# Vehicle Classification Using Raspberry Pi: A Guide to Capturing WiFi CSI Data

Tharidhu Schmidt
*Department of Computer Science and Engineering*
*University of Moratuwa*
Moratuwa, Sri Lanka
tharidhu.22@cse.mrt.ac.lk

Thanuja Ambegoda
*Department of Computer Science and Engineering*
*University of Moratuwa*
Moratuwa, Sri Lanka
thanuja@cse.mrt.ac.lk

Kutila Gunasekera
*Department of Computer Science and Engineering*
*University of Moratuwa*
Moratuwa, Sri Lanka
kutila@cse.mrt.ac.lk

2023 Moratuwa Engineering Research Conference (MERCon) | 979-8-3503-4521-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/MERCon60487.2023.10355444

***Abstract*—Traffic monitoring Systems are an essential data collection tool in traffic analysis and transport planning. In this paper, we aim to identify whether the popular low-cost single board computer Raspberry Pi 3B+ can be used as an alternative to existing solutions which require bulky and expensive setups to effectively capture WiFi CSI data for vehicle classification. We also look into the problems faced in this implementation and their solutions. We propose a data processing pipeline for this approach to aid in creating an annotated dataset for the classification of vehicle types. The results show that the proposed system can successfully capture WiFi CSI data for vehicle classification.**

***Index Terms*—Channel state information (CSI), Raspberry Pi, Nexmon, Vehicle Classification**

## I. Introduction

Traffic management is increasingly an essential part of maintaining the health of current road infrastructure in an ever-increasing urban landscape. Traffic management is traditionally performed by relying on data collected by manned surveys. Due to high traffic volumes in urban settings, automated data collection is preferred, as manned surveys are costly in these scenarios. Automated traffic data collection tools affixed to road infrastructure are known as Traffic Monitoring Systems (TMS). The need to keep up with demand and be a cost-effective, robust, and privacy-preserving way of gathering traffic data is needed. Previous research shows that radio-based vehicle identification systems fit these criteria [1], [2]. These proposed methods are costly and need to be prepared for mass deployment since they involve bulky laptops and antennas to be placed beside roads.

In this paper, we propose a novel solution to the problem. We use the unique WiFi interference patterns captured on low-cost devices like the Raspberry Pi from vehicles passing through a road segment. This proposed approach contributes to a hardware and software solution to collect WiFi Channel State Information (CSI) data and process these into an annotated dataset. The following is a breakdown of the key contributions of this research;

- We design and implement roadside units based on Raspberry Pi 3B+ deployed to effectively capture WiFi interference patterns generated by vehicles passing through and related ground truth video.
- We discuss the challenges faced in implementing the proposed system, the troubleshooting methods, and the solutions required to solve them.
- We develop a semi-automated method for processing of video ground truth data to create annotations.
- We also develop an automated process to generate an annotated dataset of CSI data using the annotations of ground truth and packet capture file.

979-8-3503-4521-6/23/$31.00©2023 IEEE

## II. Related Work

Through earlier work, researchers have compared various methods used for gathering classification data directly from the road [1]– [3]. These methods can be divided into five general categories: In-Road based solutions, Acoustics, Inertial, Vision and Radio Frequency. In-road based solutions are the classic approach for monitoring traffic, which typically involves devices such as piezoelectric sensor [5] or loop detector [6] being dug into a road surface, which makes this type of system costly in deployment and maintenance, which limits its deployment to select locations. The acoustics-based approach uses a microphone array placed roadside to collect acoustic data, where extracted features are used to perform traffic monitoring [7]. Inertial based approach uses Inertial Measurement Unit (IMU) placed beside the road to capture vibrations caused by vehicles moving. This is then combined with a magnetometer further to increase the accuracy of this approach [8]. Vision-based approaches focus on CCTV cameras installed near the roadside for vehicle classification. The criteria to compare these categories are: Privacy preservation, Cost efficiency, non-intrusiveness and weather independence. Comparisons were made to show that the radio-based approach is the most suitable for these properties. Radio-based approaches [1]– [4], [9] are non-intrusive since they do not require induction loops or similar hardware dug into the road surface, which also results in cost efficiency and can easily be set up beside the road using existing infrastructure. This approach also avoids privacy-related issues that arise from vision-based systems. Performance is also not heavily constrained by weather.

There are several ways of gathering classification data from radio-based approaches. These include using the ZigBee standard and WiFi standard. The primary data parameter collected in the ZigBee approach is the Received Signal Strength Indicator (RSSI) [2], while for WiFi, it is Channel State Information (CSI) [3], [9]. Since WiFi allows for CSI, it is preferred over RSSI due to CSI containing rich contextual information of the transmission link and providing more stability [3]. These properties of CSI result in higher accuracies for multi-class classification [3] over RSSI [2], which are unreliable for similar body sizes from different vehicle classes.

Current implementations of WiFi CSI data for vehicle classification involve the use of Intel NIC 5300, which involves costly and bulky setups due to the need for laptop computers and antennas [3], [9]. Recently researchers have implemented an alternate method for capturing WiFi CSI data, using Raspberry Pi, primarily used for Human Activity Classification [10], [11], [13]. Use of Raspberry Pi is more cost-effective and allows a compact setup compared to the Intel Network Interface Cards (NICs). Therefore our research looks into the prospects of using Raspberry Pi for gathering CSI data for vehicle classification.

## III. Methodology

This research aims to identify whether the Raspberry Pi 3B+ can effectively capture WiFi CSI interference patterns of vehicles passing through a two-lane road, as demonstrated with the Intel 5300 NICs. A system was proposed and implemented to achieve this goal, which is further detailed in Section IV. A data processing pipeline was created to handle the data generated by this particular system implementation. In Section V, the capability of this system and its associated data processing pipeline is analysed in generating an annotated dataset.

### A. Data Collection

The particulars of a system implementation for data collection can profoundly impact the quality of the CSI data captured. The configuration implemented aligns with the expectations of data capture for traffic monitoring and is similar to other Raspberry Pi related CSI capturing setups. Fig. 1 details how each device implemented in the system performs to achieve this goal. The task assigned to the transmitter is to send *ping* packets to the Receiver. The receiver replies to each *ping* with a *pong*. This *pong* response contains the CSI information explaining the physical and Radio Frequency (RF) interferences on the link between the transmitter and receiver as observed during the *ping*. A third Raspberry Pi - deployed as a data capture device - captures these *pong* packets and also contains a wide-angle Raspberry Pi camera to capture the ground truth video footage. The collected *pong* packets and video footage are stored on the SD card of this device.

### B. Data Processing

The data processing pipeline consists of two main components semi-automated tagging of ground truth to obtain annotations and extracting CSI from pong packets using annotations to create a dataset. The former can further be broken down into two components: Automated tagging using YOLOv3 and manual review.

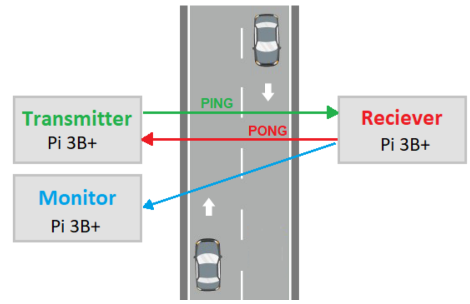


Fig. 1. WiFi packets transmission at each device

*1) Automated Tagging:* Vehicle classification and lane detection software is designed to annotate ground truth video data. The YOLOv3[1] model, trained on the Microsoft Common Objects in Context (MS COCO) dataset, contains 4 vehicle classes; Motorbike, Car, Truck, and Bus. Since we are interested only in vehicle classification, the labels containing the vehicle classification are chosen.

The first step is pre-processing the video frames into a format suitable for vehicle classification using the YOLOv3 model. YOLO sometimes gives multiple bounding boxes for a single vehicle. Non-max suppression is used to reduce the number of detection boxes and have to take the best detection box for each class.

The next step is implementing a tracking algorithm with defined boundary lines for lane detection and obtaining the timestamp when the vehicle passes the detection area. The tracking algorithm uses the Euclidean distance concept to keep track of the vehicle as it moves between the video frames. It works by calculating the difference between two centre points of an object in the current frame vs the previous frame. If the distance is less than the threshold, then it confirms that the vehicle is the same as the previous frame. Boundary lines are defined as follows; a centre line is placed on the video frames at the position perpendicular to the link between the receiver and transmitter. Apart from the centre line, two other boundary lines are placed to the left and right. These boundary lines can be seen in the annotated frame shown in Fig. 2.

Finally, lane detection and annotation are performed. When the centre point of the vehicle's bounding box moves from the right boundary line to the left, cutting through the centre line and vice versa, the *timestamp in the form of frame number*, *vehicle class* and *lane* are extracted and recorded to a *csv* (comma-seperated values) file as annotations.

*2) Manual Review:* This process reviews the annotations generated by the automated tagging process for errors, such as wrongful classifications and multiple detections caused by large vehicles, which can affect the dataset's quality. It also adds the 2 classes, three-wheelers and vans, unavailable in

[1]Since YOLOv3 provides satisfactory levels of accuracy and speed, there was no need to upgrade to more recent versions of YOLO.

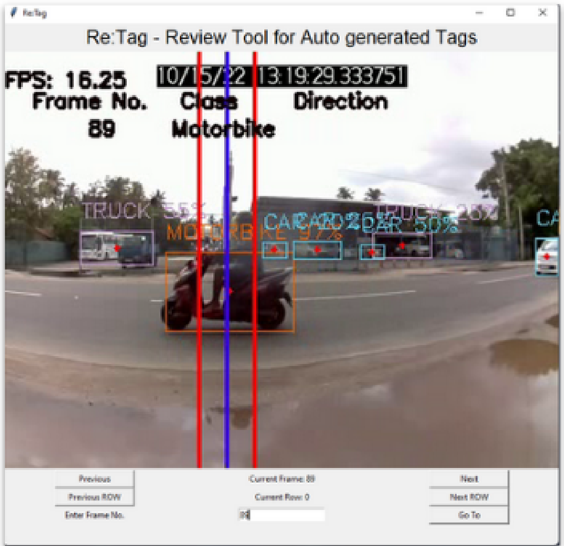


Fig. 2. Software created for Manual Review

YOLOv3. Custom software was written for this particular purpose, as shown in Fig. 2. Annotations and labelled ground truth video generated by the above process are input into the software, which allows the review of annotations by scrolling through the records in the *csv* file and showing its associated video frame. This process refines the annotations and makes them suitable for the next step, extracting CSI and producing annotated dataset.

*3) Dataset Preparation:* Refined annotations file from the above process is then used to extract each vehicle's CSI (amplitude and phase) records and create an annotated dataset. Annotations *csv* file and packet capture (pcap) file are input to a Python script designed for this research to extract CSI records on each vehicle from the pcap file to create an annotated dataset. The first step in this process is converting the frame number in annotations to a timestamp;

$$ot + (fn/k) \tag{1}$$

In Eq. (1) ot - timestamp of the beginning of the ground truth video, fn - frame number at point of vehicle passing center line, k - frame rate of the ground truth video, in this case, 30 fps (frames per second).

This timestamp is then converted to a time window determined experimentally for each vehicle type. Window size is determined experimentally by finding the average number of frames taken for each vehicle class, assuming the speed is constant. This time window is used as a time filter in *tshark* command line tool to extract the packets contained within the main *pcap* file into the *pcap* file for each vehicle. Then *CSIkit* is used on the output *pcap* file to extract CSI records of amplitude and phase, creating an annotated dataset for each vehicle.

## IV. Experimental Setup

The environment chosen for system implementation was a two-lane road 7 meters in width, and the transmitter and monitor were placed 1.5m from the edge of the road and receiver on the other side at the exact opposite position on the roadside, as seen in Fig. 3.

- Transmitter - Raspberry Pi 3B+ was used as a transmitter to create a packet rate between itself and receiver using the *ping* command or *iperf3*. As seen in Table I various configurations were tested. These include using TCP (Transmission Control Protocol) or UDP (User Datagram Protocol) protocols in *Iperf3*.
- Receiver - Raspberry Pi 3B+ was configured as an Access Point (AP) with OpenWrt OS version 22.03.0 operated in 802.11ac with 5 GHz Frequency, Channel 157 and 80 MHz bandwidth.
- Monitor - Raspberry Pi 3B+ with a modified WiFi driver developed by SEEMON LABS [12] called *Nexmon CSI* was used to capture packets containing raw CSI data. Configured Debian version 11 (Bullseye/ Kernal 5.4) with Nexmon CSI. The input filter for *Nexmon CSI* was configured with the following options; Channel 157/80, Core 1, NSS mask 1, and MAC address filter for receiver.

In order to collect ground truth data, a Raspberry Pi camera (with a wide-angle lens) was connected to the Monitor, to capture passing vehicles in sync with packets containing CSI data. Another camera was used on the opposing side to identify any overlapping vehicles not captured by the Raspberry Pi camera.

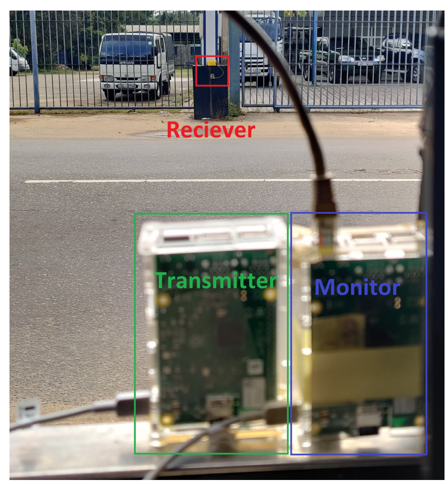


Fig. 3. Experimental System Implementation

TABLE I
PILOT SURVEY CONFIGURATIONS

| Survey No. | Channel/Bandwidth | Traffic Generator |
|---|---|---|
| Survey 1 | 157/80 | TCP - No limit (Iperf3) |
| Survey 2 | 157/80 | UDP - 300Hz (Iperf3) |
| Survey 3 | 157/20 | PING - 1000Hz |

## V. Results and Discussion

This section evaluates the proposed system, including the hardware implementation and the accompanying data processing pipeline. Two metrics were used to evaluate the proposed system for its capability to capture CSI data for vehicle classification, and these are: Packet capture Rate and CSI visualization of Amplitude. Packet capture Rate distribution was plotted and analyzed to see whether the capture has a consistent rate. Heatmap visualization is performed on a sample of CSI data when no vehicles are passing through to analyze the CSI as a control sample.

### A. Issues and Solutions

Pilot Survey 1 was conducted using *Iperf3*'s TCP with no bandwidth limit for WiFi traffic generation. Packet capture was performed with the proposed system in the environment shown in Fig. 3 with vehicles passing through. Results show an inconsistent packet capture rate, as seen in Fig. 4. The heatmap shows that bandwidth saturation is intermittent (Fig. 10), which can cause problems in capturing an interference pattern on WiFi CSI data for vehicles. It was assumed these issues arose due to problems in network traffic generation from the *Iperf3* and not from outside interference on the 5GHz - 157 Channel, as this was checked to be the only operating band in the vicinity.

Then Pilot Survey 2 was conducted using *Iperf3*'s UDP, and a packet rate of 300 per second was set. Analysis of this survey also showed that the packet capture was inconsistent (Fig. 5), and the bandwidth saturation was intermittent.

Pilot Survey 3 switched to using *ping* over *Iperf3*, and a 2000Hz *ping* rate was set. Due to the intermittent bandwidth saturation issue of the previous two surveys, the bandwidth of receiver and capture parameters of *Nexmon CSI* was set to 20 MHz. Results showed improvements in packet capture rate (Fig. 6) over previous surveys. Setting the bandwidth to 20 Mhz resolved the intermittent bandwidth saturation issue.

As the packet capture rates are still inconsistent, it was determined that this might be due to an issue in *Nexmon CSI* in Kernel 5.4. Therefore, another version, Kernel 5.10.92-v7, was used. *Nexmon CSI* and WiFi configurations are the same as before; 3 experiments were conducted to analyze the performance of the packet capture rate. The experiments were conducted for *ping* intervals of 0.01 seconds, 0.005 seconds and 0.002 seconds. Results from this can be seen in Fig. 7 - 9 and Table II. Fig. 10 shows the improvement of bandwidth saturation consitency in CSI samples captured using *Nexmon CSI* for Kernel 5.10.92-v7 over Kernel 5.4. We have finally achieved an almost consistent sampling rate from these newly obtained results, and these results can be linearly interpolated to the related sample rate.

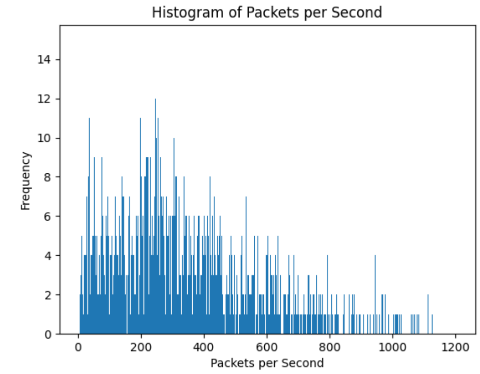


Fig. 4. Pilot Survey 1 - Packet Capture Rate Distribution

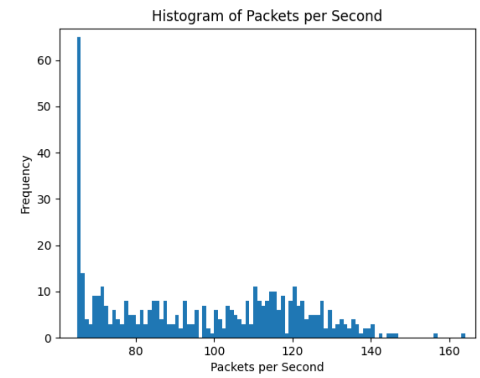


Fig. 5. Pilot Survey 2 - Packet Capture Rate Distribution

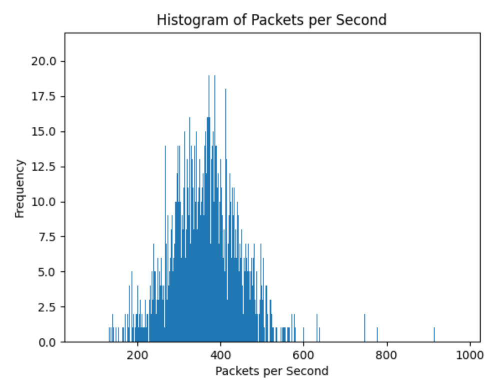


Fig. 6. Pilot Survey 3 - Packet Capture Rate Distribution

TABLE II
Ping Configurations and Capture Rate

| Ping Interval | Packet Capture Rate |
|---|---|
| 0.01 sec (100 Hz) | 97.82 Hz |
| 0.005 sec (200 Hz) | 192.21 Hz |
| 0.002 sec (500 Hz) | 461.85 Hz |

### B. Selection of Window Sizes

It was observed that when a vehicle crosses through the line of sight path between the transmitter and receiver, CSI values change significantly. Therefore a window size needs to be determined based on vehicle classes to effectively extract the CSI values from the packet capture file of the survey. By analysing the ground truth footage of each vehicle class and getting the average time to pass the detection area, the

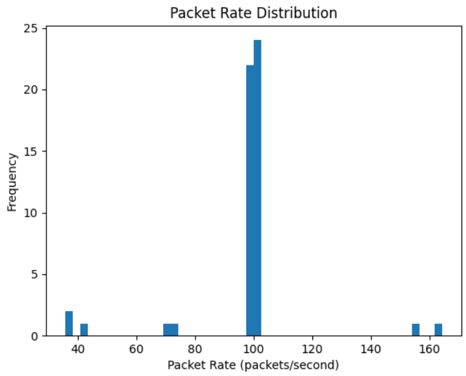


Fig. 7. Ping Interval (0.01) - Packet Capture Rate Distribution

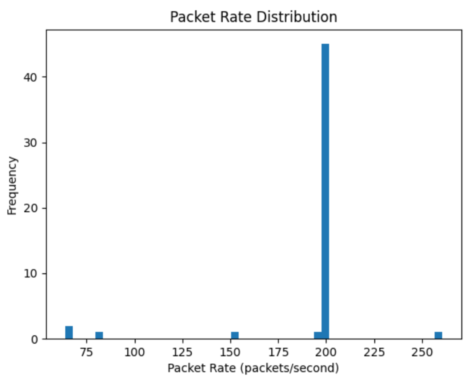


Fig. 8. Ping Interval (0.005) - Packet Capture Rate Distribution

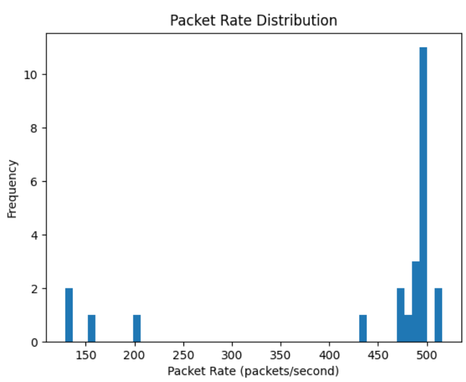


Fig. 9. Ping Interval (0.002) - Packet Capture Rate Distribution

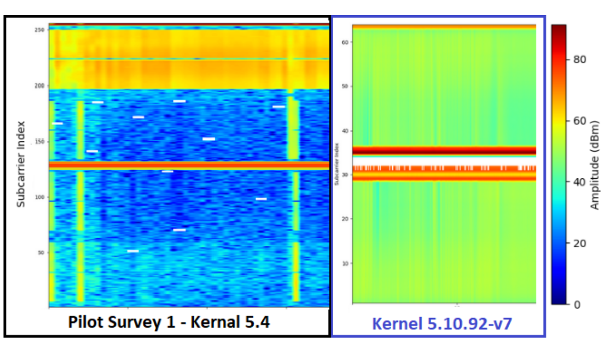


Fig. 10. Bandwidth Saturation

following window sizes were obtained; Motorbike - 0.10 seconds, Threewheeler - 0.17 seconds, Car and Van - 0.26 seconds, Truck and Bus - 0.40 seconds. These window sizes only apply to the location shown in Fig. 3 at the speeds observed. They need to be adjusted for other locations.

### C. Comparison to Existing Techniques

Previous research for capturing CSI data for vehicle classification has used Intel NIC 5300. Intel NIC 5300 is a WiFi card with 3 antennas installed in a PC or Laptop [3] [9]. This makes their setup bulky, having to place three antennas, and unsuitable for quick deployment. Our proposed system uses a raspberry pi with no additional hardware making it cheap and easy to deploy.

Packet capture rate frequencies by the Intel NIC 5300 are relatively high, 1000 Hz [9] and 2500 Hz [3] respectively. In comparison, Raspberry Pi in Human Activity Recognition scenarios, 50 Hz [13] and 100 Hz [11]. Our system achieved 500 Hz in the high end, significantly higher than previous achievements using a Raspberry Pi. We consider the current capture rate sufficient for vehicle classification, and there may be potential for enhancing the packet capture rate further.

## VI. Conclusion

The data analysis of the proposed system implementation shows that Raspberry Pi can indeed be used to capture WiFi CSI data from moving vehicles effectively. To the best of our knowledge, this is the first attempt to classify moving vehicles using WiFi CSI data captured with low-cost Raspberry-Pi devices. Furthermore, our proposed data processing pipeline can handle the captured data to create an annotated dataset. This also shows that the proposed system is a viable way of capturing CSI data for the application of vehicle classification.

The next step of our research is to use the collected CSI dataset to train and test a machine learning model for vehicle classification. Our future work will also address the challenges of incorporating vehicles operating at varying speeds, and within fluctuating traffic densities. By doing so, we aim to construct a robust dataset capable of training a machine learning model to adeptly handle the myriad situations encountered on crowded urban roadways, where such systems are in most demand. To tackle the complexities associated with classifying vehicles moving in parallel, we recognize the necessity of employing a robust annotation methodology. This approach will enable the precise capture of this behaviour, ultimately facilitating the development of a highly effective machine learning model.

## Acknowledgment

This research was supported by the Accelerating Higher Education Expansion and Development (AHEAD) Operation of the Ministry of Higher Education, Sri Lanka funded by the World Bank.

## REFERENCES


[1] B. Sliwa, M. Haferkamp, M. Al-Askary, D. Dorn and C. Wietfeld, "A radio-fingerprinting-based vehicle classification system for intelligent traffic control in smart cities," 2018 Annual IEEE International Systems Conference (SysCon), Vancouver, BC, Canada, 2018, pp. 1-5, doi: 10.1109/SYSCON.2018.8369511.

[2] B. Sliwa, N. Piatkowski and C. Wietfeld, "The Channel as a Traffic Sensor: Vehicle Detection and Classification Based on Radio Fingerprinting," in IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7392-7406, Aug. 2020, doi: 10.1109/JIOT.2020.2983207.

[3] M. Won, S. Sahu and K. Park, "DeepWiTraffic: Low Cost WiFi-Based Traffic Monitoring System Using Deep Learning," in 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Monterey, CA, USA, 2019, pp. 476-484, doi: 10.1109/MASS.2019.00062

[4] A. Wijesinghe, T. D. Ambegoda, A. S. Perera and T. Sivakumar, "Siamese Networks for RF-Based Vehicle Trajectory Prediction," 2023 IEEE 17th International Conference on Industrial and Information Systems (ICIIS), Peradeniya, Sri Lanka, 2023, pp. 495-500, doi: 10.1109/ICIIS58898.2023.10253541.

[5] Z. Ye, H. Xiong, and L. Wang, "Collecting comprehensive traffic information using pavement vibration monitoring data," Computer-Aided Civil and Infrastructure Engineering, pp. 1–16, May 2019.

[6] J. J. Lamas-Seco, P. M. Castro, A. Dapena, and F. J. Vazquez-Araujo, "Vehicle classification using the discrete fourier transform with traffic inductive sensors," Sensors, vol. 15, no. 10, pp. 27 201–27 214, 2015.

[7] J. George, L. Mary, and Riyas K S, "Vehicle detection and classification from acoustic signal using ANN and KNN," in 2013 International Conference on Control Communication and Computing (ICCC), Dec 2013, pp. 436–439.

[8] W. Ma, D. Xing, A. McKee, R. Bajwa, C. Flores, B. Fuller, and P. Varaiya, "A wireless accelerometer-based automatic vehicle classification prototype system," IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 1, pp. 104–111, Feb 2014.

[9] M. Won, S. Zhang and S. H. Son, "WiTraffic: Low-Cost and Non-Intrusive Traffic Monitoring System Using WiFi," 2017 26th International Conference on Computer Communication and Networks (ICCCN), Vancouver, BC, Canada, 2017, pp. 1-9, doi: 10.1109/ICCCN.2017.8038380.

[10] J. Schäfer, B. R. Barrsiwal, M. Kokhkharova, H. Adil, and J. Liebehenschel, "Human Activity Recognition Using CSI Information with Nexmon," Applied Sciences, vol. 11, no. 19, p. 8860, Sep. 2021, doi: 10.3390/app11198860.

[11] G. Forbes, S. Massie and S. Craw, "WiFi-based Human Activity Recognition using Raspberry Pi," 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), Baltimore, MD, USA, 2020, pp. 722-730, doi: 10.1109/ICTAI50040.2020.00115.

[12] M. Schulz, D. Wegemer, and M. Hollick, "Nexmon: The c-based firmware patching framework," 2017. [Online]. Available: https://nexmon.org

[13] H. Lowe, M. Lamahewage and K. Gunasekera, "Towards a Low-cost WiFi based Real-time Human Activity Recognition System," 2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS), Barcelona, Spain, 2022, pp. 1-6, doi: 10.1109/COINS54846.2022.9854935.