Received 9 March 2023; revised 6 June 2023 and 16 September 2023; accepted 7 October 2023. Date of current version 11 April 2024.
Digital Object Identifier 10.1109/OJCAS.2023.3326274

A Computer Vision-Based Framework for Snow Removal Operation Routing

MOHAMED KARAA[1] (Student Member, IEEE), HAKIM GHAZZAI[2] (Senior Member, IEEE), YEHIA MASSOUD[2] (Fellow, IEEE), AND LOKMAN SBOUI[1] (Senior Member, IEEE)

[1]Systems Engineering Department, Ecole de Technologie Supérieure, Montreal, QC H3C 1K3, Canada
[2]Innovative Technologies Laboratories, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia

This article was recommended by Associate Editor Y. Du.

CORRESPONDING AUTHOR: H. GHAZZAI (e-mail: hakim.ghazzai@kaust.edu.sa)

ABSTRACT During snowfall, the utility of the road infrastructure is critical. Roads must be effectively cleared to ensure access to important locations and services. In this paper, we present an end-to-end framework for snow removal vehicle routing based on road priority. We offer an artificial intelligence-based image-based approach for estimating snow depth and traffic volume on roads. For segments monitored by CCTV cameras, we exploit images and supervised learning models to perform this task. For unmonitored roads, we use the Graph Convolutional Network architecture to predict parameters in a semi-supervised manner. Following that, we assign priority weights to all graph edges as a function of image-based attributes and road categories. We test the method using a real-world example, simulating snow removal within a study area in Montreal, Quebec, Canada. As input for the framework, we collect CCTV image data and combine it with a 2D map. As a result, more efficient snow removal operation can be achieved by optimizing the trajectories of trucks based on the computer vision module outputs.

INDEX TERMS Snow removal, road service prioritization, graph analytics, computer vision, vehicle routing.

I. INTRODUCTION

During the winter season, many areas around the world witness heavy snowfall. Cities face serious challenges maintaining smooth access to the road network in such adverse weather conditions. The accumulation of snow on the roads affects vehicle mobility, causes traffic congestion, and alters the daily course of life [1], [2]. Furthermore, failing to remove snow actively prevents access to vital services and public transport. On the other hand, snowfall can cause a threat to road users' safety, especially when accompanied by ice, black ice, and rain. These occurrences make the road surface slippery and increase the chances of a vehicle collision and skidding. The Canadian National Collision Database reports that during the last 20 years, 20% of car accidents happen on snowy or icy roads. The U.S. Federal Highway Administration records over 1300 fatalities in vehicle crashes during snowfall annually. In addition, over the past few years, snowstorms have become less frequent but more severe and unpredictable due to climate change. These facts emphasize the importance of snow removal to keep the road network accessible and safe.

However, multiple constraints imposed on snow removal make it a very challenging task. Snow removal operations require winter service vehicle fleets, de-icing chemicals, and fuel with the growing demand. Hence, government institutions need to scale budgets to accommodate increasing costs. Time also represents a crucial constraint for planning snow removal. Active and on-time operations execution helps the road network recover faster. Additionally, operations must be completed before more snow falls to prevent accumulation as unattended roads might become more difficult to plow later. Other problems include difficult access to certain roads, traffic congestion, and prohibited vehicle parking. The coexistence of these challenges within large urban areas makes snow removal management and planning more complex. For this reason, implementing intelligent decision-making systems is crucial. These solutions can exploit the big data produced by different connected IoT devices (such as sensors, cameras, etc.) to acquire knowledge about road status and weather conditions. Artificial Intelligence (AI) algorithms can be also incorporated to analyze data and plan snow removal operations efficiently. Therefore, in this study, we propose a novel framework for snow removal vehicle routing in an urban road network. This framework leverages closed-circuit television (CCTV) cameras and deep learning methods to measure snow levels covering roads and compute road priorities. The framework aims to recommend an optimal route for snow removal trucks to plow all roads based on road priorities.

Measuring snow level covering is part of active research on the estimation of winter road surface condition (RSC). Recent studies used different measurement tools and approaches including road weather information systems, IoT sensors, and traffic cameras. In [3], Shibara et al. designed a V2X sensor platform that uses weather, infrared, and motion sensors to estimate the road surface cover including snow and ice. However, with the growing number of cameras installed in road networks, multiple deep learning models are trained using image datasets produced by these cameras. Zhang et al. [4] introduced a visible and infrared image fusion method for the road snow cover classification. A recent work studied the use of webcams installed on interstate roads to estimate weather and surface conditions [5]. They identified three weather conditions (clear, light snowfall, and heavy snowfall) and three surface covers (dry, wet, and snowy). Grabowski and Czyżewski [6] used sensor measurements to annotate an image dataset and trained a convolutional neural network (CNN) that achieved 98.34% accuracy in winter road cover classification. In [7], the authors proposed a LiDAR-based approach to estimate the snow level covering the road using point cloud data, hence allowing coverage of wide-area ranges. Pan et al. [8] trained a model for to classify road snow cover and implemented an online model update strategy to optimize the model with new images.

Winter service vehicle routing is a challenging problem. Recent work on this subject includes various approaches to road network modeling, relevant parameters selection, objective functions design, and routing algorithms implementation. In [9], the author addressed the problem of determining the optimal path for one vehicle while considering the number of sweeps necessary to clear the road. The road network is modeled as a graph with directed and undirected edges. The author proposed an asymmetric traveling salesman problem approach with the objective function to minimize traveling time. In [10], Soloveva et al. proposed a solution that optimizes operation costs while penalizing uncleaned roads based on their vehicle flow capacity. Another approach proposed by [11] aimed to optimize the snow removal vehicle path by reducing the deadhead travel, i.e., when the vehicle travels but does not remove snow. The authors considered different objective functions to optimize cost, completion time, and total working time.

Unlike the aforementioned studies that focus on minimizing operation time and cost, the authors of [12] proposed another approach for optimizing urban snow removal. They treated the snow removal planning problem from a resilience perspective. The aim of this study is to plan operations such as the road network recovery is optimized, i.e., the road service is restored quickly after the snowfall. The first is the resource location-allocation problem, which aims to optimize the location of winter service resource centers. Based on these locations, the vehicle routing problem (VRP) is studied with the constraint of attaining a resilience value representing the road service capability. In [13], Rasul et al. combined the Chinese Postman Problem and Tabu search algorithm to optimize the snowplowing routes. In another work, Jin et al. [14] proposed a vulnerability-based approach to urban snow plowing optimization. They applied a rule to prioritize critical roads, which can negatively affect the road capacity. They modeled the snow plow problem as a capacitated arc routing problem for which they proposed a heuristic algorithm that minimizes the operation completion time. In [15], the authors proposed a snowplow route optimization framework that takes as input real-time traffic demand in order to find dynamic optimal routes for plowing trucks that minimize travel time for road users and operation completion time. While all these works use 2D maps to represent road networks, Park et al. [16] opted to use a 3D model for the road network. This model considered the road slope, which is an important factor that affects vehicle travel time and cost. For the VRP, the authors designed a genetic algorithm to minimize the objective function, aiming to reduce the total length of traveled roads while clearing priority roads.

Most existing works on snow removal VRP use existing databases or generated data to approach the problem. In addition, real-world applications still rely on manual inspection or simple measurement tools to determine snow-covered streets. Although there are several works on estimating snow levels, no research has taken advantage of cameras to assist in routing snow removal vehicles. In this work, we intend to exploit CCTV cameras to estimate snow levels and compute road priorities for snow removal. We employ deep learning algorithms to process the video frames collected from deployed CCTV infrastructure. In addition, we develop graph analysis methods to generate a priority-weighted graph representing the road network of the geographical area of interest.

The contributions of this paper can be summarized as follows:

- We develop a computer vision-based framework for modeling the road network in an urban snow removal context.
- We leverage CCTV image data and deep learning algorithms to estimate snow levels covering the roads and traffic volume in supervised and semi-supervised settings.

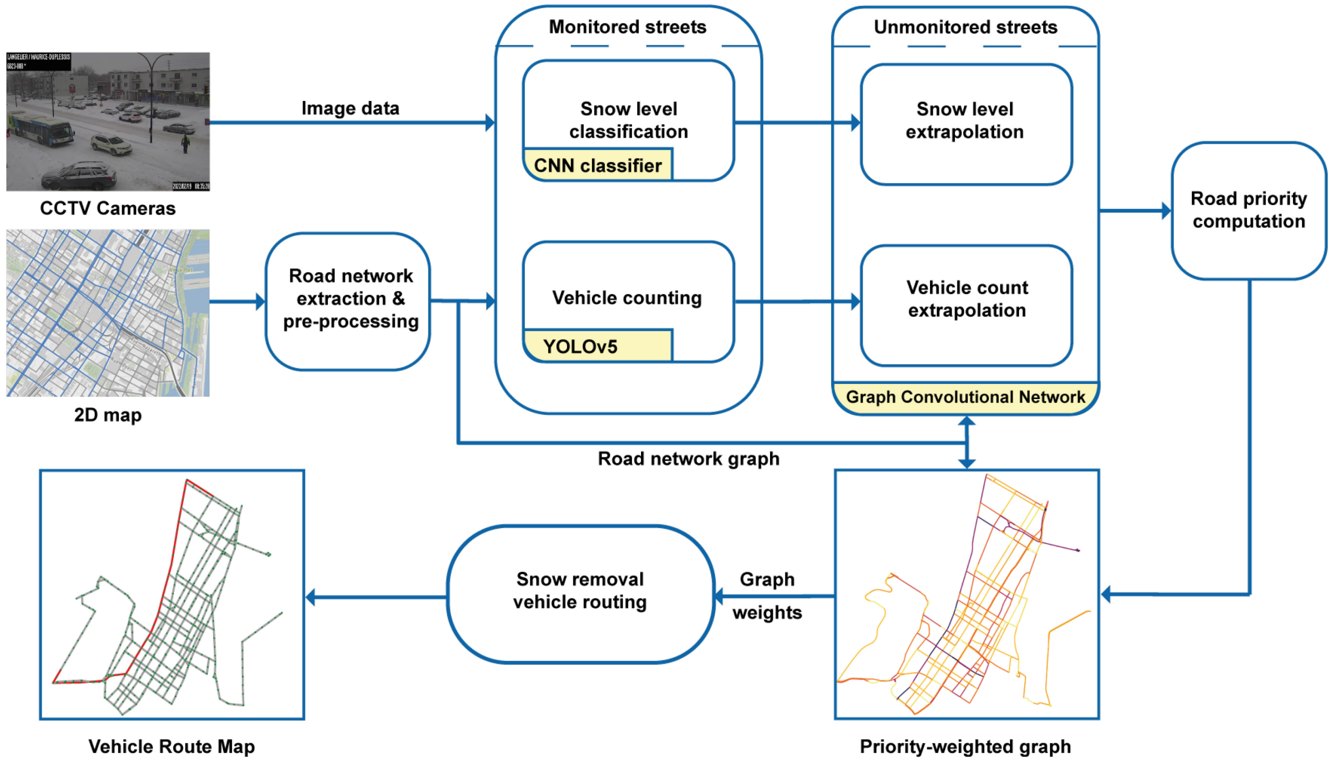


FIGURE 1. Overview of the snow removal vehicle routing framework using CCTV camera images for road priority computation.

- Based on the inputs of the computer vision modules, we design a graph-based approach to represent the road network and compute road priority weights.
- We validate our proposed framework for routing snow removal trucks in a real-world scenario and practical use-cases.

The remainder of this paper is organized as follows. We introduce the components of the AI-based snow removal vehicle routing framework in Section II. In Section III, we model the road network with a weighted graph structure, leverage deep learning algorithms to compute the weights, and discuss the snow removal vehicle routing problem. In Section IV, we present a case study to validate the proposed framework. Finally, we conclude the article in Section V and discuss potential future work.

II. METHODOLOGY

We feed CCTV camera images and a 2D map of the road network as input data to the framework. We develop AI and graph-based methods for data processing to compute road priorities. These priorities, expressed as graph weights, are used for the snow removal VRP. In global, the framework comprises five blocks as illustrated in Fig. 1. In the first block, we pre-process the 2D input map to extract a graph structure representing roads and intersections. We identify intersections equipped with CCTV cameras and perform some graph simplification operations such as removing private roads, duplicating edges to model bi-directional roads, and splitting long roads into small segments. The second block represents the computer vision component of the framework. In this block, we leverage the CCTV cameras to estimate snow level and traffic volume on monitored roads using supervised deep learning algorithms. We exploit an EfficientNet CNN for classification and a YOLOv5 object detector to achieve this task parallelly. The third block features a graph-based approach to estimate snow level and traffic volume on unmonitored road segments. We exploit the output of the previous blocks to predict the attributes for each road covered by a CCTV camera. Furthermore, we train a Graph Convolutional Network (GCN) in a semi-supervised way to extrapolate the missing attributes.

Once we estimate the snow level and traffic volume for all roads, we compute the weight for each graph edge in the fourth block. These weights indicate the importance of roads in the snow removal operation, i.e., their priority. The weights are a function of the estimated attributes and the network topology. The final block is for vehicle routing. Based on the extracted weights, we determine a path for a snow removal vehicle from a source point to a destination point. The objective is to plow high-priority roads first as fast as possible.

In summary, starting from a 2D map and video frames captured by CCTV cameras, we generate a fast routing strategy for snow removal truck to plow high-priority roads. Computer vision modules, a GCN, and a VRP algorithm are sequentially designed to achieve these objectives.

III. ROAD WEIGHT COMPUTATION AND ROUTING OPTIMIZATION

In this section, we discuss the different steps to generate a weighted road network graph for a given geographical area. Specifically, we present the different computer vision modules to extract relevant parameters from images and estimate road priority weights.

A. ROAD NETWORK GRAPH EXTRACTION

We model the road network as a graph, denoted by $G(V, E)$, where V is the set of vertices (nodes) and E is the set of road segments. A vertex $v \in V$ initially corresponds to the intersection of multiple streets. An edge $e \in E$ corresponds to a road connecting two nodes. For this work, we assume that the graph is undirected. We use CCTV cameras installed in different locations in the road network to estimate snow levels and traffic volumes in order to create edge weights for snow removal priorities. For each vertex v, we assign a boolean attribute indicating the presence of a CCTV camera. We assume that each camera has a maximum coverage distance d. For each edge e, we assign an attribute indicating which camera monitors this road. If the road length $l > d$, we create new intermediate vertices and divide this road into equal length segments ($l_{inter} = d$), except the last segment $l_{last} = l \bmod d$. We add the new vertices and edges to V and E, respectively. In the next three subsections, we leverage three deep learning models to assign weights for all graph edges. These weights will be used for the snow removal VRP.

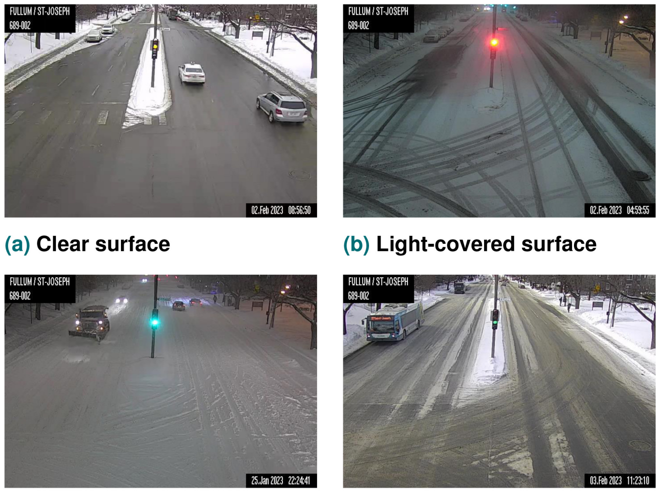
B. SNOW LEVEL CLASSIFICATION

We use CCTV camera feeds to determine the level of snow covering the roads. In this part, we are solely interested in the classification of snow levels on camera-monitored roads. We assume that the snow level on all roads connected to a single node is the same. We perform the snow level classification for all cameras at a given instance t_{exp}.

We acquire and annotate a large image dataset to train and test the CNN classification model [17]. This dataset describes four classes corresponding to different snow cover levels that are in accordance with snow removal operations. The classes are the following: "clear surface", "light-covered surface", "Medium-to-heavy-covered surface", and "plowed surface". Fig. 2 showcases a sample of each class.

We exploit pre-trained CNNs that we adapt to the snow level classification task. We exclude the CNN top layers and replace them with two fully connected layers of sizes 256 and 128, respectively. The output of the network is the output of the softmax function applied to the last layer. The output size corresponds to the four target classes.

As snowfall events are not as frequent as normal weather, the collected dataset is imbalanced, i.e., some classes have a higher instance number than others. We use a class-weighted loss function to avoid the model being biased toward majority classes. This loss function considers the skewed distribution of the classes while training the CNN model. For each class i, the class weight w_i is given by:



(a) Clear surface (b) Light-covered surface (c) Medium-to-heavy-covered surface (d) Plowed surface

FIGURE 2. Sample images of the different snow cover classes.

$$\omega_i = \begin{cases} \frac{n}{N \times n_i} & \text{if } i \in [\text{light, plowed}], \\ 1 & \text{else}, \end{cases} \tag{1}$$

where N is the number of classes, n is the total number of dataset instances, and n_i is the number of samples in class i. The loss function is initially calculated for a single sample as:

$$\text{Loss} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i), \tag{2}$$

where y_i is the ground truth label, $\hat{y}_i$ is the predicted value, and log() is the natural logarithm. The loss function is now expressed as follows:

$$\text{Loss} = -\sum_{i=1}^{N} \omega_i y_i \log(\hat{y}_i). \tag{3}$$

The output of this step provides an attribute of each road segment covered by a CCTV camera. The label indicates whether at a given instance t_{exp}, a road segment is cleared, plowed, lightly covered, or Medium-to-heavy-covered. Hence, an indication on the status of the road is provided instantaneously. Subsequently, priority will be given to medium-to-heavy- and light-covered surfaces.

C. VEHICLE COUNTING AND TRAFFIC ESTIMATION

Traffic volume is a necessary parameter for determining the streets' priorities for snow removal. In this phase, we estimate the traffic volume in intersections equipped with CCTV cameras. In our case, we do not have access to the video stream. Instead, the cameras post only images controlled by a refresh rate. To estimate traffic importance,

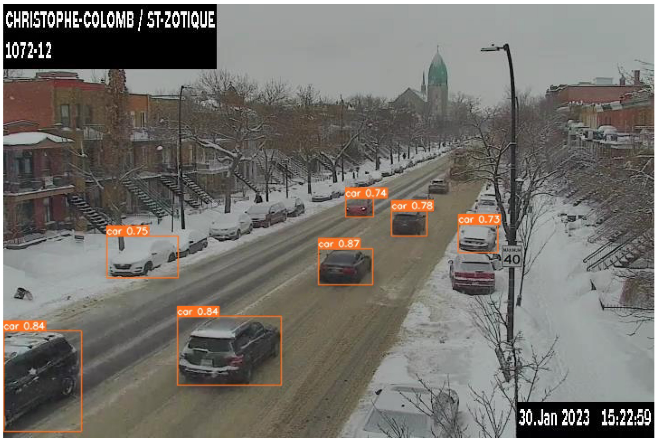


FIGURE 3. Example of the YOLOv5 object detector output.

we calculate the number of vehicles during the past period T_{count} for each node.

The counting step consists in using the YOLOv5 [18] object detector to detect different objects in an image, filter out vehicles, and count them. YOLOv5 is one of the most efficient and widely used object detectors, as it can accurately detect objects at different scales within a fast inference time [19]. For the vehicle counting task, we use the YOLOv5l (large) variant having 46.5 million parameters. The detector is trained on the COCO dataset and achieves 48.8% mean Average Precision for detection. The classes we consider are "car", "bus", and "truck". We filter the detected vehicles and keep ones having prediction confidence greater than 70%.[1]

We also distinguish parked and moving vehicles. We leverage one property of the CCTV cameras, which is having a fixed view during large time periods. For each image, we compare the detected vehicles with the previous image. Parked vehicles would have almost the same detected bounding boxes in both images. We use the Intersection over Union (IoU) metric to detect the parked vehicle boxes, with a threshold of 90%.

Fig. 3 shows an example of the vehicle detection algorithm output. Using the YOLOv5 object detector, we can then count the number of moving vehicles during a certain period of time before the beginning of the snow removal operation. Hence, an indication on the congestion level of the roads is given and priority levels will be assigned accordingly, as it will be detailed in the next section.

D. GRAPH WEIGHT COMPUTATION

In this section, we aim to compute the priority weights of all edges of the graph representing the road network of the area of interest. We proceed with two steps: first, we compute the weights for the edges/roads covered by the CCTV cameras. Then, we employ a weighted extrapolation strategy on the graph to estimate weights for unmonitored roads.

1) WEIGHT COMPUTATION FOR MONITORED ROADS

Having estimated snow levels and traffic volumes on roads monitored by CCTV cameras, we can assign weights to their corresponding edges. Each weight represents a priority coefficient to perform a snow removal operation on the road. In addition to snow level and vehicle count, each road has a type attribute. The road type can be "primary", "secondary", "tertiary", or "residential". We characterize each type by a coefficient r_{ij} ranging from 0 to 1. We define r_1, r_2, r_3, and r_4 as the coefficient values for each road type, respectively, which are the parameters to define when operating depending on the need for snow removal. Following the same approach, we assign coefficients s_{ij} for snow level categories "clear surface", "light-covered surface", "Medium-to-heavy-covered surface", and "plowed surface" with the values with values s_1, s_2, s_3, and s_4, respectively. Finally, we assign for each road a binary coefficient a_{ij} that indicates if any important amenity (such as hospitals, schools, fire and police departments) are located on that road. The weights measuring the p[priority of each road e_{ij} connecting the nodes i and j and measuring are computed as follows:

$$W_{ij} = s_{ij} \times \frac{\left(r_{ij} + v_{ij}^{norm} + a_{ij}\right)}{3}, \tag{4}$$

where v_{ij}^{norm} is the normalized vehicle count for edge e_{ij}, and $a_{i,j}$ represents the amenity coefficient.

Note that, s_{ij} and v_{ij}^{norm} have the same values for all edges monitored by the same camera, while r_{ij} may vary according to the 2D map input.

2) WEIGHT EXTRAPOLATION FOR UNMONITORED ROADS

Previously, we proposed a computer vision-based method to estimate snow levels and traffic volume on roads monitored by CCTV cameras and compute their priority coefficients. However, edges corresponding to unmonitored roads remain unweighted.

We propose a method that leverages the network topology and the GCN [22] to extrapolate the weight values. GCN is a type of neural network that handles graph-structured data. GCNs can perform different node-level or graph-level tasks, such as graph classification [23], node classification [24], or link prediction [25]. The GCN takes as inputs a feature vector and a graph representation to produce a node-level output. The GCN uses graph convolution layers to locally aggregate the features from neighboring nodes. It computes a weighted sum of the feature vectors of neighboring nodes and incorporates it into the current node's representation. This aggregation process is performed for every node in the graph. Later, a node update function is applied to

1. In this study, we use YOLOv5 pre-trained weights as they are. Hence, the efficiency of the object detector in adverse weather conditions might be degraded. Multiple computer vision techniques can be employed to cope with this issue, including deraining and dehazing using generative models [20], [21]. This requires elaborate efforts and has been left to a future extension of this work.

incorporate the aggregated information into the current node's representation.

Multiple works applied the GCN to road network graphs to extrapolate node attributes or edge weights. In [26], Friji et al. used the GCN as an encoder to embed a road network graph and then extrapolate edges weights by decoding the embedding. In [27], the authors proposed a framework for stochastic weight completion using GCN in the context of vehicle routing. They exploit the network topology and neighbor correlations to generate a fully-weighted graph. In [28], the authors leveraged a GCN model to extract spatial correlation features in an urban road network in order to build a framework for traffic flow prediction. This model takes as input two different graphs representing regional traffic information. In our work, the CGN estimates snow level and vehicle count at each unmonitored camera. We then apply (4) to compute the weights.

We exploit the GCN in a semi-supervised setting for node classification, as we possess a training set that consists of labeled samples (monitored vertices) and unlabeled samples (unmonitored vertices). The number of unlabeled samples is much greater than labeled samples. In our case, we train the GCN model in a featureless mode, as the nodes have no input features, which is our case. An identity matrix is used instead of the input features as suggested in [22]. During training, the GCN leverages the graph topology and the labeled nodes to share features between neighboring nodes and extract node embeddings that are used for classification. The GCN architecture that we use has three hidden graph convolutional layers with ReLU activation function as a node update function. We add a softmax layer on top of the GCN output is added to obtain class probabilities.

We train the GCN for the snow level and traffic volume estimation separately. For snow level, the values of camera nodes are one-hot encoded and fed to GCN as the labeled instances. On the other hand, we categorize the vehicle count attribute into four even bins ranging between the minimum and maximum vehicle count and then one-hot encode it. The vehicle count extrapolation becomes a classification problem rather than a regression one.

E. SNOW REMOVAL VEHICLE ROUTING

In this section, we consider the problem of routing one snow removal truck. Let O be the starting point for the truck, for example, the dispatch center. It should follow a path from O to a destination point D that plows the maximum possible high-priority roads. The routing problem consists of minimizing the objective function F defined as:

$$F = \prod_{ij \in E} \frac{1}{W_{ij}}, \tag{5}$$

where W_{ij} is the weight (priority) of edge e_{ij}.

To find the path between O and D, we modify Dijkstra's algorithm to minimize F. The original algorithm minimizes

Algorithm 1 Dijkstra's Modified Algorithm for Road Priority Maximization

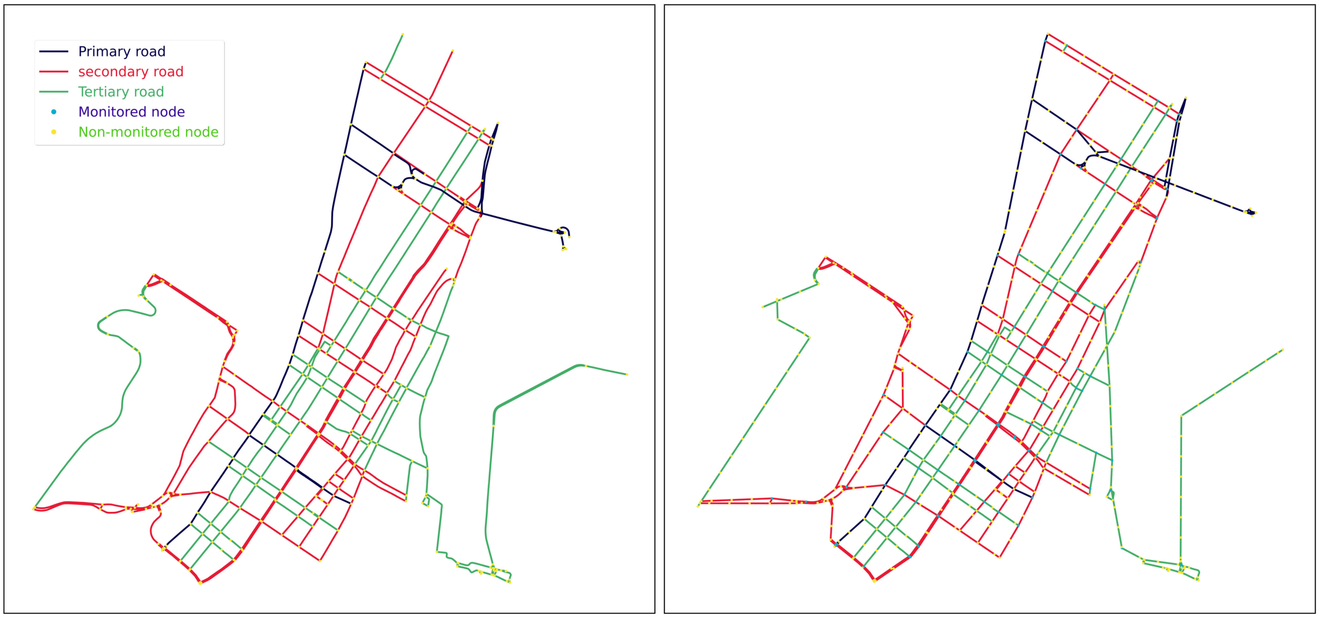
```
function WEIGHTED_DIJKSTRA(G,O,D)
    dist[node] ← ∞ for node in G.nodes
    dist[O] ← 1
    prev ← {}
    heap ← [(1, O)]
    while heap ≠[]  do
        (d, node) ← heap.pop()
        if node = D then
            path ← []
            while node in prev do
                path.append(node)
                node ← prev[node]
            end while
        end if
        path.append(O)
        path.reverse()
        break
        if d > dist[node] then
            continue
        end if
        for neighbor, weight in G[node] do
            alt ← dist[node] × (1/weight)
            if alt < dist[neighbor] then
                dist[neighbor] ← alt
                prev[neighbor] ← node
                heap.push(alt, neighbor)
            end if
        end for
        return dist[target], path
    end while
end function
```

the sum of edge weights to build the shortest route. At the start of the algorithm, the distance to the source is initialized to 0. At each node, the algorithm selects the next node that would make the minimum cumulative sum. In our case, we aim to build a path that prioritizes visiting high-priority edges. As $0 < W_{ij} \leq 1$, the algorithm should select the neighboring node that would make the maximum cumulative product, i.e., closer to 1. Since Dijkstra's algorithm is used for minimizing the edges' weights, we use the inverse of the priorities instead. Hence, we minimize the product of the priority inverses.

We modify Dijsktra's objective function to $\min(F)$. To avoid multiplication by 0, the distance to the source is initialized as 1. The pseudo-code for the algorithm is shown in Algorithm 1.

IV. SIMULATIONS AND DISCUSSIONS

In this section, we present the different data sources we use to develop the proposed framework. We showcase the outputs of the vehicle routing framework components through a case study.



(a) Initial road network graph.

(b) Road network graph after pre-processing.

FIGURE 4. "Ville-Marie" borough road network graph representation after, camera identification, dead-end roads removal, dividing long roads into segments that are covered by cameras, and finally consolidating the intersections.

A. DATASET PRESENTATION

1) ROAD NETWORK MAP

In this section, we focus on two boroughs of Montreal, QC, Canada: "Ville-Marie" (VM) and "Le Plateau-Mont-Royal" (PMR). To extract the road network for the study areas, we use the OSMnx [29] and NetworkX[2] python packages to fetch and manipulate the map graph through the OpenStreetMap[3] project. For the simulations, we only extract "primary", "secondary", and "tertiary" roads for simplicity.

After its extraction, the graph undergoes several pre-processing steps. First, we identify nodes with CCTV cameras installed. Next, we detect and remove nodes and edges corresponding to dead-end roads. After that, we slice roads longer than the camera range d into two portions, as explained previously. In the final pre-processing step, we consolidate the graph's vertices. It consists of merging clusters of nearby vertices into one vertex and then rebuilding the graph to keep the same topology. The consolidation helps simplify the graph and remove redundant vertices. Fig. 4 shows the results of the pre-processing steps, where dead-ends are discarded, roads are split into shorter segments, and neighboring nodes are merged into one.

2) ROAD MONITORING CAMERAS IMAGE DATASET

To acquire computer vision information (snow level and traffic volume), we leverage the road-monitoring CCTV cameras installed in road intersections in Montreal.[4] There are 528 cameras with numerous urban scenes such as highways, residential, commercial, and industrial areas. The cameras' list is provided as a GeoJSON file specifying the coordinates of each camera, a unique identifier, a borough identifier, and a link to its image feed. The cameras provide an image snapshot every 5 to 10 minutes.

We collect an image dataset during the 2022 winter season.[5] The dataset presents a variety of settings, including time, traffic volume, weather conditions, and visibility. We annotate the dataset automatically using an AI-based framework that we proposed in [17]. The dataset includes 41346 annotated images that describe the previously defined snow level classes, distributed as the following:

- Clear surface: 17422 images.
- Light-covered surface: 3726 images.
- Medium-to-heavy-covered surface: 14725 images.
- Plowed surface: 3512 images.

As the numbers propose, the dataset is imbalanced with two majority classes and one minority class. As our annotation framework depends on clustering, the dataset is susceptible to label noise, with both minority classes are more likely to have misannotated images due less representation. To check the quality of annotations for these two classes, we performed a manual test by randomly selecting 100 images from both minority classes and found out that 89% of the images have correct annotations. We use these images along with the camera information for snow level classification, vehicle counting, and VRP simulations.

2. https://networkx.org/
3. https://www.openstreetmap.org/
4. Montreal traffic cameras website: https://ville.montreal.qc.ca/circulation.
5. Image dataset available at: https://github.com/mohamedkaraa/Snow-Covered-Roads-Dataset.

TABLE 2. Benchmark of the test classification models.

Dataset	Model	Accuracy (%)	Precision (%)	Recall (%)
All	ResNet50	91	92	90
Day	ResNet50	97.2	97.3	97.2
	EfficientNet	**97.3**	**97.3**	**97.3**
Night	ResNet50	97.2	97.3	97.2
	EfficientNet	**97.4**	**97.5**	**97.4**

B. GRAPH WEIGHTS COMPUTATION

1) SNOW LEVEL CLASSIFICATION RESULTS

To train and test the CNN models, we split the dataset into day and night times and then train and test sets with proportions of 80% and 20%, respectively. We randomly shuffle and split images of each class to keep the same proportions. We suppose that the day-night split can improve the models' performance as the class distribution varies in these two sets.

We train and test two models having a ResNet50 [30] (~74 million parameters) and EfficientNetB3 [31] (~48 million parameters) as backbones respectively. The models are trained using the class-weighted cross-entropy loss function defined in (3) for 60 epochs each with an initial learning rate of 10^{-3}.

Table 2 shows results for the different models trained on different dataset settings. We notice that by training with more specific datasets (day and night), the models are able to better distinguish the different classes. The ResNet and EfficientNet models have similar results for the proposed settings. We opt to use the EfficientNet network as it has less number of parameters and allows faster training and testing times.

To evaluate the model's performance in extreme conditions, we use the associated metadata to extract images during heavy snowfall. We select 50 images representing the case where visibility is the most reduced compared to other instances and we manually check the obtained classes. The EfficientNet model scores 79% accuracy on these images.

2) WEIGHT COMPUTATION

For the experimental simulations, we operate on the two previously mentioned boroughs. We select two different periods during two ongoing snowstorms which are D1 (18^{th} February at 08:00 am) and D2 (25^{th} February at 12:30 pm). The camera maximum coverage d is set to 200 meters, and the vehicle count period T_{count} to 60 minutes. We perform the snow classification and the vehicle counting on monitored edges. Then, we apply the GCN method to estimate these parameters on unmonitored streets. Table 1 recapitulates the experiment settings and the output of the weight computation phase for each graph. After that, we compute the priority weights for all edges.

TABLE 3. Performance of the priority weight-based routing method compared to the scenario of no road priorities.

	Operation time (minutes)		High-priority roads (W=1)	
	Weighted	Unweighted	Weighted	Unweighted
Case #1	25	20	25	10
Case #2	34	27	20	11
Case #3	33	28	35	10
Case #4	25	19	23	15
Case #5	18	12	17	7
Case #6	30	18	24	7

For this experimental simulation, we set the road type coefficients as: $r_1 = 1$, $r_2 = 0.75$, $r_3 = 0.5$, (we don't consider residential roads for simplicity) and the snow level coefficients as $s_1 = s_4 = 0$, $s_2 = 0.5$, and $s_3 = 1$. The obtained weights are quantified into four categories to avoid having priority values of 0. Fig. 5 illustrates an example of the distribution of high and low-priority roads in the weighted graph of "Ville-Marie" borough.

C. SNOW REMOVAL VEHICLE ROUTING

In this section, we evaluate the snow removal performance based on the routing output. We assume that the truck performs only one operation, which is snow plowing that corresponds to the snow level class "Medium-to-heavy-covered surface".

To evaluate our proposed method, we compare snow removal progress to that provided by the scenario of no prioritizing. This scenario consists in finding the shortest path between origin O and destination D points. Fig. 6 shows an example of the different vehicle routes found for the weighted graph (blue) for maximum high-priority road plowing and the unweighted graph (red) for shortest path road plowing. Compared to the priority distribution map in Fig. 5, we can observe that our proposed method for vehicle routing follows the path in which more high-priority roads are present.

To compare the two methods, we are interested in calculating two metrics. The first one being the number of high-priority roads ($W = 1$) that are plowed during an operation between an $O - D$ pair. The second metric is the operation time which is the time taken to achieve the operation in the area of interest. Table 3 shows a comparison between the two methods for six cases where each case represents a different $O - D$ pair. We assume that the snow removal truck operates at speed of 10 km/h.

We evaluate the routing through six different cases. Cases 1, 2, and 3 are in "Ville-Marie" borough, on the dates D1, D2, and D1, respectively, while cases 4, 5, and 6 are in "Le Plateau-Mont-Royal" borough on the dates D2, D2, and D1, respectively. We observe that our proposed method focuses

TABLE 1. Study area attributes extracted from computer vision and graph convolutional network.

		Vertices		**Edge**		**Date**	**Snow Level Distribution**				**Vehicle Count**	
Borough	**Category**	*CCTV*	*No CCTV*	*Monitored*	*Unmonitored*		*Clear*	*Light*	*Medium-to-heavy*	Plowed	*Max*	*Min*
VM	**Number**	99	447	325	417	D1	76	52	486	128	25	0
						D2	0	0	722	20	31	0
PMR		47	180	143	143	D1	6	15	250	15	54	0
						D2	0	11	275	0	30	0

TABLE 4. Inference runtime for the framework components.

Component	Snow level classification (per camera)	Vehicle count estimation (per camera)	Snow level extrapolation (GCN)	Vehicle count extrapolation (GCN)	Routing	Total
Execution time (ms)	226	3700	61	65	$1 \sim 2$	4054

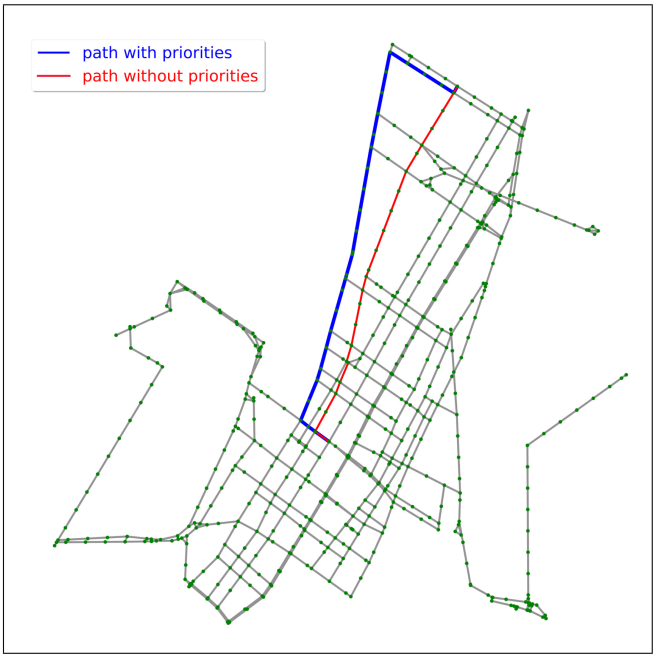


FIGURE 5. Example of quantized road priority weights distribution in "Ville-Marie" borough on 18th February 2022 at 08:00 am.

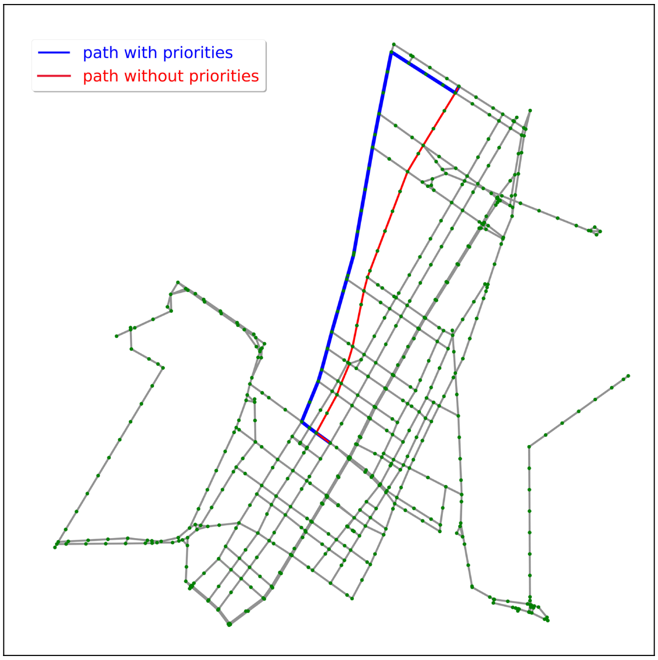


FIGURE 6. Example of snow removal truck route recommendation for the priority weighted graph (blue) and unweighted graph (red).

more on plowing high-priority roads. However, it generates longer paths hence more operation time.

We report the inference runtime for the different framework components in Table 4. We use a NVIDIA GTX 1650 GPU with 4Gb of video memory for inference. For the vision-based tasks, we take the per camera value, whereas for the extrapolation task, we take the GCN prediction time. The runtime values are averaged on multiple runs of the framework. We observe that the vehicle counting task has the most impact on the execution time, as it involves processing current and previous image frames for each camera.

V. CONCLUSION & FUTURE WORK

Snow removal is a critical task for maintaining mobility during the winter season. An efficient strategy must be put forward to prioritize access to vital services and busy locations. In this paper, we have proposed a framework for snow removal vehicle routing based on road priorities. The framework models the urban road network as a graph with road priorities as weights from a 2D map and CCTV images as input. We have developed a computer vision approach to extract the priorities. In fact, we have exploited CCTV camera images to parallelly estimate snow level covering the roads and predict traffic volume. However, in an urban road network, most roads are not equipped with CCTV cameras. Hence, we have used a Graph Convolutional Network to estimate the snow level and traffic volume in unmonitored road segments. Then, we have established a method to calculate the priorities based on the estimated parameters and the road types.

Once we have calculated the weights, we determine the path for a single snow removal truck from a source point to a destination. This path aims to plow the maximum number of high-priority roads between the two points. In order to find this path, we have modified the Dijkstra algorithm to minimize the product of the inverse of the weight. We validated our approach on a real-world use case in which we model a study area located in Montreal, Quebec, Canada.

In future work, we aim to develop an advanced vehicle routing algorithm for snow removal trucks based on the priorities generated in this work. Such an algorithm will be able to provide a trade-off between priorities and operation time. In addition, we will take into account the real topology of the graph in which road segments are directed and may need more than one passage to be plowed. In addition, many problems related to snow removal operations can be investigated such as the location of truck dispatch centers and the optimization of the number of vehicles to perform different snow removal operations. Finally, we intend to address the problem of object detection in snowy weather to improve the performance of the vehicle counting component as an adaptive preprocessing step.

REFERENCES

[1] B. Nguyen, M. J. Shinnie, C. K. Leung, M. R. Kuzie, N. N. Kokilev, and S. Kaur, "A data science solution for mining weather data and transportation data for smart cities," in *Proc. IEEE 46th Annu. Comput., Softw., Appl. Conf. (COMPSAC)*, 2022, pp. 1672–1677.

[2] H. Kinoshita, M. Yagi, S. Takahashi, and T. Hagiwara, "A method for classifying road narrowing conditions based on features of surrounding vehicles and piled snow in in-vehicle camera videos," in *Proc. IEEE Int. Conf. Consum. Electron. -Taiwan*, 2022, pp. 299–300.

[3] Y. Shibata, Y. Arai, Y. Saito, and J. Hakura, "Development and evaluation of road state information platform based on various environmental sensors in snow countries," in *Proc. Adv. Internet, Data Web Technol.*, 2020, pp. 268–276.

[4] C. Zhang, E. Nateghinia, L. F. Miranda-Moreno, and L. Sun, "Winter road surface condition classification using convolutional neural network (CNN): Visible light and thermal image fusion," *Can. J. Civ. Eng.*, vol. 49, no. 4, pp. 569–578, Apr. 2022.

[5] M. N. Khan and M. M. Ahmed, "Weather and surface condition detection based on road-side webcams: Application of pre-trained convolutional neural network," *Int. J. Transp. Sci. Technol.*, vol. 11, no. 3, pp. 468–483, Sep. 2022.

[6] D. Grabowski and A. Czyżewski, "System for monitoring road slippery based on CCTV cameras and convolutional neural networks," *J. Intell. Inf. Syst.*, vol. 55, no. 3, pp. 521–534, Dec. 2020.

[7] R. Akiyama, H. Yamamoto, and Y. Kitatsuji, "Wide-area road surface condition observation system utilizing traveling sensing by LiDAR," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2022, pp. 1–6.

[8] G. Pan, M. Muresan, R. Yu, and L. Fu, "Real-time winter road surface condition monitoring using an improved residual CNN," *Can. J. Civil Eng.*, vol. 48, no. 9, pp. 1215–1222, Sep. 2021.

[9] K. Holmberg, "The (over)zealous snow remover problem," *Transp. Sci.*, vol. 53, no. 3, pp. 623–916, May/Jun. 2019.

[10] V. Soloveva, E. Kuchinskaya, N. Antonov, and T. Volotskiy, "Analyzing efficiency of different snow removal strategies in terms of the system wide time savings using dynamic link capacity in large-scale Microsimulation," *Procedia Comput. Sci.*, vol. 178, pp. 116–124, Dec. 2020.

[11] W. Zhou and B. Wu, "Model for snow-fighting vehicle route planning considering deadheading restriction," in *Proc. 2nd Int. Conf. Comput. Sci. Appl. Eng.*, 2018, pp. 1–5.

[12] J. Wang and H. Liu, "Snow removal resource location and allocation optimization for urban road network recovery: A resilience perspective," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 1, pp. 395–408, Jan. 2019.

[13] A. Rasul, J. Seo, S. Xu, T. J. Kwon, J. MacLean, and C. Brown, "Snowplow route optimization using chinese postman problem and tabu search algorithm," in *Proc. 38th Int. Assoc. Autom. Robot. Const.*, Nov. 2021, pp. 403–409.

[14] L. Jin, S. Lin, B. Xie, L. Liu, "A vulnerability-based vehicle routing approach for solving capacitated arc routing problem in urban snow plowing operations," *Math. Biosciences Eng.*, vol. 18, no. 1, pp. 166–181, 2021.

[15] S. Xu and T. J. Kwon, "Optimizing snowplow routes using all-new perspectives: Road users and winter road maintenance operators," *Can. J. Civil Eng.*, vol. 48, no. 8, pp. 959–968, Aug. 2021.

[16] S. H. Park, Y.-H. Jang, Z. W. Geem, and S.-H. Lee, "CityGML-based road information model for route optimization of snow-removal vehicle," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 12, p. 588, Dec. 2019.

[17] M. Karaa, H. Ghazzai, L. Sboui, H. Besbes, and Y. Massoud, "Unsupervised image dataset annotation framework for snow covered road networks," in *Proc. IEEE Asia Pacific Conf. Circuits Syst. (APCCAS)*, Nov. 2022, pp. 110–114.

[18] G. Jocher et al. "Ultralytics/YOLOv5: V7.0—YOLOv5 SOTA realtime instance segmentation" Nov. 2022. [Online]. Available: https://github.com/ultralytics/yolov5/discussions/10258

[19] M. Masmoudi, H. Ghazzai, M. Frikha, and Y. Massoud, "Object detection learning techniques for autonomous vehicle applications," in *Proc. IEEE Int. Conf. Veh. Electron. Saf. (ICVES)*, 2019, pp. 1–5.

[20] Q. Qin, K. Chang, M. Huang, and G. Li, "DENet: Detection-driven enhancement network for object detection under adverse weather conditions," in *Proc. Comput. Vis. – ACCV 2022*, 2023, pp. 491–507.

[21] M. Hassaballah, M. A. Kenk, K. Muhammad, and S. Minaee, "Vehicle detection and tracking in adverse weather using a deep learning framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4230–4242, Jul. 2021.

[22] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Representations, ICLR*, 2017, pp. 1–14.

[23] F. Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein, "Fake news detection on social media using geometric deep learning," in *Proc. CoRR*, 2019, pp. 1–15. [Online]. Available: http://arxiv.org/abs/1902.06673

[24] Y. Rong, W. Huang, T. Xu, and J. Huang, "DropEdge: Towards deep graph convolutional networks on node classification," in *Proc. 8th Int. Conf. Learn. Representations, ICLR* , 2020, pp. 1–17.

[25] H. Chen, H. Yin, X. Sun, T. Chen, B. Gabrys, and K. Musial, "Multi-level graph convolutional networks for cross-platform anchor link prediction," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, Aug. 2020, pp. 1503–1511.

[26] H. Friji, A. Khanfor, H. Ghazzai, and Y. Massoud, "An end-to-end smart IoT-driven navigation for social distancing enforcement," *IEEE Access*, vol. 10, pp. 76824–76841, 2022.

[27] J. Hu, C. Guo, B. Yang, and C. S. Jensen, "Stochastic weight completion for road networks using graph convolutional networks," in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, Apr. 2019, pp. 1274–1285.

[28] Y. Zheng, S. Wang, C. Dong, W. Li, W. Zheng, and J. Yu, "Urban road traffic flow prediction: A graph convolutional network embedded with wavelet decomposition and attention mechanism," *Physica A Stat. Mechan. Appl.*, vol. 608, Dec. 2022, Art. no. 128274.

[29] G. Boeing, "OSMnx New methods for acquiring, constructing, analyzing, and visualizing complex street networks," *Comput., Environ. Urban Syst.*, vol. 65, pp. 126–139, Sep. 2017.

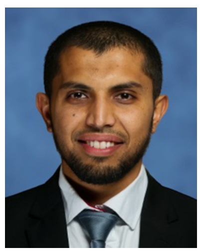
[30] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[31] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

MOHAMED KARAA (Student Member, IEEE) received the Diplome d’Ingénieur degree in telecommunication engineering from the Higher School of Communication of Tunis, Tunis, Tunisia, in 2022. He is currently pursuing the master’s degree in information technology engineering with the École de Technologie Supérieure, Montreal, Canada. He was a visiting student with the King Abdullah University of Science and Technology in 2022. His current research interests include deep learning applications for intelligent transportation systems and wireless communication.

HAKIM GHAZZAI (Senior Member, IEEE) received the Diplome d’Ingenieur degree (Hons.) in telecommunication engineering and the master’s degree in high-rate transmission systems from the Ecole Superieure des Communications de Tunis, Tunis, Tunisia, in 2010 and 2011, respectively, and the Ph.D. degree in electrical engineering from the King Abdullah University of Science and Technology (KAUST), Saudi Arabia, in 2015. He was a Researcher Scholar with Qatar Mobility Innovations Center, Qatar; Karlstad University, Sweden; and the Stevens Institute of Technology, Hoboken, NJ, USA. He is currently a Research Scientist with KAUST. He is the author and a coauthor of more than 170 publications. His general research interests include artificial intelligence-enabled applications, the Internet of Things, intelligent transportation systems, and mobile and wireless networks. He is a recipient of appreciation for being an Exemplary Reviewer for IEEE WIRELESS COMMUNICATIONS LETTERS in 2016 and IEEE COMMUNICATIONS LETTERS in 2017. Since 2019, he has been on the editorial board of the IEEE COMMUNICATIONS LETTERS and the IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY. In 2020, he joined the Board of IoT and Sensor Networks (specialty section of *Frontiers in Communications and Networks*) as an Associate Editor.

YEHIA MASSOUD (Fellow, IEEE) received the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. He is currently the Director of the Innovative Technologies Laboratories, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia. He has held several positions at leading institutions of higher education and the industry including Rice University, Houston, TX, USA; the Stevens Institute of Technology, Hoboken, NJ, USA; WPI; UAB; SLAC National Accelerator Laboratory; and Synopsys Inc. From January 2018 to July 2021, he was the Dean of the School of Systems and Enterprises, Stevens Institute of Technology. Prior to Stevens, he was the Head of the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA, USA, from 2012 to 2017. In 2003, he joined Rice University as an Assistant Professor, where he became one of the fastest Rice Faculty to be granted tenure in electrical and computer engineering and computer science in 2007. His research interests include the design of state-of-the-art innovative technological solutions that span over a broad range of technical areas including smart cities, autonomy, smart health, smart mobility, embedded systems, nanophotonics, and spintronics. He was selected as one of ten MIT Alumni Featured by MIT’s Electrical Engineering and Computer Science Department in 2012. He was the recipient of the Rising Star of Texas Medal, the National Science Foundation CAREER Award, the DAC Fellowship, the Synopsys Special Recognition Engineering Award, and several best paper awards. He was also named a Distinguished Lecturer by the IEEE Circuits and Systems Society. He was an Editor of the Mixed-Signal Letters—the Americas, as an Associate Editor of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, and a Guest Editor of a special issue of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS. He was also the 2016 IEEE MWSCAS Technical Program Co-Chair, and the 2009 General Program Co-Chair and the 2007 Technical Program Co-Chair of the ACM Great Lakes Symposium on VLSI.



LOKMAN SBOUI (Senior Member, IEEE) received the Diplome d’Ingénieur degree (Hons.) from the Ecole Polytechnique de Tunisie, La Marsa, Tunisia, in 2011, and the M.Sc. and Ph.D. degrees from the King Abdullah University of Science and Technology in 2013 and 2017, respectively. He is currently an Assistant Professor with the Department of System Engineering, École de Technologie Supérieure, Montreal, Canada. He was a Postdoctoral Researcher with the Richard J. Marceau Chair on Wireless IP Technology for Developing Countries at ÉTS from 2018 to 2020. He was a certification analyst for mobile devices at Vidéotron, a Canadian telecom operator located in Québec, from 2020 to 2021. He has been an Editorial Board Member of IEEE WIRELESS COMMUNICATIONS LETTERS since 2021. His current research interests include: energy efficient wireless communications, LEO satellites, IoT applications, automation, urban air mobility performance of cognitive radio systems, and VoIP protocols.