, vol. 5, no. 3, pp. 296–309, Jul. 2021.

[19] C. K. Moorthy and B. G. Ratcliffe, "Short term traffic forecasting using time series methods," *Transp. Planning Technol.*, vol. 12, no. 1, pp. 45–56, Jul. 1988.

[20] S. Lee and D. B. Fambro, "Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1678, no. 1, pp. 179–188, Jan. 1999.

[21] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 871–882, Jun. 2013.

[22] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, Apr. 2017.
[23] M. Chen, X. Yu, and Y. Liu, "PCNN: Deep convolutional networks for short-term traffic congestion prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 11, pp. 3550–3559, Nov. 2018.
[24] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2017, *arXiv:1707.01926*.
[25] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," 2017, *arXiv:1709.04875*.
[26] F. Li, J. Feng, H. Yan, G. Jin, F. Yang, F. Sun, D. Jin, and Y. Li, "Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution," *ACM Trans. Knowl. Discovery Data*, vol. 17, no. 1, pp. 1–21, Feb. 2023.
[27] J. Jin, D. Rong, T. Zhang, Q. Ji, H. Guo, Y. Lv, X. Ma, and F.-Y. Wang, "A GAN-based short-term link traffic prediction approach for urban road networks under a parallel learning framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16185–16196, Sep. 2022.
[28] Z. Shao, Z. Zhang, W. Wei, F. Wang, Y. Xu, X. Cao, and C. S. Jensen, "Decoupled dynamic spatial–temporal graph neural network for traffic forecasting," 2022, *arXiv:2206.09112*.
[29] G. Jocher. (2023). *Ultralytics/YOLOV5*. [Online]. Available: https://github.com/ultralytics/yolov5
[30] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.
[31] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with Python," in *Proc. Python Sci. Conf.*, 2010, pp. 10–25.
[32] A. Paszke, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Dec. 2019, pp. 8026–8037.
[33] Z. Shao. (2022). *BasicTS: A Time Series Benchmark and Toolkit*. [Online]. Available: https://github.com/zezhishao/BasicTS
[34] F. Pedregosa, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Jan. 2011.
[35] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.*, vol. 45, nos. 1–3, pp. 503–528, Aug. 1989.



BOWIE LIU received the B.S. degree in computer science and the M.S. degree in big data and IoTs from Macao Polytechnic University, in 2021 and 2023, respectively, where he is currently pursuing the Ph.D. degree in computer applied technology with the Faculty of Applied Sciences. His research interests include graph neural networks, time series forecasting, smart cities, and gray codes.

CHAN-TONG LAM (Senior Member, IEEE) received the B.Sc. (Eng.) and M.Sc. (Eng.) degrees from Queen's University, Kingston, ON, Canada, in 1998 and 2000, respectively, and the Ph.D. degree from Carleton University, Ottawa, ON, in 2007. He is currently an Associate Professor with the Faculty of Applied Sciences, Macao Polytechnic University, Macau, China. From 2004 to 2007, he participated in the European Wireless World Initiative New Radio (WINNER) Project. He has published more than 100 publications in refereed journals and conferences. His research interests include mobile wireless communications, digital signal processing, machine learning in communications, and computer vision in smart cities. He is a member of the IEEE Communications Society.

BENJAMIN K. NG (Senior Member, IEEE) received the B.A.Sc., M.A.Sc., and Ph.D. degrees in engineering science and electrical engineering from the University of Toronto, in 1996, 1998, and 2002, respectively. From 2005 to 2009, he was with Radiospire Networks Inc., Boston, MA, USA, where he was a Senior Communications Engineer, focusing on UWB and millimeter wave technologies. He joined Macao Polytechnic University, Macau, China, in 2010, where he is currently an Associate Professor with the Faculty of Applied Sciences. His research interests include wireless communications and signal processing, with an emphasis on MIMO, NOMA, and machine learning technologies.

XIAOCHEN YUAN (Senior Member, IEEE) received the B.Sc. degree in electronic information technology from the Macau University of Science and Technology, in 2008, and the M.Sc. degree in e-commerce technology and the Ph.D. degree in software engineering from the University of Macau, in 2010 and 2013, respectively. From 2014 to 2015, she was a Postdoctoral Fellow with the Department of Computer and Information Science, University of Macau. From 2016 to 2021, she was an Assistant Professor and an Associate Professor with the Faculty of Information Technology, Macau University of Science and Technology. She is currently an Associate Professor with the Faculty of Applied Sciences, Macao Polytechnic University. Her research interests include digital multimedia processing, digital watermarking, multimedia forensics, tampering detection and self-recovery, acoustic processing and diagnosis, and deep learning techniques and applications.

SIO KEI IM (Member, IEEE) received the degree in computer science and the master's degree in enterprise information systems from the King's College, University of London, U.K., in 1998 and 1999, respectively, and the Ph.D. degree in electronic engineering from the Queen Mary University of London (QMUL), U.K., in 2007. He gained the position of a Lecturer with the Computing Program, Macao Polytechnic Institute (MPI), in 2001. In 2005, he became the Operations Manager of MPI-QMUL Information Systems Research Center jointly operated by MPI and QMUL, where he carried out signal processing work. He was promoted to a Professor with the Macao Polytechnic Institute, in 2015. He was a Visiting Scholar with the School of Engineering, University of California at Los Angeles (UCLA), and an Honorary Professor with the Open University of Hong Kong.