Traffic Monitoring using an Object Detection Framework with Limited Dataset

Vitalijs Komasilovs[1], Aleksejs Zacepins[1], Armands Kviesis[1] and Claudio Estevez[2]
[1]*Department of Computer Systems, Faculty of Information Technologies,*
Latvia University of Life Sciences and Technologies, Jelgava, Latvia
[2]*Department of Electrical Engineering, Universidad de Chile, Santiago, Chile*

Keywords: Traffic Monitoring, Smart City, Video Processing, Tensorflow, Object Detection.

Abstract: Vehicle detection and tracking is one of the key components of the smart traffic concept. Modern city planning and development is not achievable without proper knowledge of existing traffic flows within the city. Surveillance video is an undervalued source of traffic information, which can be discovered by variety of information technology tools and solutions, including machine learning techniques. A solution for real-time vehicle traffic monitoring, tracking and counting is proposed in Jelgava city, Latvia. It uses object detection model for locating vehicles on the image from outdoor surveillance camera. Detected vehicles are passed to tracking module, which is responsible for building vehicle trajectory and its counting. This research compares two different model training approaches (uniform and diverse data sets) used for vehicle detection in variety of weather and day-time conditions. The system demonstrates good accuracy of given test cases (about 92% accuracy in average). In addition, results are compared to non-machine learning vehicle tracking approach, where notable vehicle detection accuracy increase is demonstrated on congested traffic. This research is fulfilled within the RETRACT (Enabling resilient urban transportation systems in smart cities) project.

1 INTRODUCTION

Vehicle recognition and tracking of its route is important task in ensuring smart traffic concept in the modern cities, because it can provide important information about traffic conditions, like velocity distribution and density of vehicles, as well as detect possible traffic congestion.

Vehicle recognition can be accomplished using different techniques and approaches, like pressure sensors, inductive loops (Bhaskar et al., 2015), magnetoresistive sensors (Yang and Lei, 2015), radars (Wang et al., 2016), ultrasound (Sifuentes et al., 2011), infrared (Rivas-López et al., 2015; Iwasaki et al., 2013), stereo sensors (Lee et al., 2011) etc. Recently, thanks to computing hardware performance boost and development of advanced machine learning techniques, vehicles can be reliably recognized on images or video streams using machine vision approach. During recent years, the use of vision-based traffic systems has increased in popularity, both in terms of traffic monitoring and control of autonomous cars. Video cameras ensure traffic surveillance and are components of intelligent transport system (ITS). Such cameras helps to identify vehicles that violate the traffic rules, e.g drive in a forbidden direction or pass the crossroad through the red light, etc. The use of image-based sensors and computer vision techniques for data acquisition on the traffic of vehicles has been intensely researched in the recent years (Tian et al., 2011).

Usage of video cameras instead of other sensors has several advantages: easy maintenance, high flexibility, compact hardware, and software structure, which enhance the mobility and performance (Thomessen, 2017). On the contrary, intrusive traffic sensing technologies cause traffic disruption during its installation process and are unable to detect slow or static vehicles (Mandellos et al., 2011). Ethical and privacy considerations related to video footage use for traffic monitoring are out of scope of current research. Usually, these topics are regulated by state or local municipality laws (e.g. in Latvia there are outdoor signs warning that video recording takes place in particular area).

The development of deep neural networks (DNNs) has contributed to a significant improvement of the computer vision tasks during the recent years. Neu-

Komasilovs, V., Zacepins, A., Kviesis, A. and Estevez, C.
Traffic Monitoring using an Object Detection Framework with Limited Dataset.
DOI: 10.5220/0007586802910296
In *Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2019)*, pages 291-296
ISBN: 978-989-758-374-2

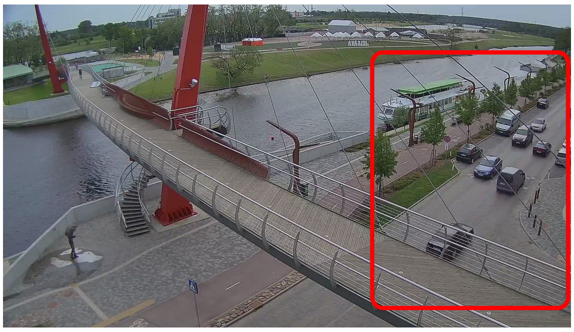


Figure 1: An example of video frame with marked area of interest

ral networks refer to a way of approximating mathematical functions inspired by the biology of the brain, and hence the name neural. The neural network method is based upon supervised training on object with known properties, and has the capability to extend this trained knowledge to detect unknown object properly. Convolutional Neural Networks (CNNs) have recently been applied to various computer vision tasks such as image classification (Chatfield et al., 2014; Simonyan and Zisserman, 2014), semantic segmentation (Hong et al., 2015; Long et al., 2015), object detection (Girshick et al., 2014), and many others (Noh et al., 2016; Toshev and Szegedy, 2014). There are also many developed solutions and scientific publications related to usage of neural networks in vision systems for vehicle recognition (Bengio et al., 2015; Huval et al., 2015).

The aim of this research is to develop and demonstrate application of machine learning framework for real-time traffic monitoring based on publicly available video stream, as authors address the underestimated availability of video information on urban roads, which can be definitely used for traffic flow monitoring. For example, in authors' hometown Jelgava, Latvia, there are more than 200 surveillance cameras installed already. The live video for this research is obtained from Jelgava municipality web page[1] from stationary camera positioned aside the road on the building wall by the address 5 J. Cakstes Blvd., Jelgava, Latvia (see Fig. 1).

Vehicle traffic in the video occurs in the diagonal direction, from top right (farthest from the camera) to bottom left (closest to the camera), and vice versa. Video has Full HD resolution of 1920x1080 px at 30 frames per second. Apart from other objects (e.g. wires, bridge, pedestrians, buildings, etc.) video stream contains regular two-way (one lane in each direction) road of Jelgava city.

In this article authors consider an approach that provides vehicle detection and its trajectory registration. The problem to recognize and monitor vehicles is usually separated into three main operations; detection, tracking and classification. Generally, detection is the process of localizing objects (vehicles) in the scene. Tracking is the problem of localizing the same object over adjacent and consecutive frames, and classification is the process of categorizing the objects. In this research authors do not classify vehicles because traffic predominantly consists of passenger cars on the given road. This research focuses the following: a) getting an image from a live video stream; b) detection of the vehicle(s) on the image applying machine learning approach; c) tracking vehicles across consecutive frames; d) registering and counting vehicles traveling in each direction.

Current work is related to the previous authors' research, where vehicles are detected by applying background modeling and motion detection methods (Komasilovs et al., 2018). Experimental results of both approaches are compared and analyzed in the results and discussion section of this publication.

2 MATERIALS AND METHODS

Basic work flow of the developed solution for vehicle traffic detection and its trajectory registration is shown in Fig. 2.

Input frames are extracted directly from YouTube Full HD stream (1920x1080), cropped to area of interest (576x648) and pushed to further processing, described in subsections below. Solution is implemented and tested using Python 3.5.2 environment.

To facilitate processing of live video stream in real-time manner threaded application structure is used. Dedicated thread is extracting frames from the live stream, cropping and putting into limited size buffer (FIFO queue of size 30). Another thread is running the vehicle detection process described in the next section. Taking into account that the detection process is significantly slower than the frame rate of the video, the buffer exceeds its limited size and older frames are being discarded in favor of newer frames. Such shifting of extracted frames within the buffer allows consistent supply of actual frames for vehicle detection process regardless of its current performance and live stream networking peculiarities with a lag proportioned to the buffer size (approx. 1 second for 30 fps video).

[1] http://www.jelgava.lv/lv/pilseta/tiessaistes-kamera/

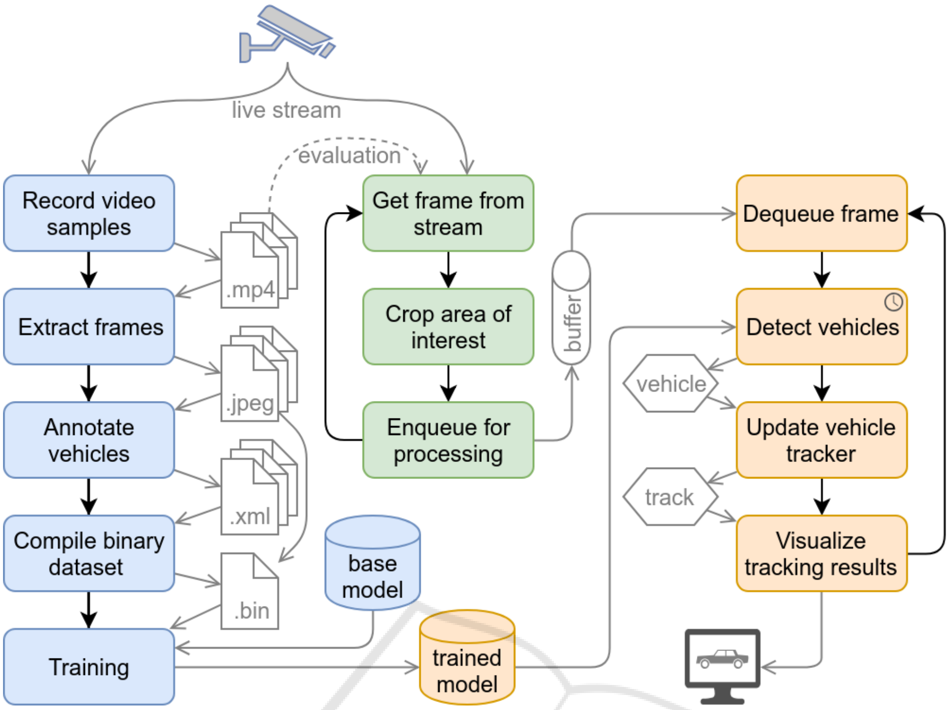


Figure 2: Principal process work flow.

2.1 Vehicle Detection

For the vehicle detection task open source machine learning framework Tensorflow[2] is used. The framework provides tools for flexible numeric computation and machine learning across variety of platforms. As a learning base SSD Mobilenet V1 model (Howard et al., 2017) pre-trained on COCO dataset (Lin et al., 2014) is used. The model structure is developed with the aim for faster execution (object detection) on commodity hardware. COCO dataset from other hand provides highly diverse objects and their contexts. Application of such pre-trained model for domain specific object detection results in much faster training process and robust results.

This model is further fine-trained on custom dataset containing training examples extracted from the camera images used in experiments. Application of models pre-trained on large datasets makes training faster and more reliable, it also simplifies the custom dataset preparation due to the huge number of background examples included into the original model.

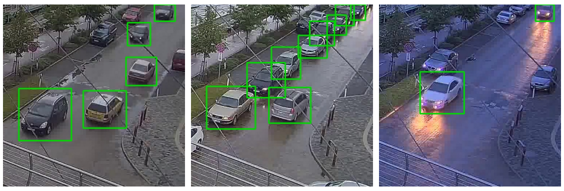
For experimental purposes a number of videos is recorded from the aforementioned outdoor camera, including different day time and weather conditions. In general, preparation of versatile and balanced training dataset is non-trivial process, requires deep understanding of target use-case domain and demand significant time and effort (e.g. COCO dataset consists of more then 200 thousands labeled images). From the other hand, if the model use-case is bounded to particular problem, then the model training can be done using *limited dataset*. Representative samples selected to training dataset can successfully cover peculiarities of the given problem. Authors use two different approaches (experiments) for preparing training datasets.

As the first experiment, for model training purposes **single** 10 minute long video recorded at 13:00 o'clock is used. Authors extracted one frame every 3 seconds (200 frames in total) and manually annotated them marking vehicles on the road. Car annotation was performed using the free software *labelImg*[3]. Taking out frames without vehicles, prepared dataset contained 137 frames. The data set preparation gave a total of 231 annotated vehicles.

The second experimental dataset is created with aim to increase the **diversity** of samples. 10 minute long videos recorded at different times of the day were used to ensure variance of illumination, weather and road conditions. Frames are extracted with rate of 1^{-10} frames per second (60 frames per video, 480 frames in total). After a manual vehicle annotation the dataset contained 277 frames with 525 vehicles.

[2] https://www.tensorflow.org/

[3] https://github.com/tzutalin/labelImg



(a) typical traffic (b) congestion (c) dusk

Figure 3: Training samples.

Some of the annotated vehicles are shown in Fig.3.

To increase the diversity of training dataset, various augmentation methods are applied to the dataset, such as random cropping, flipping, change in hue, contrast, brightness and saturation. This approach simulates different scenarios that occurs during the day, e.g illumination and vehicle orientation.

After the dataset is prepared, the next step is training and evaluation of vehicle detection model. There are several challenges within this process: with less training data, the parameter estimations will have greater variance, and on the other hand, with less evaluation data, the performance statistics will have greater variance. Ideally, that variance should be as small as possible for both.

Training of the model is performed on the cloud using the Jupyter notebook environment Google Collaboratory[4], which provides pre-configured access to computational power needed for machine learning technologies. The real-time vehicle tracking part of the system uses trained model and runs on local commodity type computer (Intel i5, 16 GB RAM, no GPU). For evaluation purposes full videos are used and fed into real-time vehicle tracking module similar to live stream.

2.2 Vehicle Tracking

Vehicle tracking task stands for the problem of following the same vehicle through multiple subsequent frames and can include various methods for trajectory assignment, motion modeling, tracking result filtering, and finally vehicle counting.

Unlike authors' previous research (Komasilovs et al., 2018), where vehicles tracking was relying on motion detection and required advanced vehicle motion modeling and prediction. The current research uses simplified approach. The vehicle detection model is applied on a provided video frame (still image) without any information about previous frames. Coordinates of each detected vehicle on subsequent frames are stored by a tracking module. Using a modified Hungarian algorithm for linear sum assignment problem, vehicle detections (coordinates) are assigned to the appropriate tracks by minimizing the sum of distances between current detection and last tracked position. Linear regression is applied on raw trajectory points resulting in a straight line approximation of tracked vehicle trajectory. Example of vehicle tracking is shown in Fig.4.

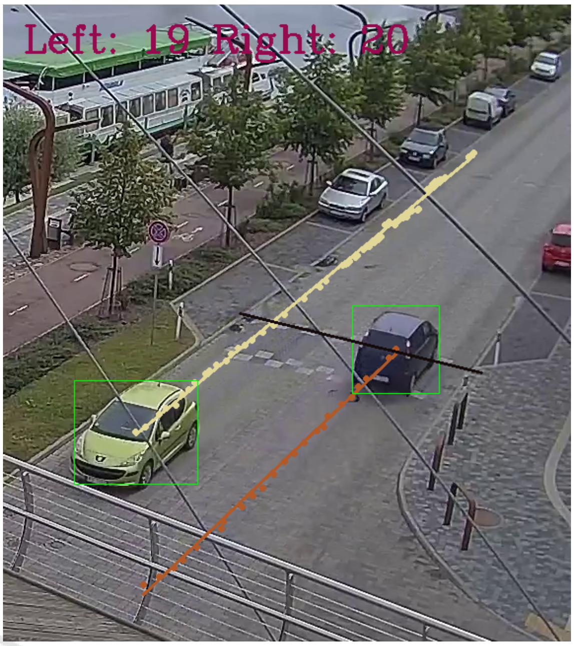


Figure 4: Principal process work flow.

Taking into account reliability of vehicle detection no additional tracking enhancements are applied. Vehicles traveling in each direction are counted at the moment when their linearized trajectory is crossing (intersecting) a pre-defined registration line.

3 RESULTS AND DISCUSSION

This section describes the achieved results and their evaluation for the proposed approach. In addition, results are compared with previous authors' results achieved by using motion tracking approach for vehicle detection (Komasilovs et al., 2018).

3.1 Setup of the Experiment

For proof of the concept and evaluation of the proposed approach eight 10-minutes long video fragments are used, which are recorded from the video stream at different times of the day. On each video fragment, vehicles are manually counted for ground truth reference. Then, each fragment is processed using the proposed solution and results are collected

[4] https://colab.research.google.com/

Table 1: Accuracy evaluation summary.

	Ground truth (number of vehicles)		Detection accuracy (first experiment)		Detection accuracy (second experiment)		Motion tracking accuracy (Komasilovs et al., 2018)	
Nr.	to left	to right	to left	to right	to left	to right	to left	to right
1	25	13	100%	92%	92%	77%	100%	100%
2	63	25	95%	96%	92%	88%	97%	96%
3	55	23	95%	96%	89%	83%	98%	100%
4	50	54	96%	93%	98%	98%	100%	98%
5	52	30	100%	93%	100%	97%	98%	93%
6	80	71	98%	94%	80%	94%	76%	97%
7	49	29	98%	83%	88%	72%	96%	93%
8	37	18	86%	89%	97%	94%	95%	94%

(see Table 1). Table represents eight different 10 minute long video fragments (test cases) recorded at time from 07:00 to 21:00, ground truth number of vehicles traveled in each direction (to left and to right) during these fragments, and accuracy of vehicle detection and tracking algorithms achieved on given fragments.

3.2 Discussion

For the first experiment, the model was trained only on about 130 frames from video recorded during midday (13:00 to 13:10), but the assessment was performed on video fragments from a variety of daytime conditions (from 07:00 to 21:00). Taking into account these peculiarities, results can be considered as acceptable. Especially, noticeable improvement is on handling congested traffic (test case 6, from 76% to 98% accuracy comparing with previous approach), where motion tracking method was not able to reliably detect vehicles. On other hand, decrease in accuracy for cases 7 and 8 can be explained by highly different image parameters from training set (dusk resulted in blueish colors dominating in the picture).

For the second experiment, the model is trained on more then twice as many frames from a variety of videos with different day time and weather conditions. The outcomes of the second experiment demonstrate a worse average accuracy when compared with the first experiment. The only notable accuracy increase is observed in test case 8 (video recorded at 21:00-21:10 during dusk). This can be explained by the fact that the training dataset contained highly diverse examples. Due to infrequent training frame extraction (1^{-10} frames per second) vehicles were not annotated during entire traveling trajectory, but instead different vehicles appeared in different positions on usual trajectory. Adding variety of weather and daylight conditions (vehicle backgrounds), all these peculiarities impeded vehicle generalization by the model and it learned only training samples.

It is worth to mention the performance of both methods applied for vehicle tracking. Motion tracking method is able to process every frame of *30 fps* video in real time on commodity type computer. Contrary, deep learning method takes about *200 ms* per frame to run vehicle detection model on the same hardware. Taking into account the comparable accuracy of these methods, the deep learning detection model is viable only when *congested traffic* monitoring is the target use case or it is executed on better (GPU equipped) hardware.

4 CONCLUSIONS

Vehicle detection model was trained using a relatively small training set and personnel hours, and achieved results (92% vehicle detection and tracking accuracy in average) can be considered as applicable for the given use case. In particular, this can be explained by the fact that pre-trained SSD MobileNet V1 model is used as a base for the fine training vehicle detection model.

This vehicle detection approach cannot be treated as universal because the training set is bound to peculiarities of particular locations, camera view angles, color modes and other parameters. For developing a universal vehicle detection model a significantly larger dataset is required. On the other hand, small training set preparation and model training is a simple process and takes relatively small amount of time, thus it can be repeated for each needed location separately.

Comparing achieved results with other vehicle tracking methods, like motion tracking, it can be concluded that machine learning is not always a viable option, as it requires significantly more processing power, and other methods can provide similar or even better results when tuned properly. Also machine learning approaches become more suitable for complex use cases, where besides vehicle tracking additional refinements are needed, such as object classification, safety rules violation detection, etc.

ACKNOWLEDGEMENTS

Scientific research, publication, and presentation are supported by the ERANet-LAC Project ”Enabling resilient urban transportation systems in smart cities” (RETRACT, ELAC2015/T10-0761).

REFERENCES

Bengio, Y., Goodfellow, I. J., and Courville, A. (2015). Deep learning. *Nature*, 521(7553):436–444.

Bhaskar, L., Sahai, A., Sinha, D., Varshney, G., and Jain, T. (2015). Intelligent traffic light controller using inductive loops for vehicle detection. In *Next Generation Computing Technologies (NGCT), 2015 1st International Conference on*, pages 518–522. IEEE.

Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

Hong, S., Noh, H., and Han, B. (2015). Decoupled deep neural network for semi-supervised semantic segmentation. In *Advances in neural information processing systems*, pages 1495–1503.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Huval, B., Wang, T., Tandon, S., Kiske, J., Song, W., Pazhayampallil, J., Andriluka, M., Rajpurkar, P., Migimatsu, T., Cheng-Yue, R., et al. (2015). An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*.

Iwasaki, Y., Kawata, S., and Nakamiya, T. (2013). Vehicle detection even in poor visibility conditions using infrared thermal images and its application to road traffic flow monitoring. In *Emerging Trends in Computing, Informatics, Systems Sciences, and Engineering*, pages 997–1009. Springer.

Komasilovs, V., Zacepins, A., Kviesis, A., Peña, E., Tejada-Estay, F., and Estevez, C. (2018). Traffic monitoring system development in jelgava city, latvia. In *Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems - Volume 1: RESIST,*, pages 659–665. INSTICC, SciTePress.

Lee, C., Lim, Y.-C., Kwon, S., and Lee, J. (2011). Stereo vision-based vehicle detection using a road feature and disparity histogram. *Optical Engineering*, 50(2):027004.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Mandellos, N. A., Keramitsoglou, I., and Kiranoudis, C. T. (2011). A background subtraction algorithm for detecting and tracking vehicles. *Expert Systems with Applications*, 38(3):1619–1631.

Noh, H., Hongsuck Seo, P., and Han, B. (2016). Image question answering using convolutional neural network with dynamic parameter prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 30–38.

Rivas-López, M., Gomez-Sanchez, C. A., Rivera-Castillo, J., Sergiyenko, O., Flores-Fuentes, W., Rodríguez-Quiñonez, J. C., and Mayorga-Ortiz, P. (2015). Vehicle detection using an infrared light emitter and a photodiode as visualization system. In *Industrial Electronics (ISIE), 2015 IEEE 24th International Symposium on*, pages 972–975. IEEE.

Sifuentes, E., Casas, O., and Pallas-Areny, R. (2011). Wireless magnetic sensor node for vehicle detection with optical wake-up. *IEEE Sensors Journal*, 11(8):1669–1676.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Thomessen, E. A. (2017). Advanced vision based vehicle classification for traffic surveillance system using neural networks. Master's thesis, University of Stavanger, Norway.

Tian, B., Yao, Q., Gu, Y., Wang, K., and Li, Y. (2011). Video processing techniques for traffic flow monitoring: A survey. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 1103–1108. IEEE.

Toshev, A. and Szegedy, C. (2014). Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660.

Wang, X., Xu, L., Sun, H., Xin, J., and Zheng, N. (2016). On-road vehicle detection and tracking using mmw radar and monovision fusion. *IEEE Transactions on Intelligent Transportation Systems*, 17(7):2075–2084.

Yang, B. and Lei, Y. (2015). Vehicle detection and classification for low-speed congested traffic with anisotropic magnetoresistive sensor. *IEEE Sensors Journal*, 15(2):1132–1138.