Siamese networks for RF-based vehicle trajectory prediction

1st Anjana Wijesinghe
Dept of Computer Science & Engineering
University of Moratuwa
Moratuwa, Sri Lanka
anjana.21@cse.mrt.ac.lk

2nd Thanuja D. Ambegoda
Dept of Computer Science & Engineering
University of Moratuwa
Moratuwa, Sri Lanka
thanujaa@uom.lk

3rd Amal Shehan Perera
Dept of Computer Science & Engineering
University of Moratuwa
Moratuwa, Sri Lanka
shehan@cse.mrt.ac.lk

4th Thillaiampalam Sivakumar
Dept of Transport & Logistics Management
University of Moratuwa
Moratuwa, Sri Lanka
tsivakumar@uom.lk

2023 IEEE 17th International Conference on Industrial and Information Systems (ICIIS) | 979-8-3503-2363-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICIIS58898.2023.10253541

***Abstract*—Traffic surveys monitor the traffic flow to generate data used in improving traffic management. These surveys were traditionally carried out solely by manpower. With the improvement of video analysis, traffic surveys have been shifting towards using automated processes. Surveys carried out by manpower are ineffective unless the surveyors are professionally trained while using video footage requires expensive systems, large digital storage space, and bandwidth for IoT applications. Smart city concepts rely heavily on real-time data from many systems to operate and would benefit from lower computational power, power to operate, and bandwidth requirements. This research investigates the use of radio frequencies (RF) to capture vehicles moving through an intersection. RF data has an inherent nature of being more privacy-preserving than video and requiring less digital storage space and bandwidth while having all the benefits of using video for this purpose. Multiple machine learning models were experimented with for vehicle detection where long short term memory neural networks achieved the best result and was used to detect the presence of vehicles. The vehicle trajectory prediction algorithm used the data of the detected vehicles to predict the trajectory using the similarity between records where siamese neural networks with triplet training outperformed other methods. The data generated can be used to compute metrics for the vehicle occupancy of an intersection. This research enables traffic surveys and real-time monitoring to be carried out with minimal manpower using low-cost low-powered devices that generate smaller sized data samples.**

***Index Terms*—Vehicle tracking, Vehicle detection, Machine learning, Neural networks, Radio frequencies**

I. Introduction

Traffic surveys are regularly conducted to generate the data required to manage the traffic flow. This is a key ingredient in sustainable transportation management [1]. These surveys are conducted at intersections as optimizing traffic flow at intersections would result in all roads leading to it being optimized as well. In addition, smart city concepts managing traffic rely on real-time data to optimize automated processes [2] and require feeds from multiple locations throughout the city for decision-making.

Traditionally, traffic surveys were conducted using manpower. Unless the surveyors are professionally trained individuals, the results produced are highly inaccurate [3] while intersection intricacy directly correlates to the amount of manpower required. Enhances to computer vision have resulted in it being adopted for traffic surveying tasks [4]. These setups require cameras with local storage solutions or cloud storage and an internet connection. The cameras should be mounted with a viewing angle that covers the required portion of the intersection without any obstructions (in the case of an obstruction, multiple cameras would be required to cover that area). Higher-resolution videos are preferred as they retain sharper images with more information but consume more storage space. Digital space required increases drastically with the video resolution, recording length, and number of cameras used. Further, bad weather (i.e. heavy rain, snowing) and time of day (i.e. night) adversely affect the data gathered by cameras. Cameras capture all the information within their viewing angle, resulting in more information captured than is required for traffic surveys which could contain information that could be considered a privacy violation [5], [6]. Researchers have come up with innovative ways to mitigate the information that could violate the privacy of an individual and some work exceptionally well [7] but costs compute power to perform. Although, it could be argued that avoiding capturing more than required wherever possible would be always preferable.

While vehicle trajectory prediction using video data has been extensively researched with success [8], research using radio frequency (RF) data is still in its early stages. This research focuses on using RF data to conduct traffic surveys using low-powered low-cost equipment that generates smaller-sized but valuable data samples to carry out traffic surveys at intersections.

As vehicles are composed of metallic substances, moving vehicles tend to disrupt the electromagnetic field in the immediate area. The disruption caused to the electromagnetic

This research work was supported by the Accelerating Higher education Expansion and Development (AHEAD) Operation of the Ministry of Higher Education funded by the World Bank (grant number - Credit/Grant #:6026-LK/8743-LK).

979-8-3503-2363-4/23/$31.00 ©2023 IEEE

field largely depends on the vehicle and its composition while other factors such as vehicle's speed and load will have minor contributions [9], [10]. These disruptions can be measured by placing an electromagnetic connection across the path of a moving vehicle as illustrated by Fig. 1. The transmitter continuously transmits dummy data while the receiver continuously records the received signal strength indication (RSSI) of the transmission. The signal attenuation patterns are derived from the RSSIs recorded and can then be used to detect vehicle presence, classify vehicles, identify the driving direction, and determine the speed of the vehicle.

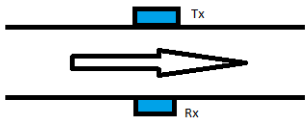

Fig. 1: Single lane device layout.

This method only captures the RSSI. Compared to using a camera system, this would require far less power and digital storage space resulting in less bandwidth and compute power used in real-time scenarios. The equipment used is cheaper allowing for more stations to be placed for the same budget in smart cities. RF signals at short ranges are not affected by the weather and time of day. Further, as this does not capture any information that could be considered a privacy violation, anonymization techniques are not required to be applied saving on compute power. Therefore, this overcomes multiple drawbacks of using video footage for traffic surveys.

This research approaches the problem of vehicle tracking with RSSI by taking a modular approach to vehicle detection and path tracking thereby allowing either algorithm to be used in isolation or replaced with ease. This research investigates the use of RF to detect and track vehicles without the use of vehicle classification or unique vehicle identification.

Our contributions are,

1) The use of a Long-Short Term Memory (LSTM) neural network (NN) in detecting the presence of a vehicle using electromagnetic data.
2) The use of Siamese NN in vehicle path tracking using pattern recognition on electromagnetic data.
3) Generating the parameters required to calculate vehicle occupancy metrics for intersections using only RF data.

This paper is structured with section II exploring current research in related areas, section III describing the data collection and analysis, section IV detailing the approach taken, and section V delivering the conclusions made.

II. Background

In the area of detecting and classifying vehicles based on sensory inputs, two methods have primarily been researched: only using RF signals and a combination of sensors (i.e. magnetometers, acoustic, and accelerometers). In comparison, research into RF only to detect vehicles has more contributions.

A. Vehicle detection

In identifying vehicles using RF signals, the common approach is to place a RF transmitter and receiver on either side of the road [9], as shown in Fig. 1 and the signal attenuation patterns are obtained.

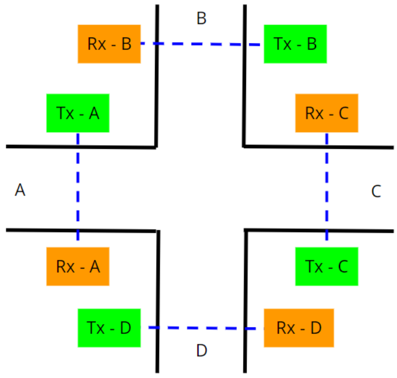

Fig. 2: Intersection device layout.

Fig. 3: RF based system device layout (Tx – transmitter, Rx – receiver).

Various types of machine learning models have been used to classify the signal attenuation patterns to determine the presence of a vehicle. NNs were used for binary classification by [11] resulting in 100% accuracy and 98% for multi-class classification (3 classes). SVMs were also able to achieve an accuracy of 100% on a 3-class classification [12]. These were followed by Decision trees with 98.4%, Naive Bayes with 94.9%, and kNN with 91.9% [13].

Multiple researchers focused on classifying vehicles with the best accuracy obtained by using an SVM to classify 2 classes where 99% accuracy was achieved with WiFi data [10]. J. Lan, et al. [14] was able to achieve 93% using SVM on a similar problem.

By increasing the number of transmitter and receiver pairs, researchers were able to increase the accuracies of vehicle detection, vehicle classification, and speed estimation. [10], [12], [15].

B. Uniquely identifying vehicles

R. A. Kerekes, et al. [16] and X. Dong, et al. [17] extend the research in detecting and classifying vehicles to uniquely recognize vehicles. Depending on the phenomenon that vehicles with internal combustion engines transmit electromagnetic emissions that are distinctive to the vehicle when running; characteristics of vehicles can be derived from their emissions to uniquely discriminate them.

X. Dong, et al. [17] implements a system with a single RF receiver created by coupling a biconical antenna with a sampling oscilloscope to collect data. The records are fed into a NN to uniquely identify the vehicle [17]. Preprocessing included Short term Fourier transform (STFT), parameter extraction and principal component analysis. With a sample size of 1,110 shared across 3 different vehicles, the research manages to obtain 99.3% accuracy in identifying the vehicles uniquely.

Further improving on this approach, an amplified RF sensor in combination with a magnetometer and microphone are used to capture data that is then fed into a Gaussian kernel regression algorithm with the nearest neighbor approach to identify vehicles uniquely [16]. With a sample size of 112 distributed between 7 target vehicles, an overall accuracy of 94% on known vehicles and 88% on all vehicles were achieved.

III. Data collection and analysis

Data was collected for 2 hours at a 4-way intersection. An RF connection was placed across the paths for all the entrances and exits to the intersection as illustrated by Fig. 2 at a height of 2 feet from the ground. Two transmitters with each paired with two receivers were used to capture data. Transmitters were generic IEEE 802.11 standard models with 2.4GHz transmission band. Raspberry Pi model 3 devices were set up as receivers. The Pis were set to reading and storing approximately 70 readings per second with timestamps. The transmitter is set to continuously transmit dummy data while the receiver records the RSSI. To ensure that no other active transmitters were present, the location was screened for them before and during data collection.

Video footage of the intersection was captured to be used as the ground truth. The ground truth contained the entry and exit timestamps, entry and exit lanes, and vehicle class (determined by the size of the vehicle).

A. Data analysis

A flow rate of approximately 150 vehicles per hour was observed at the intersection. Table I presents the findings of the analysis. A majority of vehicles passed through the intersection within a second with the slowest consuming 176s and fastest less than 1s (the maximum resolution of the data was 1s resulting in vehicles taking less than a second being documented as 0s) An average of 3.63 vehicles occupied the intersection at a given time with a mode of 3 and a maximum of 7 vehicles.

TABLE I: Data analysis.

	Mean	**Mode**	**Median**	**Min**	**Max**
Time in intersection	26.54s	1s	7s	0s	176s
No. of vehicles	3.63	3	4	0	7

IV. Methodology

The system diagram is illustrated in Fig. 4. Data collected from multiple sources are organized. The vehicle detection algorithm filters out samples where vehicles were present in the detection area. The vehicle tracking algorithm uses the data of the detected vehicles to predict the trajectory of the vehicle using the similarity between vehicle detections

A. Vehicle detection

As the data was collected from multiple devices it was rearranged based on the timestamps sequentially and normalized.

Different preprocessing techniques were experimented with during the course of the research including, FFT, Wavelets, entropy, peak and valley counts, and duration of states. Further, history was added artificially to the traditional machine learning models by feeding the actual state of the previous to the current input. Performance obtained by adding one or many of the different preprocessing techniques did not show a significant increase in improving model performance.

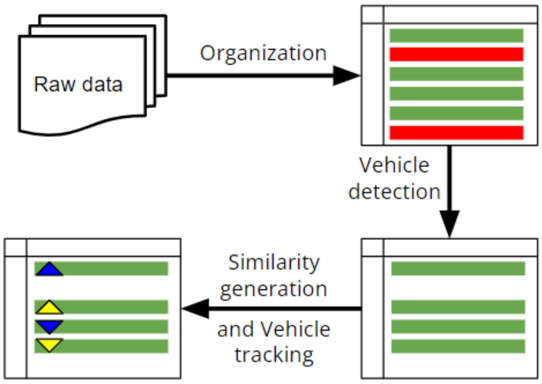

Fig. 4: System diagram.

Multiple models for binary classification of vehicle presence were experimented with and compared. A training to testing data ratio of 8:2 was used.

Traditional machine learning models; Stochastic gradient descent (SGD), kNN, SVM, Decision trees (DTC), Random forests (RFC), and XGBoost were implemented.

For NNs; MLP is a classical NN. An MLP NN consisting of 2 layers with 100 and 20 nodes respectively was used.

Long short term memory (LSTM) is a type of recurrent NN that allows information to persist [18] which is achieved by a chain structure with NNs and memory blocks. These perform well especially on data with a time axis. A LSTM NN used normalized data was input. An embedding layer converted integers into dense vectors of fixed size. Contained two LSTM layers with 100 nodes with sigmoid activation function and a deep connected NN layer with 1 node with sigmoid activation function. Binary crossentropy was used as the loss function and the output generated was the presence of a vehicle.

TABLE II: Vehicle detection model performance.

Type	**Accuracy %**	**Sensitivity**	**Specificity**	*Training accuracy %*
SGD	52	0.03	1	53
kNN	86	0.76	0.96	89
SVM	59	0.41	0.77	76
DTC	92	0.91	0.92	96
RFC	95	0.94	0.95	97
XGBoost	93	0.91	0.95	90
MLP NN	65	0.71	0.59	75
LSTM NN	96	0.94	0.98	99

Comparing the results from Table II, it was noted that LSTM NN outperforms every other model. Accuracy obtained from the testing dataset is only slightly lower than from the training dataset for all models showing that the models do not overfit to the training set. Therefore LSTM NN was selected as the optimal model.

As the data is time series, LSTMs perform better due to their use of memory. Artificially adding a memory unit to the other machine learning models does not reproduce the same impact as inputs are taken as mutually exclusive.

With successful identification of the presence of vehicles, the vehicle tracking pipeline is triggered.

B. Vehicle tracking

Previous researches show RF can be used for unique vehicle identification where the prediction would be the specific vehicle [16], [17]. These were classification problems where a vehicle should be identified from a pool of preset vehicles which requires information about the vehicle in question being present in the training data.

This research stands apart from previous work by applying vehicle tracking for any vehicle using the intersection. Each vehicle is not uniquely identified instead, features are used to match detections of the same vehicle at different locations to determine the trajectory of the vehicle.

The complexity of using RF for vehicle tracking is very high as background noise significantly affects the RF readings while the distance from the transmitter to receiver also affects the readings mainly when readings are taken from multiple devices.

As a vehicle moves through the intersection from the entrance to the exit, it would pass through two vehicle detection areas, which is illustrated by Fig. 5. As a vehicle is detected, the snippet of data from the detection is added to the detected vehicles dataset. The moving vehicle (red arrow) triggers the detection at lane A and then lane B. Fig. 7a illustrates the vehicle detection dataset for this with the color red representing the detections of the red arrow vehicle (color black represents other detections). As the dataset is ordered by time, given there were no other detections triggered, the first detection which was at lane A is the vehicle entering the intersection and the detection at B is the vehicle exiting. In this isolated scenario, we can confirm the path taken by the vehicle is from lane A to lane B without any further examination. This assumes that there is only one vehicle that occupies the intersection at any given time which certainly fails in any real world application.

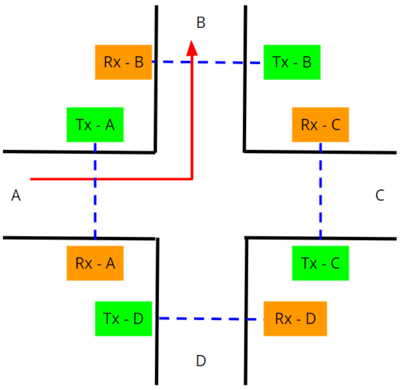

Fig. 5: Single vehicle occupying the intersection.

Expanding on the earlier example, Fig. 6 illustrates multiple vehicles occupying the intersection simultaneously. Fig. 7b shows the vehicle detections for this. Analyzing the dataset, events occur in the following sequence; Red enters, Yellow enters, Red exits, Grey enters, Grey exits, and finally Yellow exits.

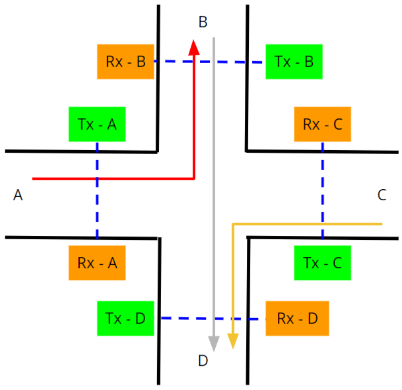

Fig. 6: Multiple vehicles occupying the intersection.

With the increased complexity of the detections, the earlier method of basing the path taken by the vehicle solely on time of detection fails as the yellow enters the intersection before red exits while grey enter and exit happen after yellow enters but before yellow exits.

(a) Single vehicle. (b) Multiple vehicles.

Fig. 7: Detection data captured.

To handle the complexity introduced by the increase in vehicle occupancy of the intersection, the research investigated machine learning techniques that could be used to re-identify the vehicle based on the signal patterns. This investigation led to the use of similarity scores generated by NNs.

Fig. 8 illustrates the system diagram of the vehicle tracking algorithm. As the first sample of the dataset in Fig. 8 is of the red and there are no vehicle detections prior to this, this sample is considered the entry of red. The task of the Siamese NN is to compare the data from the anchor (in this case the red, first sample) to the samples following it to find the sample representing the exit of red as the exit of red must come after the entry. The samples to which the anchor is compared are called the contested. Comparing the anchor with all the samples following it is ineffective computationally while also not being a common occurrence in real data. The number of contested samples is determined by the attributes of the intersection and in the case of this dataset, the average number of vehicles being present at a given time was 3.63 with a mode of 3 and a median of 4. Four was used as the number of vehicles occupying the intersection at a given time therefore the number for contested was set to 7 (a vehicle will create two detections, therefore 8 vehicle detections should be processed from which the first will be the anchor and the following are the contested samples). The similarity of the anchor to each of the contested are then computed. The contested generating the highest similarity to the anchor embedding is selected as the exit for the vehicle, i.e. from the example in Fig. 8, the second contested (third item in the dataset) generated the highest similarity, therefore, being selected as the exit. The matched samples are then dropped from the dataset and the process is repeated by considering the current first sample as the anchor.

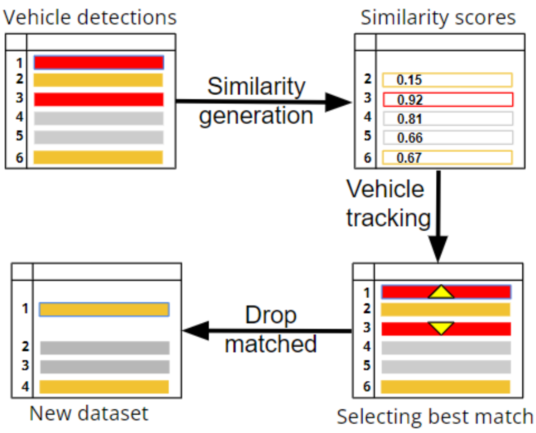

Fig. 8: Vehicle tracking algorithm.

A comparison of a CNN, transformer, two types of siamese NNs, and a siamese NN with transfer learning was conducted. Table III shows the architectures. A training to testing data ratio of 8:2 was used.

As the inputs for the models, the Fast Fourier Transforms (FFT) of the time series data were computed.

The CNN and Transformer take the anchor and contested as the input and predict the similarity score.

Siamese NNs [19] use two instances of the same NN loaded with the same weights on two different input vectors to generate two output vectors of which the similarity is computed. Siamese NN type 1 uses two identical CNNs to generate embeddings for each of the inputs (anchor and contested). The resulting embeddings are subtracted and a final layer uses this to predict the similarity score.

Siamese NN type 2 uses triplet training. Triplet training uses three inputs per sample for training which are the anchor, a positive, and a negative [20]. This enables the model to be trained to increase the distance from the negative to the anchor and decrease the distance from the positive to the anchor during the training process. The output generated is n embedding and cosine similarity is used to generate the similarity score between the embeddings of the anchor and contested. Siamese NN type 2a has the architecture given in Tab. III by siamese NN type 2.

Using transfer learning to decrease the training time and amount of data is quite common [21]. Resnet50 pretrained on Imagenet data is a widely used backbone [22]. Siamese NN type 2b uses the same architecture as 2a with two key differences. This uses resnet50 as the backbone and layers are added on top of it which are trained (i.e. the weights of the backbone aren't changed). As resnet50 was trained on image data, the inputs should have image-like dimensions. The first approach was to rearrange the data to have an image-like shape. The performance of the model was quite poor, showing that the features were not being identified effectively. The second approach was converting the data into image plots, which showed comparatively superior performance. Siamese NN type 2b uses the resnet50 to generate embeddings which are followed by the layers given in Tab. III Siamese NN type 2.

TABLE III: Vehicle detection machine learning model performance.

CNN	**Transformer**
Input: time series and FFT of both the anchor and contested.	Input: time series and FFT of both the anchor and contested.
Binary crossentropy loss function.	Binary crossentropy loss function.
3 convolutional layers with batch normalization ReLU activation.	4 transformer blocks with 256 heads and MLP with 128 nodes with ReLU activation.
Output: Deep connected NN with 1 node and sigmoid activation.	Output: Deep connected NN with 1 node and sigmoid activation.
Siamese NN type 1	**Siamese NN type 2**
Input: time series and FFT.	Input: time series and FFT.
Binary crossentropy loss function.	Triplet loss function.
Deep connected NN with 100 nodes and ReLU activation.	Deep connected NN with 128 nodes and ReLU activation.
Deep connected NN with 10 nodes and ReLU activation.	Deep connected NN with 64 nodes and ReLU activation.
Deep connected NN with 2 nodes and ReLU activation.	Output: Deep connected NN with 32 nodes and linear activation.
Substraction	
Output: Deep connected NN with 1 node and sigmoid activation.	

TABLE IV: Path tracking machine learning model performance.

Type	Accuracy %	Precision	Recall	*Training accuracy %*
CNN	54	0.86	0.21	55
Transformer	52	0.30	0.74	56
Siamese NN 1	70	0.70	0.70	78
Siamese NN 2a	75	0.75	0.75	76
Siamees NN 2b	58	0.58	0.58	65

Table IV contains the metrics obtained from the different machine learning models. The CNN and transformer do poorly compared to the siamese NNs. This was expected as the objective was to identify the most similar sample out of the contested samples, which is where the siamese models are great at. The model created with transfer learning does not perform as well because the pretrained model was trained on an image dataset whereas the current application is time series. Converting the time series data into image plots does increase the effectiveness slightly but still does not perform as well as the custom model which could be attributed to the type of image data on which the pretrained model was trained on. Siamese NN type 2a was selected as the optimal model.

Similarity scores generated by the NN were then used in the path tracking algorithm to predict the trajectory of vehicles using the intersection. An accuracy of 89% was achieved by the path tracking algorithm (this is different from the accuracy of the model because of the aptitude added by the vehicle tracking algorithm).

V. Conclusion

The research investigates the use of RF data to detect and track vehicles using machine learning. Using RF as opposed to video or manpower decreases the size of data storage and bandwidth required while also being more privacy-preserving in nature. Multiple preprocessing techniques and machine learning models were experimented where normalization was selected for preprocessing and LSTM NNs outperformed compared models by achieving an accuracy of 96% for vehicle detection. For path tracking, vehicle detections were fed into the vehicle tracking system with their FFT representations. Siamese NN with a custom model and triplet learning outperformed all competitors by achieving an accuracy of 75%. The similarity scores generated by the siamese model were used to predict the trajectory of the vehicle through the intersection which achieved an accuracy of 89%.

The results generated by the two systems can be used to calculate the intersection occupancy metrics (i.e. number of vehicles using a particular path, vehicle flow rate, vehicle flow rate per lane, and time taken to cross the intersection). This enables the use of RF for low-powered cost-effective systems to be used in traffic surveys and real-time monitoring.

The system is limited to working only on lanes with a single file of vehicles. Moreover, increasing the intricacy of the intersection would introduce additional RF noise that would result in reduced performance.

This system would not outright replace the use of videos, instead being used in combination with them for improved results or where recording videos is not viable (i.e. privacy concerns, low budget). Future work would aim to improve the models by using data from diverse locations followed by handling lanes with multiple files of vehicles.

References

[1] Minnesota Department of Transportation, "Traffic engineering manual," 2015. [Online]. Available: https://www.dot.state.mn.us/trafficeng/publ/tem/index.html

[2] B. Sliwa, T. Liebig, T. Vranken, M. Schreckenberg, and C. Wietfeld, "System-of-systems modeling, analysis and optimization of hybrid vehicular traffic," in *2019 IEEE International Systems Conference (SysCon)*. IEEE, 2019, pp. 1–8.

[3] F. A. Malik, M. A. Lone, R. A. Qasab, and A. Gul, "Traffic census and analysis a case study," *International Journal of Research in Engineering and Technology (IJRET). Volume5, Issue03*, pp. 69–76, 2016.

[4] G. Wang, D. Xiao, and J. Gu, "Review on vehicle detection based on video for traffic surveillance," in *2008 IEEE international conference on automation and logistics*. IEEE, 2008, pp. 2961–2966.

[5] N. Midler, "As facial recognition draws scrutiny nationwide, stanford research raises questions closer to home," Jul 2020. [Online]. Available: https://www.stanforddaily.com/2020/07/28/as-facial-recognition-draws-scrutiny-nationwide-stanford-research-raises-questions-closer-to-home/

[6] D. Harwell, "Police can keep ring camera video forever and share with whomever they'd like, amazon tells senator," Nov 2019. [Online]. Available: https://www.washingtonpost.com/technology/2019/11/19/police-can-keep-ring-camera-video-forever-share-with-whomever-theyd-like-company-tells-senator/

[7] M. Maximov, I. Elezi, and L. Leal-Taixé, "Decoupling identity and visual quality for image and video anonymization," in *Proceedings of the Asian Conference on Computer Vision*, 2022, pp. 3637–3653.

[8] D.-T. Iancu and A.-M. Florea, "An improved vehicle trajectory prediction model based on video generation," *STUDIES IN INFORMATICS AND CONTROL*, vol. 32, no. 1, pp. 25–36, 2023.

[9] S. Gupta, A. Hamzin, and A. Degbelo, "A low-cost open hardware system for collecting traffic data using wi-fi signal strength," *Sensors*, vol. 18, no. 11, p. 3623, 2018.

[10] B. Sliwa, N. Piatkowski, and C. Wietfeld, "The channel as a traffic sensor: Vehicle detection and classification based on radio fingerprinting," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7392–7406, 2020.

[11] M. Haferkamp, B. Sliwa, and C. Wietfeld, "A low cost modular radio tomography system for bicycle and vehicle detection and classification," in *2021 IEEE International Systems Conference (SysCon)*. IEEE, 2021, pp. 1–7.

[12] N. Kassem, A. E. Kosba, and M. Youssef, "Rf-based vehicle detection and speed estimation," in *2012 IEEE 75th Vehicular Technology Conference (VTC Spring)*. IEEE, 2012, pp. 1–5.

[13] Y. Ding, B. Banitalebi, T. Miyaki, and M. Beigl, "Rftraffic: a study of passive traffic awareness using emitted rf noise from the vehicles," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, pp. 1–14, 2012.

[14] J. Lan, Y. Xiang, L. Wang, and Y. Shi, "Vehicle detection and classification by measuring and processing magnetic signal," *Measurement*, vol. 44, no. 1, pp. 174–180, 2011.

[15] B. Sliwa, N. Piatkowski, M. Haferkamp, D. Dorn, and C. Wietfeld, "Leveraging the channel as a sensor: Real-time vehicle classification using multidimensional radio-fingerprinting," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 898–905.

[16] R. A. Kerekes, T. P. Karnowski, M. Kuhn, M. R. Moore, B. Stinson, R. Tokola, A. Anderson, and J. M. Vann, "Vehicle classification and identification using multi-modal sensing and signal learning," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*. IEEE, 2017, pp. 1–5.

[17] X. Dong, H. Weng, D. G. Beetner, T. H. Hubing, D. C. Wunsch, M. Noll, H. Göksu, and B. Moss, "Detection and identification of vehicles based on their unintended electromagnetic emissions," *IEEE Transactions on Electromagnetic Compatibility*, vol. 48, no. 4, pp. 752–759, 2006.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] G. Koch, R. Zemel, R. Salakhutdinov *et al.*, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2. Lille, 2015, p. 0.

[20] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International workshop on similarity-based pattern recognition*. Springer, 2015, pp. 84–92.

[21] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.