User Localization for Vision-Aided Wireless Communication

Katarina Vuckovic
Department of Electrical
and Computer Engineering
University of Central Florida
Orlando, USA
kvuckovic@knights.ucf.edu

***Abstract*—This work presents a framework for user localization in a single user vision-aided wireless communication (VAWC) system. VAWC systems are communication system that are equipped with cameras to provide visual awareness of the environment. The purpose of the proposed framework is to detect the user in the image captured by a camera mounted at the base station of the system and then convert the location of the user in the image into coordinates on the map. The first network is a object detection algorithm based on You-only-look-once (YOLO) that outputs a bounding box around the user in the image. This is followed by a multi-layer perception (MLP) neural network that converts the bounding box parameters output by the YOLO into x and y coordinates. The average localization error is of the entire framework is 0.9 m.**

***Index Terms*—Vision Aided Wireless Communication, Localization, YOLO, MLP, Computer Vision, Wireless Communication**

I. Introduction

Vision-aided Wireless communication (VAWC) is a new paradigm in wireless communication (WComm) that merges computer vision and wireless communication [1]. The idea of a VAWC system is to use cameras to provide the WComm system with visual awareness of the environment. Typically, the camera is mounted to the base station (BS) and it provides visual information regarding the users and other objects (blockers) in the environment as shown in Fig. 1. The BS then uses this information to optimize its performance. The next generation of WComm systems are moving towards mmWave and tetraherz multiple input multiple output (MIMO) antennas. This technology can enable faster connection speeds, lower latency and larger capacity [2]. However, it also comes with its challenges. The main issue is that mmWave and tetraherz signals are highly sensitive to blockage. They suffer from high penetration losses and attenuation, resulting in low signal-to-noise (SNR). When there is an object present between the user and the BS, the object acts as a blocker and severely attenuates the signal [3]. Therefore, these systems have an increased dependency on line-of-sight (LOS) signals.

VAWC emerged motivated by the LOS link dependency and the recent advances in computer vision and deep learning [4]. The intent of the computer vision part is to detect and track the users and blocker in the environment. This information is then used to tackle some the most important challenges in next generation WComm such as blockage prediction, beam selection, resource allocation, and proactive hand off [2]–[4]. These tasks improve the capability and reliability of the communication system.

Deep learning and computer vision networks require large datasets to train the network. The collection of real vision-wireless data is a laborious and expensive task. To address this issue, the authors in [4] developed Vision-Wireless (ViWi) dataset framework. The framework was developed using 3D modeling and ray-tracing software. It contains wireless and vision data for multiple scenes. This dataset framework is publicly available and its purpose is to provide a common vision-wireless dataset for VAWC research. The framework contains several different scenarios with single and multiple users. Furthermore, in addition to the RGB images and wireless data, the datasets also contain Light Detection and Ranging (LIDAR) images and user locations.

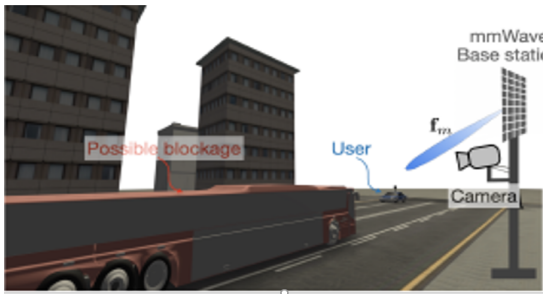

Fig. 1: Example of VAWC system for blockage prediction and beam selection.

This work focuses on the vision component of the VAWC system where the objective is to design a framework that detects the user in an image and uses that information to estimate its location on the map. The proposed framework consists of two neural networks. The first neural network uses the You Only Look Once (YOLO) object detection algorithm to create a bounding box around the user. The second neural network is a multi-layer perception (MLP) network translates the location of the bounding box parameters to the geographic location.

II. Background and Related Works

A. Computer Vision in Wireless Communication

Several surveys that discuss challenges and opportunities of using deep learning computer vision in WComm have been published in recent years [1], [5], [6]. However, the major enabler in WAVC was the deployment of the ViWi dataset framework [7]. The framework provided a common platform for researchers to benchmark performance and compare different solutions. Many researchers have taken advantage of the ViWi dataset to tackle the tasks in VAWC. In [8], the authors propose vision aided beam and blockage prediction for mmWave systems using 18-layer Residual Netork (ResNet-18). Next, the same authors introduce the idea of vision-aided beam tracking for mmWave systems and provide a baseline solution [4]. Beam tracking is accomplished by training a Residual Neural Network (RNN) to predict the future beam based on a beam-sequence. RNNs capture spatial-temporal correlations and have been shown to perform well on sequences. In [9], dynamic blockage prediction for a multi-user scenario is discussed. The proposed framework consists of a YOLO object detector and a RNN blockage prediction network. In [3], the authors expand on the blockage prediction idea and also propose a proactive hand-off model. Another beam selection method is proposed in [2] where a single stage object detector is utilized to detect the user. Next, a MLP network is trained to estimate optimal angle of the beam based on the bounding box parameter.

B. Object Detection in Computer Vision

YOLO object detection algorithm, first proposed by Redmon et. al. in [10] and later improved to YOLOv2 in [11], has demonstrated to be a fast and accurate object detection neural network. Since then, there have been incremental improvements to YOLO with version 3 published in 2018 [12] and version 4 published in 2020 [13]. In this work, YOLOv2 is utilized.

The main idea in YOLO is that the entire image is processed only once using a single CNN. This is why the algorithm is called YOLO and why it is so fast. The second advantage is that it learns general representations, which means that it is less likely to break down when applied to new domains and unexpected inputs [10]. The YOLO algorithm typically uses a pretrained network that is trained on a very large dataset to learn the general representations. Then it is fine-tuned on a smaller dataset that consist of images of interest which, in our case, are images captured by the cameras at the BS.

The YOLO network takes an input image and outputs the object class (if multiple classes exist), bounding box parameters $(x_{min}, x_{max}, y_{min}, y_{max})$, and the confidence score. If multiple objects exists, it classifies all objects in a single shot. The YOLO algorithm divides the input image into grid cells. The grids are labeled depending on weather the grid cell contains the object or not. Furthermore, the center of the bounding box is determined. If the center of the bounding box falls into a grid cell, then that cell is responsible for detecting the object. Each cell can predict B number of bounding boxes with a confidence score for each for the boxes, where B is a parameter set by the designer. The confidence interval associated with each bounding box reports how confident the YOLO algorithm is that it correctly detected the object in the bounding box. Finally if multiple bounding boxes exist, YOLO uses the confidence score to decide which bounding box to keep.

III. System Model and Dataset

A. System Model

This section described the VAWC system model and dataset used in the simulations. Consider a single BS and a single user in the environment. As shown in Fig.2, the BS is located on the side of the road and it has three cameras mounted on it. The front camera has a field of view (FOW) of 110°, while the side cameras have a FOW of 75°. The system contains a single user that is moving along the 5 trajectories depicted in with green horizontal lines. Each line contains 1000 positions, resulting in a total of 5000 user positions. The length of the of the green lines in 90 m.

Furthermore, with respect to the wireless system, the BS and user are communicating using a typical MIMO-Orthogonal Frequency Division Multiplexing (OFDM) Wireless Network. The BS is equipped with a mmWave massive MIMO antenna, while the user uses a omni-directional antenna. The operating frequency, bandwidth the number of antenna array elements at the BS is are set by the designer in the data generation framework.

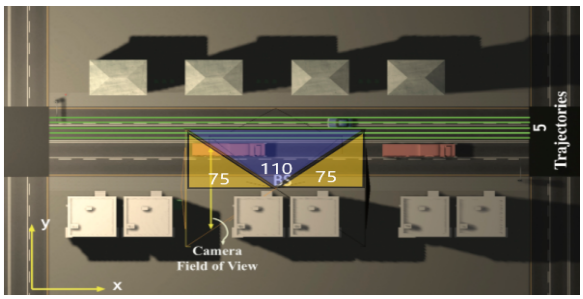

Fig. 2: Environment map showing the location of the BS, the orientation and the FOW of the cameras, the user's trajectories and the location of the buses that block the LOS link for some user's location.

B. ViWi dataset

Two scenes from the available in the ViWi framework are considered. One contains two buses that block the LOS view from the BS to the user for some user locations, as shown in Fig. 2, while the other scene is the same, except without the buses. Therefore, all user locations are in LOS view. We combine the datasets from the two scenes to increases the total size of the dataset. From this dataset a few different subsets are generated for training and testing of the two networks. The subsetes generated for YOLO and MLP are discussed in the next section.

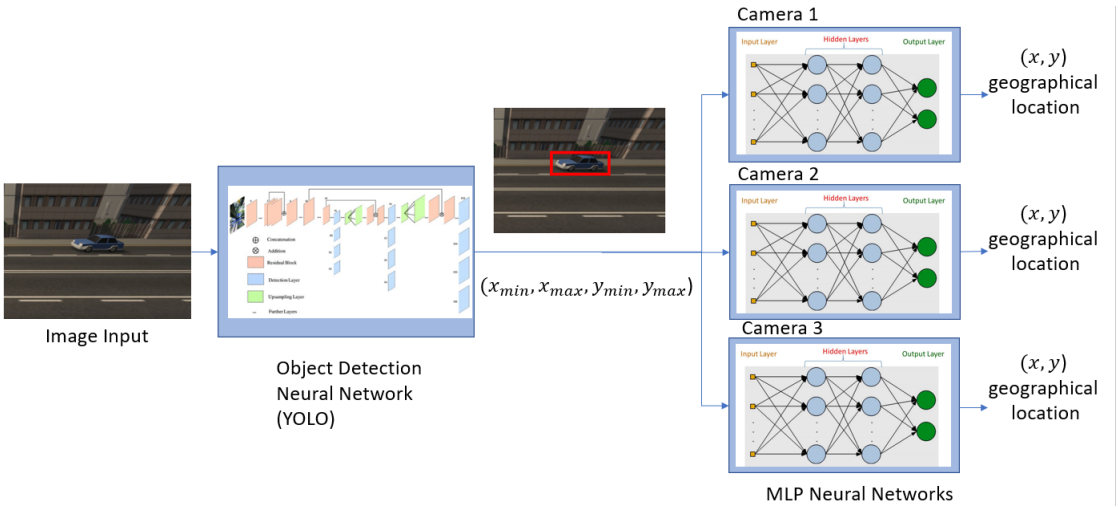

Fig. 3: Object Detection and Localization Framework

Each sample in the dataset has a (x,y) location paired with an image that captures the user. The image comes from one of the three cameras depending on where the user is located.

IV. User Detection and Localization Method

This section describes the framework design. Furthermore, it describes the neural network architectures of the two components that comprise the frameworks. The section also discusses the different dataset subsets used to train and test the two framework components as well as the dataset used for the entire framework.

A. Framework

The user detection and localization framework consists of two component. The block diagram of the framework is shown in Fig. 3. The first component is a YOLO object detection algorithm. The object detector takes the image captured at the BS as an input and outputs a bounding box surrounding the user in the image. The bounding box is defined by four parameters $(x_{min}, x_{max}, y_{min}, y_{max})$. The object detector also outputs a percentage indicating the confidence of the bounding box. The second component is a MLP neural network that takes the bounding box parameters of the YOLO output and translates them to a location on the environment map defined by x and y coordinates. Since there are three different cameras mounted on the BS that cover the different FOW, a separate network is trained for each camera.

B. YOLO Object Detection

The dataset used for YOLO object detection is a mix of images from the three different cameras. A total of 300 images were labeled manually using *imageLabeler* tool in MATLAB. The tool was used to create bounding boxes around the user in each image. To increase the dataset, the labeled images were augmented as shown in Fig. 4. The augmentation in performed by adding jitters to the image color and by horizontally flipping the image.

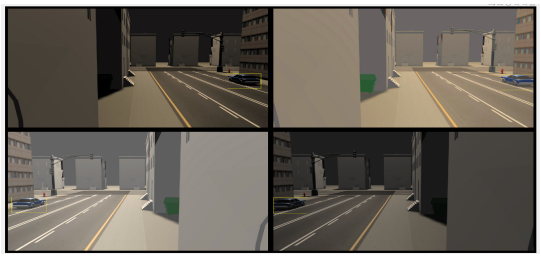

Fig. 4: Data Augmentation to Increase the YOLO Training Dataset

The parameter used to train the YOLO model are listed in Table I. The inputs to the YOLO training model are the images, the bounding box parameter that surround the user in the image and the number of classes. In this case the number of classes is trivial as there is only one class which is the user represented by a blue car. The bounding box parameters are set to 4, therfore each image in the dataset is tagged by 4 parameters that define the bounding box.

The YOLO model is first pre-trained using RestNet-50 model which is a convolutional neural network (CNN) with 50 layers. This network is trained on over a million images from the imageNet dataset and a pre-train version is available in MATLAB [14]. CNNs require a lot of data to train robust models that can learn various features. Therefore, a deep pre-trained model is used as a start for YOLO and then using the training dataset which is typically significantly smaller than the dataset used for the re-trained model. The model is re-trained and tuned for object detection on a dataset that contains images of interest. There are several acceptable options for pre-trained models. ResNet-50 was selected as it is one of the common pre-trained models used.

YOLO tunes the pre-trained models by replacing layers after *'activation_40_relu'* with a detection sub-network.The sub-network is trained to extract the features specific to the ViWi dataset. The network is trained using Stochastic Gradient Descent (SDG) optimization function with a starting learning rate of 0.001, the batch side of 16 and maximum

number of epochs of 30.

TABLE I: YOLO Training Parameters

Input size (size of image)	224x224x3
Number of Classes	1 (user)
Bounding Box Parameters	4
Retrained Network	ResNet-50
Feature layer	'activation_40_relu'
YOLO optimization function	Stochastic Gradient Descent
Initial Learning Rate	0.001
Mini batch Size	16
Number of epochs	30

C. MLP Regression

The second part of the framework is a network that relates the bounding box parameters to (x,y) coordinates on the map. This resembles the *fingerprinting* localization technique in wireless communication [15], where each location is tagged with some wireless measurement (received signal strength (RSS) or Channel State Information (CSI)) and the a neural network is trained to estimate the location based on new measurements. In this case, the wireless measurements are replace with bounding boxes from the object detection algorithm.

Since, there are three cameras covering different FOWs, the system requires a separate network for each camera. To train the networks, the input required are bound box parameters and true user locations. The dataset generated from *imageLabeler* to train YOLO is insufficient to train an MLP as it contains only 300 images (for all three cameras). As such, a new dataset was generated by passing 4893 images from all three cameras the ViWi Dataset to the YOLO object detection to obtain the bounding boxes. The dataset was then split into three subsets based on the camera that the image belong to. Camera 1, 2 and 3 subset contain 1441, 2000, and 1452 samples, respectively. The same training MLP network was used for all three cameras. The MLP architectures consists of three fully connected (FC) layers of 1024 nodes and a ReLU activation function is followed after every FC layer. A fully connected layer and a softmax layer are at the output layer of the network. Furthermore, the network is trained using the Adam optimized with a initial learning rate of 0.0001 with batch size of 10 and maximum number of epoch of 50. The MLP training parameters are summarized in Table II

TABLE II: MLP Training Parameters

Input size	4 (bounding box parameters)
Output size	2 (x,y coordinates)
Hidden Layer	3
Number of nodes in each hidden layer	1024
Optimizer	Adam
Initial learning Rate	0.0001
Batch Size	15
Number of epochs	50

V. Results and Discussion

This section presents the results for each part of the framework as well as the entire framework. First, the output and results from the YOLO object detection is discussed. Next, the testing results from MLP networks are discussed and finally the performance of the entire framework (YOLO+MLP) is presented and compared to the performance of the different parts.

A. YOLO Object Detection

The output of the YOLO object detection is the input image with a bounding box surrounding the user as shown in Fig. 5. Furthermore, the object also outputs a score that represents the confidence of the object detection. In the example in the figure, the confidence is 64%.

Fig. 5: YOLO object detector output

To evaluate the performance of the object detection algorithm the intersection over union (IOU) is computed. The IOU is a ratio of intersection area between the actual and estimate bounding box over the area actual bound box. When the estimate bound box is exactly the same as the true bounding box, the IOU is 1 and when there is no intersection between the two bounding boxes, the IOU is 0. Typically, if the IOU is above some threshold (i.e. 0.5), the detection is considered to be correct. Fig. 6 shows the cumulative distribution function (CDF) of the IOU tested on a new dataset with 100 samples (all different from the training/validation dataset). As may be seen from the CDF in the figure, only 1% of the dataset has a IOU less than 0.5. Assuming a threshold of 0.5, we conclude that 99% of the dataset was classified correctly.

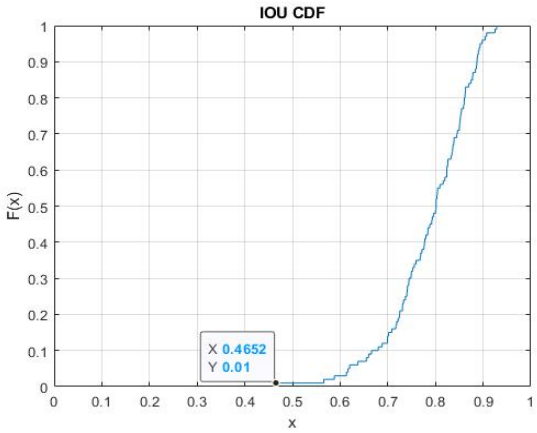

Fig. 6: CDF for the IOU test of the YOLO Model

Additionally, the recall and precision are computed on the validation samples. Precision is a measure of the ability to

identify only the relevant object, while recall is a measure of the ability to model all relevant cases. Precision and recall are computed as

$$precision = \frac{TP}{TP + FP} \tag{1}$$

$$recall = \frac{TP}{TP + FN} \tag{2}$$

where TP is true positive, FP is false positive and FN is false negative.

The recall vs. precision curve is plotted as shown in Fig.7, then the average precision is calculated as the area under the curve, which in this case is 0.97.

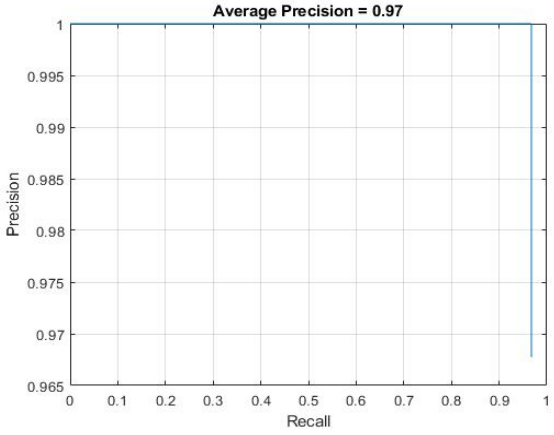

Fig. 7: Average Precision Curve

B. MLP Network

The MLP neural networks were tested on the same dataset as the YOLO. The CDF of the localization error for the three different cameras (MLP models) is shown in Fig.8. The average accuracy for camera 1, 2, and 3 is 10.6 m, 0.9 m, and 1.9 m, respectively. The total average error for the entire dataset is 3.32 m. The testing results show that the accuracy of camera 1 is significantly lower than for camera 2 and 3. There are a few explanations for this. First, the dataset used for MLP training of camera 1 is smaller than the datasets for 2 and 3. In fact, the dataset for camera 2 with 2000 samples is the largest dataset, therefore, the accuracy is also the best.The second reason for low performance of camera 1 is that both testing and training datasets for camera 1 contain images where the user is only partially present in the image. These samples were correctly detected by YOLO but since the bounding boxes size and shape were significantly different than all the other bounding boxes and there was not enough data samples of this sort for the MLP to learn the features of these instances, the MLP outputs a result that poorly estimates the location of the user. These outliers significantly impacted the overall results of camera 1. Lastly, the MLP was trained with a dataset generated from processing the images using the YOLO dataset, rather than manually labeling them (as is the case with the testing dataset). Therefore, the MLP may have been biased to better detect YOLO processed bounding boxes rather than the true bounding boxes.

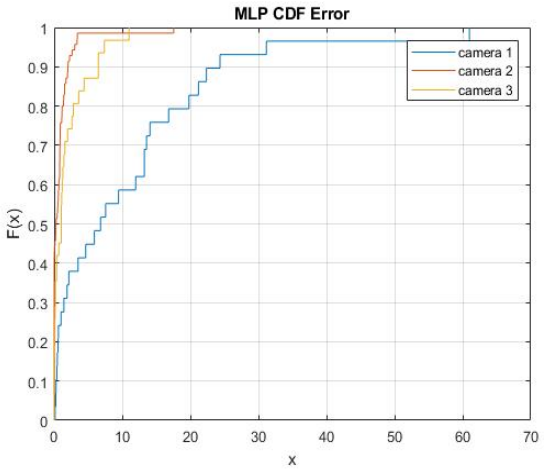

Fig. 8: CDF for MLP localization results separated by cameras

C. Localization Framework

Finally, we combined the YOLO and MLP to test the end-to-end framework. The test is performed on the same dataset as the YOLO and MLP. The CDF error for each of the three cameras is shown in Fig.9. The average error for camera 1, 2, and 3 is 1.67 m, 0.39 m, 0.85 m, respectively. The total average error over all three cameras is 0.9 m. Comparing the results of MLP to the results of the localization framework, the localization error improved significantly for the entire framework. Especially, the results for camera 1. This confirms that the MLP is biased to better perform on YOLO processed bounding boxes rather than true labels. This is also the preferable case as the end-to-end framework should have the best possible result while the performance of the subsystems composing the framework is not as important.

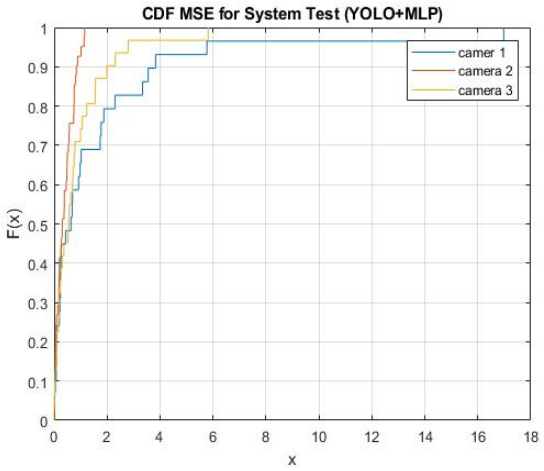

Fig. 9: CDF for Localization framework (YOLO+MLP) seperated by cameras.

VI. Conclusion

In this project an end-to-end framework for localizing the user in a VAWC system was presented. The proposed framework is designed for a single user system, where the framework not only detects the user but also estimates its location on the map in terms in x and y coordinates. The designed framework consists of two components, the first is the object detection part that uses YOLO object detection to create a bounding box surrounding the user in the image. The second part is a MLP network that translates the bounding box parameters to the x and y location coordinates. The average error of the framework is 0.9 m.

In VAWC systems, computer vision is only one part of the system, the second part is the communication system parameters. Therefore, the next step is to merge the wireless data with the computer vision to tackle one or more challenges in mmWave MIMO. Some of the applications that may be tackled in future work include but is not limited to blockage detection, beam prediction, or proactive hand-off.

References

[1] Takayuki Nishio, Yusuke Koda, Jihong Park, Mehdi Bennis, and Klaus Doppler, "When wireless communications meet computer vision in beyond 5g," 2020.

[2] Ziqiang Ying, Haojun Yang, Jia Gao, and Kan Zheng, "A new vision-aided beam prediction scheme for mmwave wireless communications," in *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, 2020, pp. 232–237.

[3] Gouranga Charan, Muhammad Alrabeiah, and Ahmed Alkhateeb, "Vision-aided 6g wireless communications: Blockage prediction and proactive handoff," 2021.

[4] Muhammad Alrabeiah, Jayden Booth, Andrew Hredzak, and Ahmed Alkhateeb, "Viwi vision-aided mmwave beam tracking: Dataset, task, and baseline solutions," 2020.

[5] Yu Tian, Gaofeng Pan, and Mohamed-Slim Alouini, "Applying deep-learning-based computer vision to wireless communications: Methodologies, opportunities, and challenges," 2020.

[6] Yuwen Yang, Feifei Gao, Chengwen Xing, Jianping An, and Ahmed Alkhateeb, "Deep multimodal learning: Merging sensory data for massive mimo channel prediction," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 1885–1898, 2021.

[7] Muhammad Alrabeiah, Andrew Hredzak, Zhenhao Liu, and Ahmed Alkhateeb, "Viwi: A deep learning dataset framework for vision-aided wireless communications," 2020.

[8] Muhammad Alrabeiah, Andrew Hredzak, and Ahmed Alkhateeb, "Millimeter wave base stations with cameras: Vision aided beam and blockage prediction," 2019.

[9] Gouranga Charan, Muhammad Alrabeiah, and Ahmed Alkhateeb, "Vision-aided dynamic blockage prediction for 6g wireless communication networks," 2020.

[10] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," 2016.

[11] Joseph Redmon and Ali Farhadi, "Yolo9000: Better, faster, stronger," 2016.

[12] Joseph Redmon and Ali Farhadi, "Yolov3: An incremental improvement," 2018.

[13] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.

[14] "Deep network designer," .

[15] Quoc Duy Vo and Pradipta De, "A survey of fingerprint-based outdoor localization," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 491–506, 2016.