# UWB Radar Applied to Lane Occupation and Vehicle Classification

Marcelo Bender Perotoni, Claudio José Bordin Jr., Fernando de A. Castilho, and Gustavo Y. M. Vieira.

***Abstract*—This article describes the use of a commercial UWB radar for vehicle classification and lane occupation detection using real-world data acquired in an urban environment. We compare two radar image processing schemes: one based on deep learning using raw data produced by the radar, and a second method employing machine learning algorithms, such as KNeighbors and Naïve Bayes, using features extracted from raw data. We verify experimentally that both schemes lead to reasonably accurate estimates without the need of large training sets.**

***Index Terms*—UWB; 5-11 GHz Radar; Vehicle classification; Machine Learning; Deep Learning.**

## I. Introduction

The concept of ultra-wide band (UWB) was first proposed in the Los Alamos National Laboratory, in 1990, and is defined as a signal whose bandwidth is at least 25% of its central frequency [1]. It has been proposed as a way to offer micro-location, complementing global positioning system (GPS) services, which allows outdoor location and have become a default in electronic appliances. The Micro-location is also achieved using other alternatives, such as Bluetooth low-energy (BLE) or WiFi, but in contrast to these two techniques, which depend on received signal strength indicator (RSSI) to determine positions, UWB is based on temporal measurements performed on short pulses in the time domain [2]. The short pulse implies a large bandwidth, so usually UWB requires bandwidths above 500 MHz. Particularly in intelligent transport systems (ITS), UWB offers advantages because of the nature of its location is less sensitive to spoofing [3]. Security is crucial in ITS applications due to the high risk involved in vehicles at high speeds. UWB radar, in turn, involves a different approach in comparison to its counterparts based on classical narrowband signals, since the emphasis is on the time-domain shape of the echo return [4], which is more difficult to be hacked and interfered with. UWB radars are reported in several applications, for instance, in [5] target classification is performed by directly correlating, in the time domain, the measured echo with that of different types of targets (e.g., metal and wood plates, cars, etc.). With the goal of counting people, an impulse radio (IR) UWB was tested with two Novelda radars and Raspberry Pi 2 modules, using high-gain antennas (10° azimuth angle) [6]. A commercial Salsa Ancho UWB radar was used to solve a robot-tracking simultaneous localization and mapping (SLAM) problem, by applying a bio-mimetic approach, similar to the one used by bats with sound waves [7]. A Salsa Cayenne UWB radar was employed as a sensor to characterize metal defects based on the return echo by means of the principal component analysis (PCA) in [8]. Tracking multiple targets with multistatic UWB radars using triangulation was presented in [9], with a frequency range of 7 GHz, in an area of 100 $m^2$. The push towards UWB in vehicles was reported back in 2007 [10], where a 90-nm complementary metal–oxide–semiconductor (CMOS) technology was used as basis for a 22-29 GHz UWB chip, conceived for the specific use in short-range anti-collision vehicular radars.

In this work, the development of an optimal vehicle classifier using classical detection theory [11] would be prohibitively complex due to the multitude of possible targets. With this in mind, we resort to machine learning techniques [12]. In [13] a Novelda X2 radar chip signal was used for robot visualization, intended to smart manufacturing applications using a convolutional neural network (CNN). A CNN method, also using UWB signals, was shown to outperform traditional threshold localization methods and CNN-only techniques by simulation-only results using a three-receivers and one-transmitter scenario [14]. In [15], CNNs and other machine learning methods were employed for indoor object classification from UWB radar images. Advantages of UWB for general radar target recognition were covered in [1], with either classical (e.g., based on impulse and natural responses of the target) or machine learning based methods [16].

This article innovates by using real-world data produced by a commercial UWB radar in an ITS application that comprises identification of the occupied lane and vehicle type with three different classes (motorcycles, cars, and trucks). We also innovate by comparing the performance of CNNs and machine learning algorithms [12], such as KNeighbors and Naïve Bayes, using a simple feature extraction scheme for the classification tasks. Differently from [16], the techniques proposed in this article do not demand multiple successive radar measurements to operate and, once trained, are more suitable for real-time applications.

The remainder of the text is organized as follows: The hardware is presented in terms of building blocks (Section II), as well as the data acquisition methodology. Then, the scenario in which data was collected (Section III) and the proposed classification approaches (Section IV) are described. Next, the results (Section V) are discussed and some conclusions are drawn (Section VI) in regard to the particularities of the observed experiment. Finally, an appendix discusses the conversion of the raw exported data to actual length units.

M. B. Perotoni and C. J. Bordin Jr. are with Universidade Federal do ABC (UFABC), Santo André, SP, Brazil (email: marcelo.perotoni, claudio.bordin@ufabc.edu.br).

Fernando de A. Castilho and Gustavo Y. M . Vieira are with Flexmedia, São Paulo, Brazil (email: fernandocastilho, gvieira@flexmedia.com.br)

Digital Object Identifier: 10.14209/jcis.2023.3

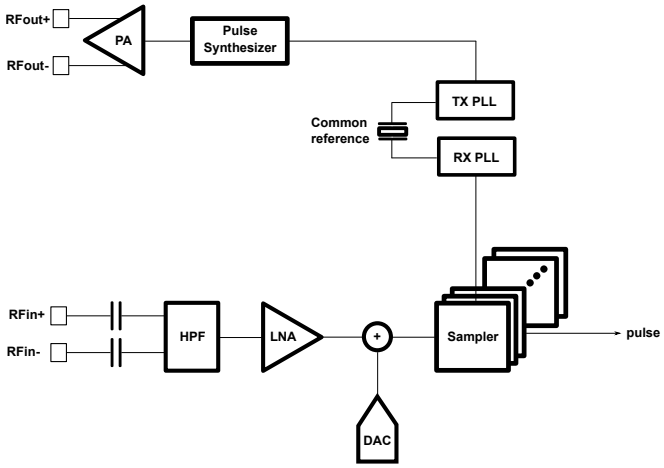


Fig. 1. Block diagram of the RX and TX branches of the UWB IC.

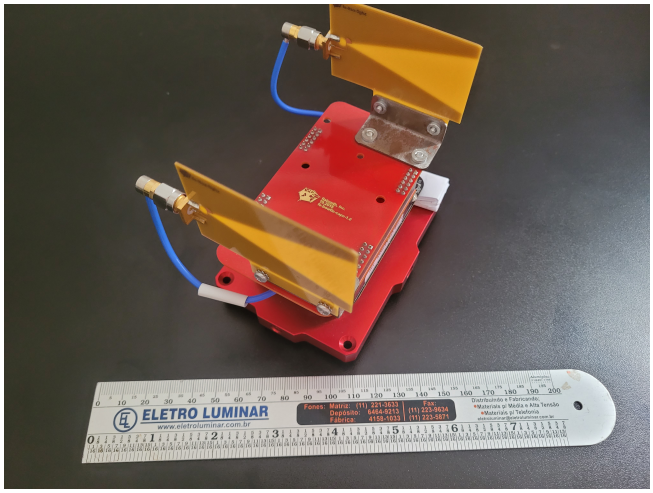


Fig. 2. UWB Radar used in the test along a ruler for size reference.

## II. Hardware

The deployed UWB radar is a commercial unit named Salsa Onza, based on the Xe Thru X4 System on a Chip (SoC). This specific unit is not sold anymore, but a similar version (named SLMX4), operating in a lower frequency range (1.4 GHz to 7.29 GHz) costs USD 425. A board containing the X4 chip can be found under USD 100, leaving the developer with the need to interface the digital pins that receive the reflected pulses. Salsa Onza follows a concept called Strobed Sampling, where the electromagnetic echo is sampled after a determined time offset. The integrated circuit (IC) transmits a maximum output power of -14 dBm, compliant to both Federal Communications Commission (FCC) and European Telecommunications Standards Institute (ETSI) standards, and can achieve a nominal spatial resolution of 6.4 mm, using a sampling rate of 23.3 GS/s. The IC block diagram of the radio frequency (RF) front end is depicted in Fig. 1, where the differential input and output ports (100 Ω) connect to external antennas/amplifiers/filters and, the received pulse, after converted to the digital domain by 1536 samplers, is further down-converted and decimated. The digital-to-analog converter (DAC) in the middle of the block diagram controls direct current (DC) offsets in the received pulse. Along the transmitter side, pulses in time domain are generated at the pulse-repetition frequency (PRF) of 60.75 MHz. Time coherence is assured by using the same reference for both transmitter and receiver branches, symbolized by a crystal symbol in the diagram. Communication with external circuits is made by means of an serial peripheral interface (SPI) with 4 or 6 wires.

A BeagleBone [17] board connects to the radar IC board, which in turn communicates to a host computer by means of USB or an Ethernet RJ45 connector. The frequency range of this particular unit ranges from 5 to 11 GHz, and it has two planar linear tapered slot antennas, whose gains can be estimated around 9 dB on the average in the frequency range. The complete radar is shown in Fig. 2.

The radar board has a native application programming interface (API), with Matlab and Python wrappers. However, we used instead its browser application named SalsaScope, which captures the reflected pulse and enables setting some of the radar internal registers from an ordinary web browser. The application shows in real time the reflected signal (vertical axis proportional to voltage and horizontal to time or space). The reflected signal is exported as a CSV file, with the format shown in Fig. 3. Basically, data is organized into a matrix $M$, with columns (bins) representing the space scale (known in radar jargon as *fast time*) and rows (frames) representing the time scale (known as *slow time*). The number of bins (1536) is defined by the number of board internal samplers and corresponds to a maximum range of approximately 9 meters. The number of frames, in turn, is proportional to the acquisition time. Fig. 3 also shows the visualization of a real target, in movement. Notice that there is a constant clutter [18] on the left part of the figure. At the plot center, target reflections are visible, moving leftwards.

Besides commercial UWB radars, the literature reports other options where UWB pulses are generated at the circuit level. A popular technique employs non-linear devices such as step-recovery diode (SRD) [19], [20], [21], with help of transmission lines and P-type intrinsic N-type (PIN) switching diodes. In case a foundry is available for microelectronic processing, CMOS circuits have been used for the fast pulse generation, employing pulse compression implemented with digital logic blocks [22], or analog circuits operating in baseband [23].

## III. Data Acquisition Methodology

The radar was orthogonally positioned across a three-lane (named L1 to L3) urban avenue, each lane 2.9 meters wide, as shown in Fig. 4, and the return signal generated by each vehicle was recorded in CSV files. We performed a first visual analysis of the acquired signals and discarded measurements containing stray echoes from secondary vehicles.

Another issue affecting the proposed methods is the presence of vehicles outlying dimensions, such as large VW Vans that should be classified as a car and large Harley-type motorcycles. Their return signals occupy ranges close to larger vehicles, hampering the classification. Fig. 5 displays the echo

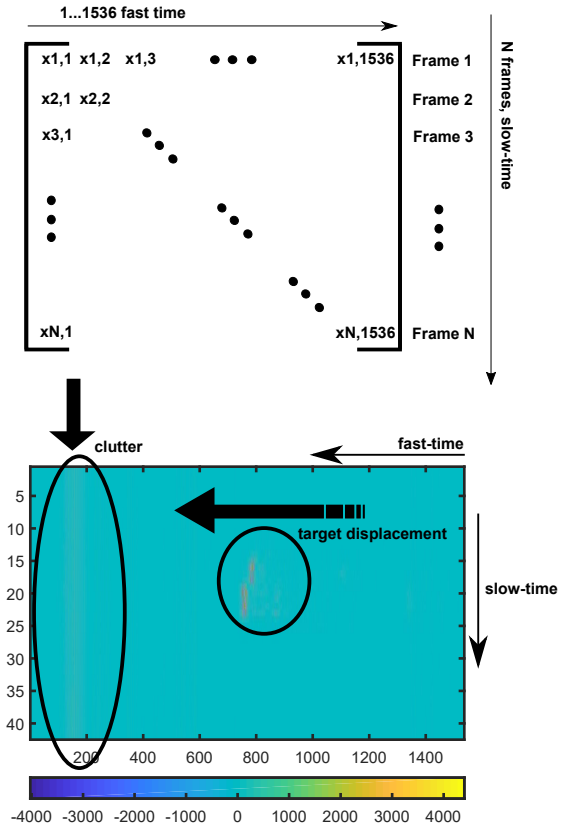


Fig. 3. Top: radar exported dataset format, forming a matrix $\boldsymbol{M}$. Bottom: example of a real vehicle echo signal.

Fig. 4. Illustration of the data acquisition setup. Lanes L1 to L3 are progressively far from the radar.

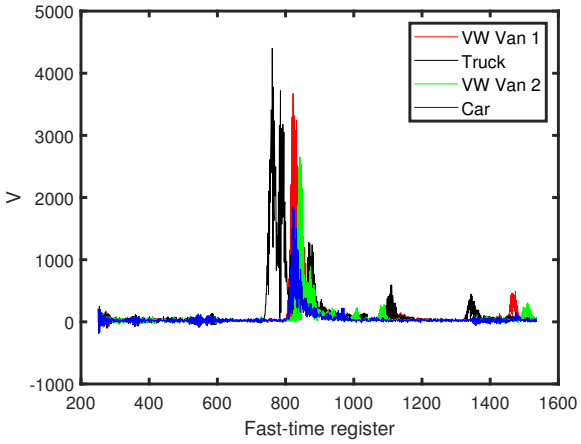


Fig. 5. Comparison of the frames for two VW Vans, a truck and a car.

(i.e., the row of $M$ with the maximum amplitude) due to a VW Van. Note that the return signal is much larger than that of cars on the same lane. Large SUV (sport utility vehicles) are also prone to generate misclassifications. To combat this issue, we discarded all measurements that did not correspond to vehicles that could not unequivocally be classified as cars, motorcycles, or trucks.

After those initial screening procedures, we kept a total of 112 vehicles echoes, with true classification as shown in Table I. There is a larger proportion of cars in contrast to heavier vehicles since measurements were made in an urban area due to security and safety concerns.

TABLE I
NUMBER OF MEASUREMENTS PER CLASS

| Lane | Motorcycles | Cars | Trucks |
|---|---|---|---|
| L1 | 3 | 29 | 1 |
| L2 | 5 | 29 | 4 |
| L3 | 9 | 29 | 3 |

For illustrative purposes, we show in Fig. 6 the waveforms corresponding to the row of matrix $M$ with the maximum amplitudes for three distinct moving targets (trucks). The vertical axis contain a variable ($V$) proportional to the voltage, whose absolute value is not defined by the instrument datasheet. Its horizontal axis represents the distance from the radar, and spans the 1536 registers responsible for the fast-time variable. An initial part encircled by a blue circle in the plot contains clutter due to the antenna and structural reflections (also shown in the matrix plot shown in Fig. 3). We observed that the first 250 of the 1536 samples were always contaminated by clutter, so these samples were discarded. It can also be seen that, as the targets move away from the radar, their return amplitude decreases and the relative position of the fast-time registers moves towards larger indexes.

## IV. CLASSIFICATION APPROACHES

Given the signals acquired as described in Section III, we aim to estimate the type of the passing vehicle, defined as one of the following: car, motorcycle or truck, and the number of the lane occupied by the vehicle, from 1 to 3. The optimal, Bayesian solution to this estimation problem requires knowledge about the radar pulse shape, the noise statistics, and the signatures of the targets, which are unavailable and cannot be adequately estimated from the measurements due to their limited number. With this in mind, we propose two distinct, machine learning based supervised approaches to the solution of this problem as described in the sequel.

### A. Classification via Deep Learning

In a first approach, we trained a CNN to jointly estimate the occupied lane and the type of the vehicles. The network receives as input $133 \times 1286$ raw matrices produced by the

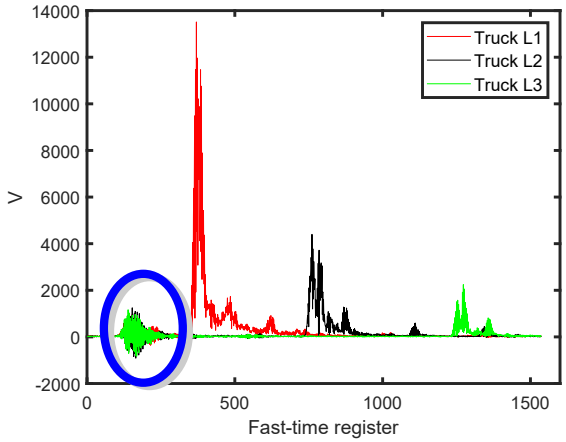


Fig. 6. Example of frames for three different targets. L1–L3 represent the lanes progressively farther from the radar.

radar and outputs a 9-entry vector with the probabilities of each class in the set {'lane 1', 'lane 2', 'lane 3'} × {'car', 'motorcycle', 'truck'}, were × denotes a Cartesian product.

The network structure that led to the most accurate results is summarized in Table II and depicted in Figure 7. This structure was determined empirically: we started from a simpler structure, with fewer filters and fewer convolutional layers. The number of filters and layers was then gradually increased and the dimensions of the layers were adjusted using the pooling layers to keep the number of trainable parameters small. We also verified that average pooling layers led to better performance compared to the more usual max pooling layers.

The network was implemented in Python using the Tensorflow 2 framework [24]. The convolutional layers were initialized with zero biases and random weights, drawn from a Xavier uniform initializer [25]. The last layer is a flat, dense layer and was initialized with zeros weights and biases.

We employed the Adam adaptation algorithm with the default learning rate ($10^{-4}$) and the sparse categorical cross-entropy loss function. Training employed 100 epochs but could be interrupted earlier if the accuracy, evaluated using the own training set, did not improve for more than 20 successive epochs; in this case, the weights that led to the highest accuracy were restored.

### B. Classification via Machine Learning

As a second approach, we evaluated the performance of traditional machine learning methods in the solution of the same classification problems. To deploy those methods, each radar brute image $\boldsymbol{M}$ is mapped onto features as follows. First, we select the row of the matrix $\boldsymbol{M}$ with largest energy, i.e., in Matlab-like notation $\boldsymbol{m} \triangleq \boldsymbol{M}^2[:, \hat{\imath}]$, where

$$\hat{\imath} = \arg\max_i \sum_j [\boldsymbol{M}^2]_{i,j} \ , \tag{1}$$

where $[\cdot]_{i,j}$ denotes the element with the given indexes, and the squaring operation is applied elementwise.

TABLE II
DEEP LEARNING STRUCTURE FOR TARGET CLASSIFICATION

| Layer | Size | Input Shape | Parameters |
|---|---|---|---|
| Input | - | $133 \times 1286$ | - |
| 2-D Convolution (8×) | $4 \times 8$ | $133 \times 1286$ | 264 |
| ReLu | - | $8 \times 133 \times 1286$ | - |
| Average Pooling | $4 \times 8$ | $8 \times 133 \times 1286$ | - |
| 2-D Convolution (16×) | $4 \times 8$ | $8 \times 33 \times 160$ | 4112 |
| ReLu | - | $16 \times 33 \times 160$ | - |
| Average Pooling | $4 \times 8$ | $16 \times 33 \times 160$ | - |
| 2-D Convolution (32×) | $4 \times 4$ | $16 \times 8 \times 20$ | 8224 |
| ReLu | - | $32 \times 8 \times 20$ | - |
| Average Pooling | $4 \times 4$ | $32 \times 8 \times 20$ | - |
| Flatten | - | $32 \times 2 \times 5$ | - |
| Dropout (p=0.5) | - | 320 | - |
| Dense | - | 320 | 2889 |
| SoftMax (Output) | - | 9 | - |

The vector $\boldsymbol{m}$ can be interpreted as an (unnormalized) power density of the target echo. To summarize the information provided by this density, we used as features its first two moments, calculated as

$$m_0 = \sum_i [\boldsymbol{m}]_i \ , \qquad m_1 = \sum_i i \frac{[\boldsymbol{m}]_i}{m_0}. \tag{2}$$

We verified experimentally that the use of additional features, such as higher order moments, the whole vector $\boldsymbol{m}$ or subsampled versions of it, did not increase performance.

Prior to being used in training and testing, the features are normalized using the Scikit-learn's function *MaxAbsScaler* [26], which maps each feature independently into the interval $[-1, 1]$. Alternative scaling methods led to similar performances.

We deployed the following machine learning algorithms for classification [12], [27]: KNeighbors (using 1 to 10 nearest neighbors), Linear Discriminant Analysis, Gaussian Naïve Bayes, Gradient Boosting Classifier, Random Forest, Decision Tree, Multilayer Perceptron, and (Linear) Support Vector Machine. All algorithms were implemented using Scikit-learn routines [26], with default configurations.

## V. RESULTS

To evaluate the performance of the classification methods, we computed their respective accuracy in terms of the classification of the vehicle type and its lane occupation. Due to the limited number of measurements, we evaluated the performance of the proposed methods using as testing set a single distinct measurement for each run; for that run, the training set was composed by the remaining 111 measurements.

For the deep-learning based method, the Tensorflow random seed, which defines the parameters initialization, was reset (to 42) before each run. For the Random Forest, Decision Tree, and Multilayer Perceptron machine learning methods, the variable *random_state* was reset to the same value.

The mean results for lane detection and vehicle classification are displayed in Table III and IV, respectively. As one may observe, despite the relatively low amount of training images, the proposed methodology achieved a maximum accuracy of 98% for lane occupation, using a KNeighbors classifier, and

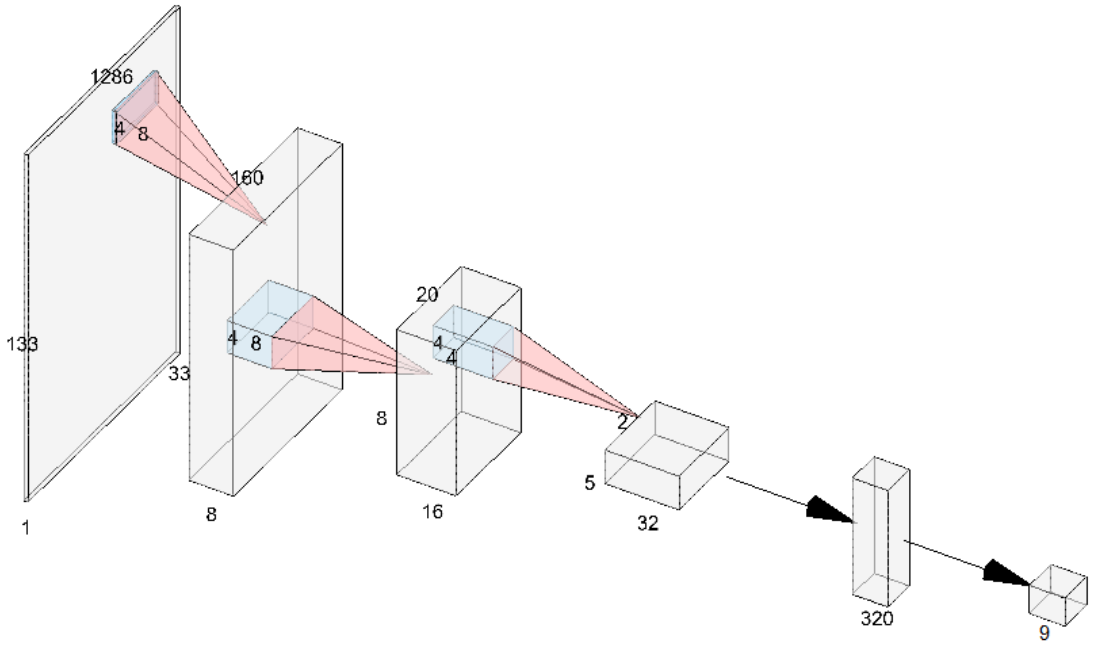


Fig. 7. Structure of the convolutional neural network deployed for image classification. Note that the arrays' widths are not to scale.

TABLE III
ACCURACY OF VARIOUS ALGORITHMS IN THE DETECTION OF THE LANE OCCUPIED BY A VEHICLE

| Algorithm | Accuracy |
|---|---|
| Deep-Learning | 0.955 |
| KNeighbors (n_neighbors=3) | 0.982 |
| Linear Discriminant Analysis | 0.982 |
| Linear SVM | 0.973 |
| Support Vector Machine (SVM) | 0.973 |
| Multilayer Perceptron | 0.964 |
| Gaussian Naïve Bayes | 0.964 |
| Gradient Boosting Classifier | 0.964 |
| Decision Tree | 0.946 |
| Random Forest | 0.946 |

TABLE IV
ACCURACY OF VARIOUS ALGORITHMS IN THE CLASSIFICATION OF THE TYPE OF VEHICLE

| Algorithm | Accuracy |
|---|---|
| Deep-Learning | 0.866 |
| Gaussian Naïve Bayes | 0.875 |
| Gradient Boosting Classifier | 0.875 |
| Random Forest | 0.866 |
| KNeighbors (n_neighbors=1) | 0.830 |
| Linear Discriminant Analysis | 0.830 |
| Multilayer Perceptron | 0.830 |
| Decision Tree | 0.812 |
| Support Vector Machine (SVM) | 0.786 |
| Linear SVM | 0.777 |

87% for vehicle type classification, using the Gaussian Naïve Bayes classifier. Both the KNeighbors and the Naïve Bayes algorithms can be interpreted as methods for approximating the optimal, maximum a posteriori classifier; see [27] for a detailed discussion.

TABLE V
TOTAL PROCESSING TIME (FOR THE 112 RUNS) OF THE DEPLOYED ALGORITHMS

| Algorithm | Processing Time (s) |
|---|---|
| Deep-Learning | 1658 |
| Multilayer Perceptron | 110 |
| Gradient Boosting Classifier | 58.8 |
| Random Forest | 3.45 |
| Support Vector Machine (SVM) | 0.45 |
| Linear SVM | 0.25 |
| Gaussian Naïve Bayes | 0.24 |
| Linear Discriminant Analysis | 0.19 |
| KNeighbors | 0.18 |
| Decision Tree | 0.14 |

The processing times required by the deployed algorithms is shown in Table V. The simulations were run on a Windows 10 computer with a i7 4770S CPU using Python 3.9.12 and Intel-optimized versions of Tensorflow (version 2.8.0) and Scikit-Learn (version 0.24.2). As one may observe, the CNN deep learning method has by far the largest computational requirement; however, most of it (99%) is demanded by the training step.

The resulting confusion tables for the best performing estimators are shown in Tables VI and VII, respectively. As one may observe, there were only 2 lane detection errors; for vehicle classification, in turn, there was a bias towards classifying vehicles as cars. We tried to circumvent this problem using techniques for dealing with unbalanced training sets (i.e., training sets in which classes have distinct frequencies), such as oversampling the least frequent classes [28], but the results did not change significantly.

TABLE VI
RESULTING CONFUSION TABLE FOR OCCUPIED LANE DETECTION USING THE THE KNEIGHBORS CLASSIFIER.

| | | True | | | |
|---|---|---|---|---|---|
| | | 1 | 2 | 3 | Total |
| Estimated | 1 | 33 | 0 | 0 | 33 |
| | 2 | 0 | 37 | 1 | 38 |
| | 3 | 0 | 1 | 40 | 41 |
| | Total | 33 | 38 | 41 | 112 |

TABLE VII
RESULTING CONFUSION TABLE FOR VEHICLE TYPE CLASSIFICATION USING THE THE GAUSSIAN NAÏVE BAYES CLASSIFIER.

| | | True | | | |
|---|---|---|---|---|---|
| | | car | moto | truck | Total |
| Estimated | car | 83 | 2 | 2 | 87 |
| | moto | 4 | 13 | 0 | 17 |
| | truck | 6 | 0 | 2 | 8 |
| | Total | 93 | 15 | 4 | 112 |

## VI. CONCLUSION

In this article, we tackled the problem of vehicle classification and lane occupation detection from UWB radar images. Given the looseness of the classification problem at hand, the lack of detailed information on the radar signals and noise statistics, and the paucity of radar measurements due to practical constraints, we resorted to machine learning classification methods. We developed two distinct classification approaches: one based on deep learning, employing a 5-layer convolutional neural network, and second one using off-the-shelf machine learning algorithms available in the Scikit-learn software package in tandem with a simple feature extraction method. Experimental results show that, using a set of 112 training images, the proposed methods achieved accuracies of 95% and 87% for lane occupation and vehicle type classification, respectively.

## ACKNOWLEDGMENT

This project was supported by São Paulo Research Foundation (FAPESP) under grants #2019/12268-4 (PIPE Program) and #2020/09838-0 (BI0S - Brazilian Institute of Data Science).

## REFERENCES

[1] J. Li, Q. Zhang and G. Zheng, "Research on Ultra-Wideband Radar Target Recognition Method", *Journal of Physics: Conference Series*, vol. 1651, Aug. 2020, doi:10.1088/1742-6596/1651/1/012194.

[2] M. Viot, A. Bizalion and J. Seegars., "Exploring Ultra-Wideband Technology for Micro-Location-Based Services", *Microwave Journal*, vol. 64, no. 10, Jun 2021. URL: https://www.microwavejournal.com/articles/36143-exploring-ultra-wideband-technology-for-micro-location-based-services

[3] K. Glover and B. Peterson,, "UWB: Enhancing Positioning, Safety and Security for Connected Vehicles", *Microwave Journal*, vol. 64, no. 9, Sep 2021. URL: https://www.microwavejournal.com/articles/36726-uwb-enhancing-positioning-safety-and-security-for-connected-vehicles

[4] J. D. Taylor, *Ultra-Wideband Radar Technology*. Boca Raton, FL, USA: CRC, 2001.

[5] L. Sakkila, A. ivenq, C. Tatkeu, Y. El Hillali, J. Ghys and J. Rouvaen, "Methods of target recognition for UWB radar", in *2010 IEEE Intelligent Vehicles Symposium*, 2010, pp. 949-954, doi: 10.1109/IVS.2010.5547962.

[6] J. W. Choi, X. Quan and S. H. Cho, "Bi-Directional Passing People Counting System Based on IR-UWB Radar Sensors", *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 512-522, April 2018, doi: 10.1109/JIOT.2017.2714181.

[7] G. Schouten and J. Steckel, "RadarSLAM: Biomimetic SLAM using ultra-wideband pulse-echo radar", in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2017, pp. 1-8, doi: 10.1109/IPIN.2017.8115932.

[8] A. M. J. Marindra and G. Y. Tian, "Multiresonance Chipless RFID Sensor Tag for Metal Defect Characterization Using Principal Component Analysis", *IEEE Sensors Journal*, vol. 19, no. 18, pp. 8037-8046, Sept 2019, doi: 10.1109/JSEN.2019.2917840.

[9] J. S. Park, I. S. Baek and S. H. Cho, "Localizations of multiple targets using multistatic UWB radar systems", in *3rd IEEE International Conference on Network Infrastructure and Digital Content*, 2012, pp. 586-590, doi: 10.1109/ICNIDC.2012.6418822.

[10] A. Oncu, B. B. M. W. Badalawa and M. Fujishima, "22–29 GHz Ultra-Wideband CMOS Pulse Generator for Short-Range Radar Applications", *IEEE Journal of Solid-State Circuits*, vol. 42, no. 7, pp. 1464-1471, July 2007, doi: 10.1109/JSSC.2007.899099.

[11] H. L. Van Trees,*"Optimum array processing: Part IV of detection, estimation, and modulation theory."*, John Wiley & Sons, 2004.

[12] I. G. Maglogiannis, *"Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in EHealth, HCI, Information Retrieval and Pervasive Technologies"*, ser. Frontiers in artificial intelligence and applications, Amsterdan, The Netherlands: IOS Press, 2007.

[13] Z. Wen, D. Liu, X. Liu, "Deep learning based smart radar vision system for object recognition", *J Ambient Intell Human Comput*, vol. 10, pp. 829–839, March 2019. https://doi.org/10.1007/s12652-018-0853-9.

[14] D. T. A. Nguyen, H.-G. Lee, E.-R. Jeong, H. L. Lee and J. Joung, "Deep Learning-Based Localization for UWB Systems", *Electronics*, vol. 9, no. 10, pp. 1712, Oct. 2020, https://doi.org/10.3390/electronics9101712.

[15] C. Jiang, J. Shen, S. Chen, Y. Chen, D. Liu, and Y. Bo, "UWB NLOS/LOS Classification Using Deep Learning Method," *IEEE Communications Letters*, vol. 24, no. 10, pp. 2226-2230, Oct. 2020, doi: 10.1109/LCOMM.2020.2999904.

[16] Z. Yang, F. Shi, and H. Liang, 'A Portable Traffic Counting, Speed Estimation, and Classification Terminal Using IR-UWB Radar," *IEEE Sensors Journal*, vol. 22, no. 13, Jul. 2022, doi: 10.1109/JSEN.2022.3181215.

[17] BeagleBone. [Online]. Available: https://beagleboard.org/, accessed Nov, 11, 2021.

[18] B. R. Mahafza, *Radar Systems Analysis and Design Using Matlab*, 3rd Edition. Boca Raton, FL, USA: CRC, 2013.

[19] C. Zhang, and A. E. Fathi, "Reconfigurable Pico-Pulse Generator for UWB Applications", in *2006 IEEE MTT-S International Microwave Symposium Digest*, 2006, pp. 407-410, doi: 10.1109/MWSYM.2006.249557.

[20] S. Oh, and D. D. Wentzoloff, "A Step Recovery Diode Based UWB Transmitter for Low-Cost Impulse Generation", in *2011 IEEE International Conference on Ultra-Wideband (ICUWB)*, 2011, pp. 63-67, doi: 10.1109/ICUWB.2011.6058923.

[21] D. Lee, G. Shaker, and W. Melek, "Development of a compact monocycle pulse generator for UWB impulse radar applications," *Microwave and Optical Techonology Letters,* vol. 62, no. 10, pp. 3119-3123, Oct. 2020, doi: https://doi.org/10.1002/mop.32424.

[22] J. Yang, G. Pyo, C. -Y. Kim and S. Hong, "A 24-GHz CMOS UWB Radar Transmitter With Compressed Pulses," *IEEE Transactions on Microwave Theory and Techniques,* vol. 60, no. 4, pp. 1117-1125, Apr. 2012, doi: 10.1109/TMTT.2012.2184136.

[23] M. Parlak, M. Matsuo, and J. F. Buckwalter, "Analog Signal Processing for Pulse Compression Radar in 90-nm CMOS," *IEEE Transactions on Microwave Theory and Techniques,* vol. 60, no. 12, pp. 3810-3822, Dec. 2012, doi: 10.1109/TMTT.2012.2222433.

[24] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard. Tensorflow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pp. 265–283, 2016. URL: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

[25] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks", *Proc. Int. Conf. Artificial Intell. Statist.*, vol. 9, pp. 249-256, May 2010. URL: https://proceedings.mlr.press/v9/glorot10a.html

[26] O. Kramer. "Scikit-learn". *Machine learning for evolution strategies*, Cham, Switzerland: Springer, pp. 45–53, 2016.

[27] S. Theodoridis, *Machine learning: a Bayesian and optimization perspective*. London, U.K.: Academic press, 2015.
[28] A. Viloria, O. B. P. Lezama, and N. Mercado-Caruzo,"Unbalanced data processing using oversampling: Machine learning." *Procedia Computer Science,* vol. 175, pp. 108–113, 2020. doi: 10.1016/j.procs.2020.07.018.

**Marcelo B. Perotoni** Elec. Eng. (1995, UFRGS, Porto Alegre, Brazil), MsC (2001) and PhD (2005) in Electrical Engineering from USP (Sao Paulo, Brazil). Interests in RF and EMC. He is currently a professor at UFABC. Orcid ID: 0000-0002-7398-5766

**Claudio J. Bordin Jr.** received the B.S., M.S., and Ph.D. degrees, all in Electrical Engineering, from Escola Politécnica, Universidade de São Paulo, Brazil, in 2000, 2002, and 2006, respectively. He is currently an Associate Professor at Universidade Federal do ABC, Brazil. His research interests lie in the area of Signal Processing and Applied Statistics. Orcid ID: 0000-0002-7016-5922

**Fernando de Andrade Castilho** Elec. Eng. (2003, USP, Sao Paulo, Brazil). Interests in RF and embedded systems. Currently working at Flexmedia (design/development).

**Gustavo Yamasaki Martins Vieira** Gustavo received the B.S. and M.S. degrees, all in Electrical Engineering, from Escola Politecnica, Universidade de Sao Paulo, Brazil, in 2003 and 2007 respectively. He is currently an Engenieer at Flexmedia and works with embedded systems for ITS (Intelligent Transportation System). Orcid ID: 0000-0002-0622-7957