Machine Learning to Improve Multi-hop Searching and Extended Wireless Reachability in V2X

Manuel Eugenio Morocho-Cayamcela, *Member, IEEE,* Haeyoung Lee, *Member, IEEE,* and Wansu Lim, *Member, IEEE*

***Abstract*—Multi-hop relay selection is a critical issue in vehicle-to-everything networks. In previous works, the optimal hopping strategy is assumed to be based on the shortest distance. This study proposes a hopping strategy based on the lowest propagation loss, considering the effect of the environment. We use a two-step machine learning routine: improved deep encoder-decoder architecture to generate environmental maps and *Q*-learning to search for the multi-hopping path with the lowest propagation loss. Simulation results show that our proposed method can improve environmental recognition and extend the reachability of multi-hop communications by up to 66.7%, compared with a shortest-distance selection.**

***Index Terms*—Machine learning, multi-hop wireless communication, *Q*-learning, vehicle-to-everything.**

I. Introduction

VEHICLE to everything (V2X) networks constitute automobiles and involve entities that act as wireless nodes for exchanging standardized information. This data is transmitted using one-way or two-way dedicated short-range communication (DSRC), which are specifically designed wireless communication channels that correspond to a set of protocols and standards [1]. Mechanisms such as intelligent multi-hop relay selection and route searching are interesting and challenging methods of extending the communication over a large area. For further enhancement of DSRC, researchers have looked into several areas including the improvement of reachability (currently limited to approximately 300 m) [2]. Previous works based the design of multi-hop search algorithms in the minimization of the Euclidean distance to the destination [3]. Nevertheless, a real non-line-of-sight (NLOS) V2X scenario includes obstructions where the shortest-distance hopping might not be always ideal. A better approach would be to minimize the propagation loss along the entire multi-hop path by recognizing the location of the obstructions that attenuate the signal, and avoiding transmission through them.

Recent wireless communications and artificial intelligence efforts have investigated the recognition of the patterns in these obstructions [4]. Before adopting deep learning (DL), researchers used decision-tree based approaches such as *texton forests* with low accuracy results [5]. These days, convolutional neural networks (CNNs) have increased the segmentation performance significantly. *Patch-based* segmentation classifies the entire image using a collection of small patches [6]. A major drawback of using patches is that the classification network is composed of connected layers requiring fixed size images. To overcome this limitation, designs based on *fully-convolutional networks* (FCNs) use a pre-trained CNN to serve as a down-sampling encoder, and then up-sample the features using fractional-stride convolution [7]. However, this up-sampling approach introduces information loss and generates coarse segmentation maps [8]. Moreover, these DL models have not been exploited before to improve multi-hop selection by detecting obstructions in satellite imagery.

In this paper, we propose a two-step machine learning (ML) process to improve multi-hop relay selection and extend wireless reachability in a NLOS V2X network.

- First, we segment satellite images into different classes (i.e., buildings, open fields, and streets). To achieve this, we modify SegNet [8], a pretrained CNN encoder-decoder by updating its parameters according to the exponentially weighted average of the gradients. Our approach solves the coarse map problem, accelerates segmentation convergence, and increases accuracy as compared to FCNs.
- Second, we use the map generated by our segmentation network to assign a *reward* and *penalty* when transmission is done by a vehicle via a path with low propagation loss, and a high propagation loss obstruction, respectively. Our method enables the transmitting vehicle to learn an optimal hopping policy by maximizing the cumulative reward.

From the results in the first stage of our proposal, the obstacles recognition accuracy increases by an average of 3.04% when compared with FCNs. Results from the second stage show that the reachability of a V2X link can be extended by approximately 66.7% using our multi-hop selection policy compared with the shortest-distance strategy.

Manuscript received November 12, 2020; revised December 14 2019; accepted March 19, 2020. Date of publication March 31 2020; date of current version March 31 2020.

This work was supported by the National Research Foundation of Korea (NRF) under grant 2017R1C1B5016837; in part by the Technology Development Program (S2508336) funded by the Ministry of SMEs and Startups (MSS, Korea); and in part by the European Union's Horizon 2020 research and innovation programme 5G-HEART project under grant 857034.

M. E. Morocho-Cayamcela is with the Department of Electronic Engineering, Kumoh National Institute of Technology, Gumi, Gyeongsangbuk-do, 39177 South Korea. e-mail: (eugeniomorocho@kumoh.ac.kr).

H. Lee is with the 5G Innovation Centre (5GIC), Institute for Communication Systems (ICS), University of Surrey, Guildford, GU2 7XH U.K. (e-mail: haeyoung.lee@surrey.ac.uk)

W. Lim is with the Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi, Gyeongsangbuk-do, 39177 South Korea. e-mail: (wansu.lim@kumoh.ac.kr).

Digital Object Identifier

II. Addressing the reachability limitation with Machine Learning

The first stage of our proposal involves enhancing a convolutional encoder-decoder to detect obstacles that might block or attenuate the signal (section II-A). The second stage presents an off-policy model-free algorithm to find the multi-hop path with the lowest propagation loss, extending V2X reachability (section II-B). The model assumes: 1) massive machine-type communications (mMTC) scenarios with a large number of devices spread geographically, and 2) no prior knowledge of the propagation loss between devices.

A. Obstruction Detection in Satellite Imagery

To recognize obstructions, a DL network is trained to segment three types of classes (i.e., buildings, open fields, and streets) from aerial images. We use the INRIA aerial image dataset to train our algorithm [9]. INRIA is a collection of high-resolution imagery from different European and American landscapes (e.g., Austin, Bellingham, Bloomington, Chicago, Colorado, Innsbruck, Kitsap County, San Francisco, East Tyrol, and Vienna). The dataset contains 180 color images of 5000×5000 pixels, covering a surface of $1500\,\mathrm{m} \times 1500\,\mathrm{m}$ each. As opposed to [9], [10], we generate a ground truth map on top of INRIA by labelling every pixel in each of the 180 images with one of our environmental categories in $\boldsymbol{c} = \{\text{buildings, open fields, streets}\}$. We let each image $\boldsymbol{X}^{(i)} \in \mathbb{R}^{5000\times5000}$ be a matrix of 5000×5000 pixels. The set of m images on INRIA as $\boldsymbol{X} \triangleq \{\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(m)}\}$, where each pixel in $\boldsymbol{X}^{(i)}$ can be mapped to one of the categories in $\boldsymbol{c}$. $\boldsymbol{Y}^{(i)} \in \mathbb{R}^{5000\times5000}$ is used to present the corresponding ground-truth map. Similarly, we can express $\boldsymbol{Y} \triangleq \{\boldsymbol{Y}^{(1)}, \ldots, \boldsymbol{Y}^{(m)}\}$ for its corresponding set of labels. Labeling was conducted according to the following criteria:

- *Buildings*: include residential areas, and any field where the signal might be blocked by man-made structures.
- *Open fields*: include parks, and any open area where the signal might be blocked by sparse vegetation.
- *Streets*: include highways, or wherever a line of sight between the transmitter and receiver can be guaranteed.

The dataset was augmented by applying random left/right *reflection*, and *X/Y translation* to the images with a ±10 pixels range. After dataset augmentation, majority of the pixels correspond to the 'buildings' class. This imbalance biases the learning process in favor of the dominant class. To overcome this challenge, the classes are balanced by computing the *inverse frequencies*, where the class weights are set to the inverse of their frequencies. This method increases the weight given to the under-represented classes (i.e., 'open field', and 'street'). After the pixel-wise labeled dataset $[\boldsymbol{X} : \boldsymbol{Y}]$ is balanced, the dataset is divided into training set $[\boldsymbol{X}_{\text{tr}} : \boldsymbol{Y}_{\text{tr}}]$ and testing set $[\boldsymbol{X}_{\text{te}} : \boldsymbol{Y}_{\text{te}}]$. Fig. 1(a) illustrates how the convolutional encoder-decoder takes an input image $\boldsymbol{X}^{(i)}$ through *convolution*, *batch normalization*, *ReLu*, and *pooling* layers to build the down-sampling encoder. The encoder is followed by an up-sampling network with reverse architecture, where the *max pooling indices* are recalled from the corresponding encoder layer to decode lower resolution feature maps. Non-linear up-sampling improves boundary delineation and reduces the number of parameters for training [8]. The network is initialized using the weights of the pre-trained VGG-16 model [11], to exploit transfer learning by dropping the last fully-connected layers and replacing them with the categories in $\boldsymbol{c}$. If we let f be our network parametrized by $\boldsymbol{\theta}$, the segmentation output of the network is $\boldsymbol{M} = f(\boldsymbol{X}, \boldsymbol{\theta})$, where $\boldsymbol{M} \in \mathbb{R}^{5000\times5000}$ is a categorical matrix that maps every pixel from the input image to the corresponding category in $\boldsymbol{c}$. Our network is trained by updating $\boldsymbol{\theta}$ iteratively, moving the loss towards the minimum of the *cost function* $J(\boldsymbol{\theta})$. The cost function measures the pixel-wise performance of our model prediction against its corresponding ground truth. To quantify the difference between the two distributions, we let $J(\boldsymbol{\theta})$ be defined as the *pixel-wise cross-entropy*, because it penalizes the model when it estimates a low probability for a target category, producing larger gradients and converging faster [12]. For our multi-class segmentation problem with $K = |\boldsymbol{c}| = 3$ number of categories, and a training set with the values of $(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)})$ for $i \in \{1, \ldots, m_{tr}\}$, we find the set of parameters $\boldsymbol{\theta} = \{\theta^{(1)}, \ldots, \theta^{(n)}\}$ that minimizes $J(\boldsymbol{\theta})$ by computing a *per-example loss* $L(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\theta}) = -\log p(\boldsymbol{Y}|\boldsymbol{X}; \boldsymbol{\theta})$ for each category $k \in \boldsymbol{c}$, on every pixel-wise observation and sum the outcomes as follows:

$$\begin{aligned} J(\boldsymbol{\theta}) &= -\frac{1}{m_{\text{tr}}} \sum_{i=1}^{m_{\text{tr}}} \sum_{k=1}^{K} L(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)}, \boldsymbol{\theta}) \\ &= -\frac{1}{m_{\text{tr}}} \sum_{i=1}^{m_{\text{tr}}} \sum_{k=1}^{K} \boldsymbol{Y}_k^{(i)} log\left(\hat{p}_k^{(i)}\right), \end{aligned} \tag{1}$$

where $\boldsymbol{Y}_k^{(i)}$ represents the desired output for the i^{th} instance on class k, and $\hat{p}_k^{(i)}$ represents the estimated probability that the i^{th} instance belongs to k. Because the optimization of $\boldsymbol{\theta}$ involves calculating the derivative of $J(\boldsymbol{\theta})$ through partial differential equations, we can write the gradient vector of (1) with respect to $\theta^{(k)}$ as follows:

$$\begin{aligned} \nabla_{\theta^{(k)}} J(\boldsymbol{\theta}) &= \frac{1}{m_{\text{tr}}} \nabla_{\boldsymbol{\theta}} \sum_{i=1}^{m_{\text{tr}}} L(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)}, \boldsymbol{\theta}) \\ &= \frac{1}{m_{\text{tr}}} \sum_{i=1}^{m_{\text{tr}}} \left(\hat{p}_k^{(i)} - \boldsymbol{Y}_k^{(i)}\right) \boldsymbol{X}^{(i)}. \end{aligned} \tag{2}$$

Note that each class has its own dedicated parameter vector $\theta^{(k)}$, that constructs the parameter matrix $\boldsymbol{\theta}$. To find the local minimum of $J(\boldsymbol{\theta})$, we take steps proportional to the *negative* of the gradient of (2) for every i. The process is initiated by estimating an initial $\boldsymbol{\theta}$, and iteratively updating its value to reduce the value of the cost function.

Considering the training instances in the dataset may scale to a size where the optimization of $\boldsymbol{\theta}$ may be computationally prohibitive, we sample a minibatch $\mathbb{B} = \left\{\left[\boldsymbol{X}^{(1)} : \boldsymbol{Y}^{(1)}\right], \ldots, \left[\boldsymbol{X}^{(m'_{tr})} : \boldsymbol{Y}^{(m'_{tr})}\right]\right\}$, and let the optimization algorithm follow the gradient $\boldsymbol{g}$ downhill with:

$$\boldsymbol{g} \leftarrow \frac{1}{m'_{tr}} \nabla_{\boldsymbol{\theta}} \sum_{i=1}^{m'_{tr}} L(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)}, \boldsymbol{\theta}) \tag{3}$$

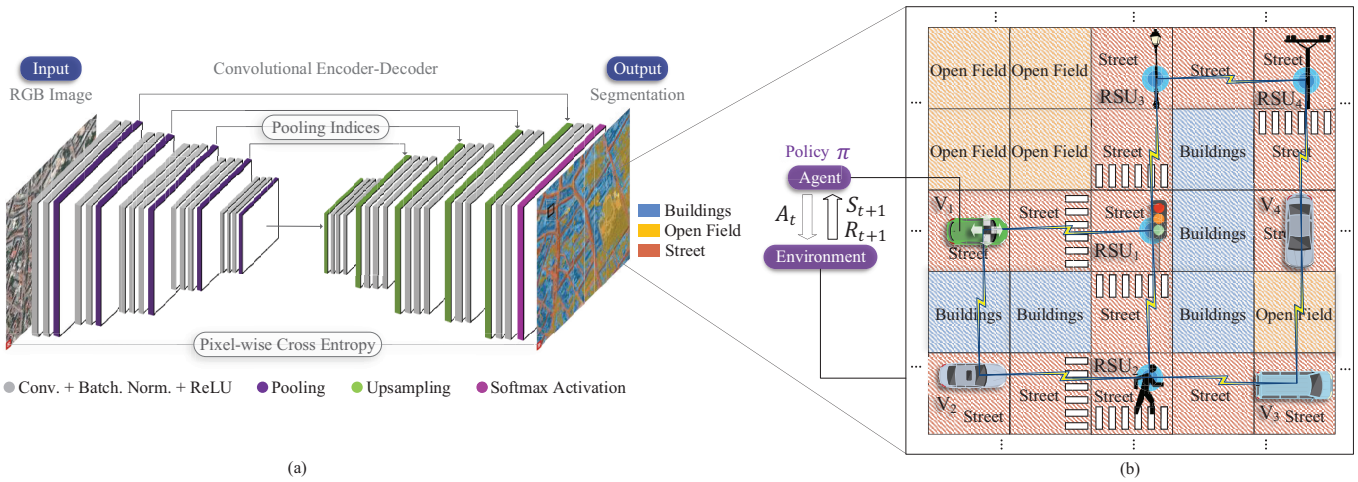


Fig. 1. Proposed system architecture. (a) Our convolutional encoder-decoder takes an aerial image as its input, and outputs a pixel-wise categorical matrix containing semantic information for each pixel. (b) The *agent* vehicle V_1 takes a transmitting *action* A_t on the *environment* and receives a *reward* R_t at state S_t. The action of the vehicle is taken based on a wireless transmission *policy* π. For the following iterations, V_1 receives a reward R_{t+1} and the state changes to S_{t+1}. V_1 learns an optimal transmission policy by maximizing the cumulative reward.

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta \boldsymbol{g}, \tag{4}$$

where β is a positive value that determines the size of each step in the minimization process, known as *learning rate*. A limitation of updating $\boldsymbol{\theta}$ with (4), is that the created oscillations prevent the use of a high value of β during training. To aggressively move towards the minimum, our algorithm updates $\boldsymbol{g}$ through a momentum parameter η, and an initial velocity $\boldsymbol{v}$ to compute the exponentially weighted average of the gradients and use the value to update $\boldsymbol{\theta}$, damping the oscillations in high-curvature directions by combining the gradients with opposite signs. The algorithm is initialized by the minibatch $\mathbb{B}$, $\boldsymbol{v} = 0$, and initial values for β and $\boldsymbol{\theta}$. For each iteration, $\boldsymbol{g}$ is computed with (3), and the values of $\boldsymbol{v}$ and $\boldsymbol{\theta}$ are updated as follows:

$$\boldsymbol{v} \leftarrow \eta \boldsymbol{v} - \beta \boldsymbol{g} \tag{5}$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{v}. \tag{6}$$

A pseudo-code of the segmentation task can be found in Algorithm 1.

Algorithm 1 Parameter optimization and image segmentation

Input: TX_{lat}, TX_{lon}, RX_{lat}, RX_{lon}, m, k, K, $\boldsymbol{x}$, $\boldsymbol{y}$, learning rate β, momentum η, initial parameter $\boldsymbol{\theta}$, initial velocity $\boldsymbol{v}$, R_b, R_o, R_s.
Output: Categorical matrix $\boldsymbol{M}$.

Initialization:
1: Initialize η to 0.9, β to 0.02, and $\boldsymbol{v}$ to 0. ▷ Selected after trials.
DATA ACQUISITION AND PRE-PROCESSING. (IN SECT. II-A.)
2: **Get** INRIA aerial images dataset ▷ From online server.
3: **for** each image **do**
4: Resize, translation, rotation, and class weighting.
5: **end for**
6: Sample a minibatch of m'_{tr} examples from the training set $\mathbb{B} = \left\{\left[\boldsymbol{X}^{(1)}: \boldsymbol{Y}^{(1)}\right], \ldots, \left[\boldsymbol{X}^{(m'_{tr})}: \boldsymbol{Y}^{(m'_{tr})}\right]\right\}$
CROSS-ENTROPY COST FUNCTION DEFINITION (SECT. II-A.)
7: $J(\boldsymbol{\theta}) = -\frac{1}{m_{\text{tr}}} \sum_{i=1}^{m_{\text{tr}}} \sum_{k=1}^{K} L(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)}, \boldsymbol{\theta})$
8: $\nabla_{\theta^{(k)}} J(\boldsymbol{\theta}) = \frac{1}{m_{\text{tr}}} \nabla_{\boldsymbol{\theta}} \sum_{i=1}^{m_{\text{tr}}} L(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)}, \boldsymbol{\theta})$
PARAMETER OPTIMIZATION FOR CONVOL. ENC.-DEC. (II-A.)
9: **while** stopping criterion not met **do**
10: Compute gradient estimate:
$\boldsymbol{g} \leftarrow \frac{1}{m'_{tr}} \nabla_{\boldsymbol{\theta}} \sum_{i=1}^{m'_{tr}} L\left(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)}, \boldsymbol{\theta}\right)$
11: Compute velocity: $\boldsymbol{v} \leftarrow \eta \boldsymbol{v} - \beta \boldsymbol{g}$.
12: Update parameters: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{v}$
13: **end while**
PIXEL-WISE SEGMENTAT. OF UNSEEN AERIAL IMAGE. (II-A.)
14: **Get** aerial image with TX_{lat}, TX_{lon}, RX_{lat}, and RX_{lon}
15: **Segment** image using optimized $\boldsymbol{\theta}$ parameters.
16: **Extract** categorical matrix $\boldsymbol{M}$ from line 15.
17: Overwrite $\boldsymbol{M}$ values as follows:
buildings $\leftarrow R_b$, open field $\leftarrow R_o$, street $\leftarrow R_s$.

B. Off-policy Model-free Algorithm for Optimal Path Search

In this section, we consider the use of ML to solve the optimal path search problem. The categorical matrix $\boldsymbol{M}$, generated in section II-A is set as the *environment* of our off-policy model-free algorithm. As an example, the possible scenarios revealed in Fig. 1(b) are considered. Assuming V_1 needs to establish a connection with V_2, it might select the shortest-distance path $\{V_1, V_2\}$ (transmission through buildings that affects the signal propagation loss directly), or hop through the neighboring radio side units (RSU) as $\{V_1, RSU_1, RSU_2, V_2\}$ (which do not have any obstruction). Further, assuming that V_3 requires to connect to V_4, transmission through the open field may be better than surrounding the obstacle. To address all possible paradigms and find the optimal propagation-based multi-hop policy, we represent our problem mathematically as a Markov decision process, consisting a finite set of states **S**, a set of actions **A**, a transition probability matrix **P**, and a set of rewards **R**. Fig. 1(b) illustrates vehicle V_1 taking a signal transmitting *action* a_t on the environment $\boldsymbol{M}$ [13]. The location of V_1 is defined as the *state* $s = (s_x, s_y)$ and V_1 can take one *action* $a \in A = \{\text{up, down, left, right}\}$ to hop to the next vehicle location. The spatial resolution of $\boldsymbol{M}$ is 0.3 m/pixel, which makes the four actions in A sufficient for finding paths in composite directions, reducing complexity. For our V2X learning task, we let V_1 transmit the signal one pixel at a time.

The obtained *reward* R, as a result of the action is defined as

$$R = \begin{cases} R_b & \text{for transmission across buildings,} \\ R_o & \text{for transmission across an open field,} \\ R_s & \text{for transmission across a street,} \\ \infty & \text{for reaching the end goal receiver.} \end{cases} \quad (7)$$

For each iteration epoch t, V_1 observes the state s_t and takes an action a_t, then the state transit into s_{t+1} and V_1 receives the reward R_t. The process of observing, selecting an action, and obtaining a reward is repeated and the agent V_1 learns a policy $\pi(s_t) \in A$ that maximizes the sum of the rewards obtained over a time period. In our scenario, maximizing the sum of the rewards involves minimizing the propagation loss experienced at the receiver vehicle. To evaluate the value of a state under the policy π, the *state-value function* can be utilized and is defined as follows.

$$V^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t)) | s_0 = s\right], \quad (8)$$

where $\gamma \in [0, 1)$ represents a discount factor and $E[\cdot]$ is the expectation operator. To find the optimal policy π^*, the following Bellman's optimality criterion can be used.

$$V^*(s) = \max_{a \in A}[R(s, a) + \gamma \sum_{s'} P_{s,a}(s')V(s')], \quad (9)$$

where $P_{s,a}(s')$ is the transition probability from state s to s' when action a is chosen. In this scenario, because $P_{s,a}(s')$ cannot be easily obtained, *Q-learning* is considered. For a policy π, the Q-value corresponding to the state and action pair (s, a) can be defined as,

$$Q^{\pi}(s, a) = R(s, a) + \gamma \sum_{s'} P_{s,a}(s')V^{\pi}(s'), \quad (10)$$

which is the expected discounted reward of executing action a at state s and following policy π thereafter. By setting

$$Q^{*}(s, a) = R(s, a) + \gamma \sum_{s'} P_{s,a}(s')V^{*}(s'), \quad (11)$$

$V^*(s)$ can be replaced by $\max_{a \in A} Q^*(s, a)$. According to [14],

$$\begin{aligned} Q_{t+1}(s, a) =& Q_t(s, a) + \alpha(R(s, a)) + \\ & \alpha\big(\gamma \max_{a' \in A} Q_t(s', a') - Q_t(s, a)\big). \end{aligned} \quad (12)$$

Here, α is the learning rate. With (12), Q-learning helps the agent learn the optimal Q-values recursively by obtaining state s_t and reward R_t and selecting an action a_t at each time t. The iteration will then converge to the optimal value V^* and policy π^*. An ϵ-greedy strategy is adopted to decide an action for each iteration. While the agent chooses a random action with a probability p to gather more information, it selects the best action given current information with a probability $1 - p$. Algorithm 2 provides the pseudo-code of the Q-learning task.

Algorithm 2 Optimal path search for multi-hop connectivity.

Input: Categorical matrix $\boldsymbol{M}$.
Output: Optimal $Q(s, a)$ function for multi-hop communication.
Q-LEARNING ITERATION (IN SECTION II-B.)
Set an arbitrary initial value for Q-table
Let s be the initial state of the matrix environment $\boldsymbol{M}$
for each episode **do**
  Initialize S
  repeat(for each step of episode)
    Choose a_t for current s_t by the ϵ-greedy strategy
    Take action a_t
    Observe the reward R_{t+1}, and the new state s_{t+1}
    $Q(s_t, a_t) \leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha\big[R + \gamma \max_a Q(s_{t+1}, a)\big]$
    $t \leftarrow t + 1$
  until $Q(s, a)$ converges or reach max. number of iterations.
end for
return $Q(s, a)$ ▷ Optimal state-action function

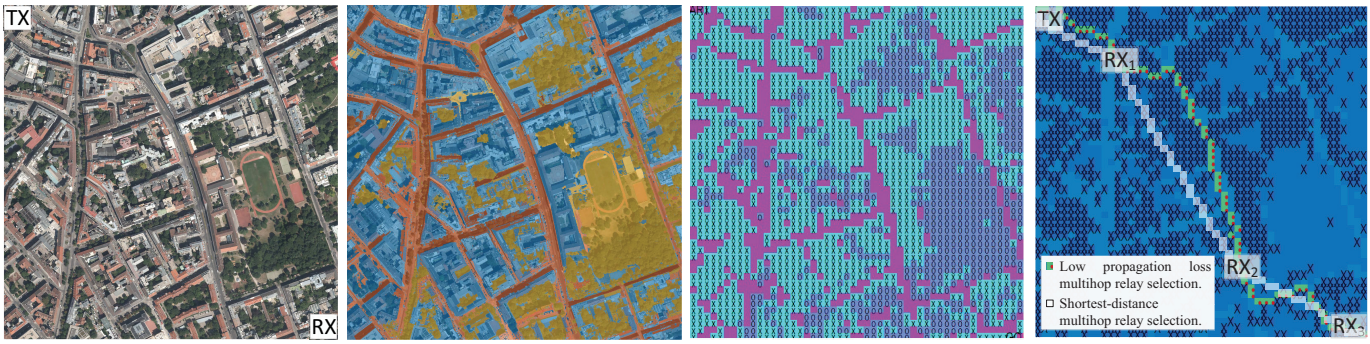
III. Performance Analysis and Results

The wireless scenario is simulated in MATLAB using an end-to-end IEEE 802.11p link under a V2X fading channel with additive white Gaussian noise for different distance scales. The network is configured without node signal amplification. The convolutional model is trained for 200 epochs, using a mini-batch size of 36, an initial learning rate of 75×10^{-3}, η of 9×10^{-1}, and $L2$ regularization factor of 5×10^{-4}. In our simulations, when β is set to a low value, the training time increases; conversely, when a high value of β is used, the training duration decreases at the expense of accuracy. After applying *grid search*, $\beta = 0.02$ is selected. In our experiments, an end-to-end pixel-wise semantic inference is achieved at 65ms per image. The first stage of our model is compared against well-known segmentation techniques. Our proposal increases the average *per-class* segmentation accuracy from 87.76% in FCNs to 90.8% (Table I). The complexity of the value-iteration algorithm in the Q-learning problem is $O(en)$, where e represents the total number of actions, and n is the size of the state space [15]. Fig. 2(a) shows an aerial image used for the validation of our model. Fig. 2(b) presents the semantic image obtained using our upgraded architecture. Fig. 2(c) shows the collection of all states s that constitutes the categorical matrix $\boldsymbol{M}$ down-sampled to a size of 50×50 pixels. Finally, Fig. 2(d) shows the path generated using our transmission policy π^* applying the off-policy model-free algorithm on the categorical matrix $\boldsymbol{M}$, with the transmitting vehicle located in the top-left corner, and the receiver in the bottom-right edge of the map. The reward values employed are $R_b = -50$, $R_o = -10$, and $R_s = -1$. Fig. 3 illustrates the central tendency of the received signal strength (RSS) at the receiver for both scenarios under study. We find that with a typical sensitivity of -98dBm at the receiver, the signal is lost at approximately 300 m when using the shortest-distance multi-hop, whereas the proposed lowest-propagation multi-hop strategy does not compromise the range until 500m

TABLE I
Segmentation Accuracy obtained with different Models*.

Classes	Texton Forests [5]	Patch Based [6]	FCNs [7]	Proposed Model**
Buildings	47.9%	86.7%	89.4%	**92.5%**
Open Field	45.8%	85.4%	86.5%	**90.1%**
Streets	40.1%	85.2%	87.4%	**89.8%**

*Trained using 4 NVIDIA GTX 1080Ti with local parallel pool.
**Convolutional encoder-decoder architecture, optimized with stochastic gradient descent with momentum.



(a) (b) (c) (d)

Fig. 2. Simulation results. (a) An aerial image unseen by our model, (b) the pixel-wise segmentation of our enhanced CNN encoder-decoder showing the buildings on blue, open fields in yellow, and streets in red, (c) the categorical matrix generated from the image segmentation (down-sampled to 50×50 pixels), (d) paths to three goal receivers generated with the shortest distance strategy [3], and with our multi-hop selection policy that minimizes the propagation loss.

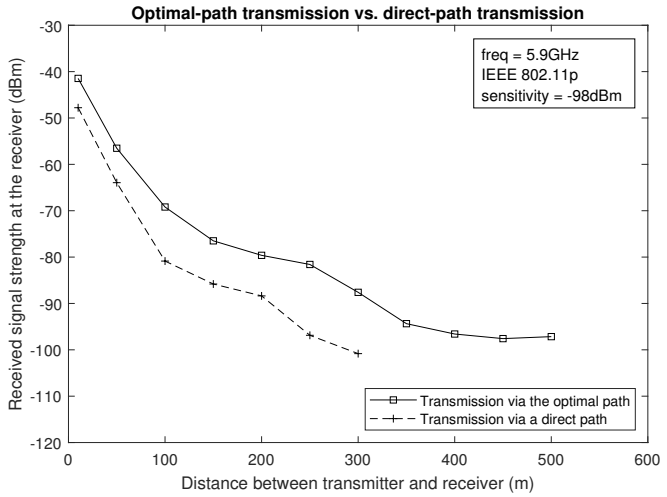


Fig. 3. Proposal performance and comparison. With a receiver sensitivity of -98dbm, the signal is lost around 300m when using a shortest-distance path, whereas the use of our lowest-propagation loss policy extends the range to approximately 500m. Our proposal attains a higher RSS at the receiver for all the range of distances when compared with the direct link.

approximately. These results establish that our system can extend the coverage of a V2X links by 66.7%. In addition, Fig. 3 reveals that for all distances, the RSS at the receiver is higher when the proposed multi-hop strategy is used.

IV. Conclusions and Future Work

The first stage of our proposal increased the average segmentation accuracy by 44.6%, 5.03%, and 3.04%, compared with the texton forest, patch-based, and FCN architectures, respectively. Simulations revealed that our selection policy improved the multi-hop reachability by 66.7% compared with the shortest-path selection strategy. Our solution could be further studied in device-to-device communications, nomadic nodes, network routing, path planning, etc. The frequency can be modified to 5G and beyond 5G millimeter wave ranges. Further work will include enhancing the segmentation model to include more precise categories (i.e., mountains, suburban environment, highways, etc.). The prospects of the agent to surround new obstacles, or discover policies that allows transmission across new obstacles to maximize reachability are of interest. Finally, we hope our findings will encourage researchers to fuse satellite imagery with 3-dimensional maps of cities (i.e., height of the terrain, buildings, trees, etc), to enable full machine-learning-driven channel characterization.

References

[1] G. Karagiannis *et al.*, "Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions," *IEEE Commun. Surveys & Tuts.*, vol. 13, no. 4, pp. 584–616, 2011.

[2] M. Tullsen, L. Pike, N. Collins, and A. Tomb, "Formal Verification of a Vehicle-to-Vehicle (V2V) Messaging System," in *Computer Aided Verification*. Springer, Cham, 7 2018, pp. 413–429.

[3] I. A. Abbasi *et al.*, "A Reliable Path Selection and Packet Forwarding Routing Protocol for Vehicular Ad hoc Networks," *EURASIP J. on Wirel. Comms. and Netw.*, vol. 2018, no. 1, p. 236, 12 2018.

[4] M. E. Morocho-Cayamcela *et al.*, "Machine Learning for 5G/B5G Mobile and Wireless Communications: Potential, Limitations, and Future Directions," *IEEE Access*, vol. 7, pp. 137 184–137 206, 9 2019.

[5] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 6 2008, pp. 1–8.

[6] D. Ciresan *et al.*, "Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images," in *NIPS Proceedings: Advances in Neural Info. Processing Systems*, 2012, pp. 2843–2851.

[7] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 4 2017.

[8] V. Badrinarayanan *et al.*, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2015.

[9] E. Maggiori *et al.*, "Can Semantic Labeling Methods Generalize to Any City? The Inria Aerial Image Labeling Benchmark," *2017 IEEE Int. Geoscience and Remote Sensing Symp. (IGARSS)*, pp. 3226–3229, 2017.

[10] ——, "High-Resolution Aerial Image Labeling with Convolutional Neural Networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 7092–7103, 12 2017.

[11] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *International Conference on Learning Representations (ICRL)*, pp. 1–14, 2015.

[12] A. Geron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to build Intelligent Systems*, 1st ed., N. Tache, Ed. O'Reilly Media, Inc., 2017.

[13] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. London, England: The MIT Press, 2018.

[14] Junhong Nie and S. Haykin, "A dynamic channel assignment policy through Q-learning," *IEEE Trans. on Neural Netw.*, vol. 10, no. 6, pp. 1443–1455, 1999.

[15] S. Koenig and R. G. Simmons, "Complexity analysis of real-time reinforcement learning," in *AAAI Proceedings*, 1993, pp. 99–107.