ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

DESIGN OF A LOW-COST AND FLEXIBLE PEDESTRIAN VOLUME INVESTIGATOR WITH RASPBERRY PI AND MACHINE LEARNING

Junghoon Lee[1] and Cheol Min Kim[2]
[1]Department of Computer Science and Statistics, Jeju National University, Jeju-City, Republic of Korea
[2]Department of Computer Education, Jeju National University, Jeju-City, Republic of Korea
E-Mail: jhlee@jejunu.ac.kr

ABSTRACT

This paper designs a low-cost pedestrian volume investigator orchestrating Raspberry Pi nodes, wireless network connectivity, and machine learning techniques. Under the control of a coordinator, 3 sensors capture the distance to the closest object in the target space for both learning and estimation. To obtain learning patterns, a human operator initiates a data acquisition transaction and records the number of objects he or she observes. With the set of learning patterns, each of which consists of 3 distance measurements and the number of objects, we build a 3-layer artificial neural network model with 3 inputs, 20 hidden nodes, and 1 output. Next, the investigator periodically collects the sensor readings and estimates the number of objects. The simulation study shows the error size hardly exceeds 1 object until the number of objects is 8, indicating that the proposed scheme, as a new Raspberry Pi application, can economically trace thenumber of objects with reasonable accuracy and flexibility.

Keywords: raspberry Pi, machine learning, pedestrian volume investigation, master-slave cooperation, flexible, low-cost deployment.

1. INTRODUCTION

Low-cost sensor nodes, such as Raspberry Pi and Arduino, make it possible to build tremendously diverse sensor networks of their own purposes [1]. They are explosively extending their application areas along with the release of new sensors as well as the improvement of network bandwidth. Particularly, not only recent wireless channels can deliver a large volume of data in reasonable time but also a sensor network can necessarily integrate sophisticated computer algorithms especially in machine learning and big data analysis [2]. In this sensor network, sensor nodes are also computing devices embedding CPUs and memory as well as installing an operating system, thus the group of sensor nodes provides a distributed computing framework, efficiently employing ever-growing communication technologies and appropriate software solutions customized for a specific environment [3]. For example, in case of vehicle-based sensor networks, special purpose database systems can store and conduct queries even for fast-moving objects [4].

Among many diverse sensor network applications, this paper focuses on the pedestrian volume investigation, which measures and also models how many persons or other objects pass through the target area. Cheap sensor nodes and network connectivity can efficiently reduce the cost and complexity of measuring equipment, not much sacrificing the accuracy of object detection. Existing methods are usually based on beacons or video analysis [3,5]. However, not every object carries a beacon device in practice or allows it to be identified by an unknown entity without its agreement. Hence, beacon methods are used just in restricted conditions. Additionally, video acquisition modules may get quite expensive and not easy to manage safely especially in outdoor areas. Here, image transfer to a central analysis module, which is highly likely to be located far away from a sensor module, requires not a little network bandwidth, sometimes requiring the installation of exclusive channels.

In the meantime, a sensor node-based approach can flexibly adapt to a different environment by placing or relocating a set of sensors, considering geographic restrictions as long as the network connectivity is stably maintained within the small target area, for example, a 5 *m* by 5 *m* square. Here, more sensors can be employed and a different sensing method can be taken together for better performance. A distance sensor costs just a few US dollars. In addition, on-line and off-line data processing can improve the accuracy by incrementally upgrading a constituent module, be it hardware or software, with a better one. Such an application can also work for any other moving objects including vehicles, bicycles, wild animals, and the like [6]. Moreover, an open architecture mainly based on Raspberry Pi and the Linux operating system can benefit from the development of new analysis tools, intelligent data mining techniques, and distributed file systems.

According to [7], a pedestrian flow model can be built from either macroscopic or microscopic viewpoint. While the macroscopic model focuses on an overview of movement patterns, the microscopic model tries to find the rules of pedestrian movement. One of the most common models is the grid-based approach, in which the target space is represented by a series of discrete spatial areas having their own movement rules [8]. In the development of a movement model, nonlinear behaviour can be efficiently captured by ANN (Artificial Neural Network). Such models are very useful in many areas. To begin with, they can help to design public transportation facilities and monitor their performance for the management purpose. Moreover, airport systems can measure passenger-sidetransit time through various vehicular and pedestrian services [3]. Among those applications, we focus on the

VOL. 12, NO. 3, FEBRUARY 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

analysis of the commercial potential of a target area, as it is desirable to open a shop in the place where many people gather.

This paper is organized as follows: After introducing what we will challenge in Section 1, Section 2 describes the main idea of the proposed low-cost investigator. Then, Section 3 measures the performance of the proposed pedestrian volume detector mainly in terms of detection accuracy. Finally, Section 4 summarizes and concludes this paper with a brief explanation of future work.

2. MAIN IDEA

Figure-1 depicts the configuration of our pedestrian volume investigator. Currently, it consists of 3 sensor nodes located at predefined positions. Each of them is stuffed with a WLAN interface and an ultrasonic distance sensor, running the Raspbian operating system [9]. A coordinator running on a Windows PC controls the operation of the overall investigator and each unit communicates via the commonly available TCP/IP socket. According to the master-slave style interaction, sensor nodes infinitely wait for a command to arrive from the coordinator. When it arrives, each sensor application captures the distance to the nearest object through the *wiringPi* library and sends the measured distance back to the coordinator. After collecting all reports from sensor nodes, the coordinator either creates a learning record or estimates the number of objects according to its operation mode [10].

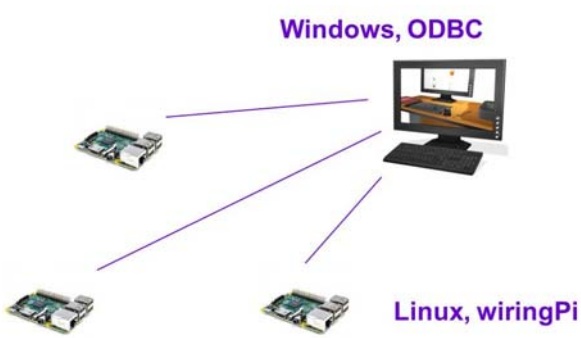

Figure-1. Sensor node configuration**.**

With this configuration, the coordinator can collect sensor readings any time it wants. Just like video-based methods, our scheme periodically takes a snapshot for the target area. As mentioned previously, a snapshot is a set of distance sensor readings, namely, just a few floating-point numbers not a large-size video image. This is the main advantage of our design and it is necessary to find the mapping function from a snapshot to the number of objects. Our scheme exploits machine learning techniques, such as ANN and it is the core of this paper. For machine learning, learning patterns are prerequisites and a human operator takes the initiative of the whole process in acquiring a set of learning patterns. The operator triggers the acquisition transaction on an as-needed basis, by making the coordinator issue an initiation command. In response to this order, the coordinator sends to all sensor nodes, receives replies from all recipients, and associates the number of objects he or she observes in the target area.

Figure-2 and Figure-3 show an example on what records our system collects. In Figure 2, 4 cases are displayed. The location of each sensor is fixed, and the number of people and thus the sensor values are different in each case. In the upper left case, 3 sensors report their sensor readings as 2.5*m*, 2.0 *m*, and 2.7 *m*, respectively, while the operator observes 2 persons. The operator can key in his or her observations or give a voice input. Likewise, 4 records are created, as shown in the bottom of each subfigures. It must be mentioned that learning patterns much be re-obtained if a sensor is relocated. This overhead is the cost for the enhanced flexibility and reduced expense.

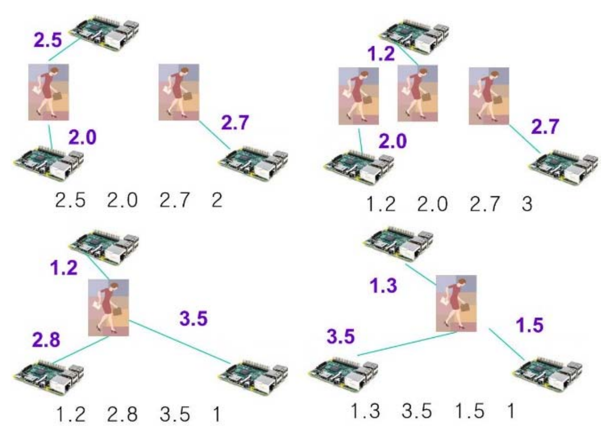

Figure-2. Learning pattern acquisition.

After accumulating a sufficient number of learning patterns, we can feed them to an ANN builder as shown in Figure-3. Specifically, this paper employs FANN (Fast ANN) open software [11], which provides an easy-to-link C language library functions and clear data file definition. It is also true that recent statistics package such as R has abundant set of machine learning functions and we can employ them [12]. However, the FANN library converges much efficiently and allows us to consistently develop the whole software components with the C language, especially, taking advantage of various file system calls.

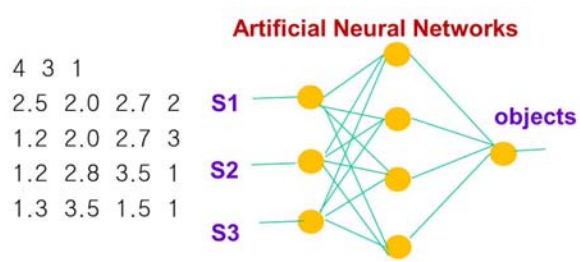

Figure-3. Main idea.

ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

In addition, in the Figure, 4, 3, and 1 at the first line mean the total number of learning records, that of input nodes and that of output nodes, respectively, as defined in the FANN file format. The number of input nodes coincides with the number of sensor nodes, while the number of output node is one and it denotes the number of objects. Next, the number of hidden nodes is decided by trial-and-error, and set to 20, as this selection minimizes the root square error. 4 snapshots taken in Figure 2 are converted into learning patterns as shown in the left side of Figure-3. Each line contains 4 numbers. According to the FANN specification, the output value must be mapped to the range from 0 to 1.0. Hence, the maximum of the record set must be converted to 1.0 in the learning phase.

Upon the completion of the learning phase, an ANN model is built and the investigator unit can now run autonomously. Its coordinator periodically collects a snapshot, inputs those sensor readings to the ANN model, and finally gets the estimated number of objects. Once a model is available, the computation time will be very short and the result is just an integer. Hence, we can send the result to a server in any location without a long delay. It is true that this system sometimes cannot exactly capture all objects, especially when they overlap in the sensing direction. However, it can sufficiently grab the overall flow of objects. This time series will be given to an analysis model to find the dynamics of the pedestrian volume change.

3. EXPERIMENT RESULTS

In our assumption, a sensor node needs to be able to measure the distance to the closest object located in its front side direction. However, current ultrasonic distance sensors can catch the object within a narrow angle, as the ultrasonic wave propagates and echoes back straight. Hence, as a pilot prototype version for feasibility test, this section measures the accuracy of our pedestrian volume investigator via computer-based simulation. The sensors, numbered as S_1, S_2 and S_3 are located at each vertex of an equilateral triangle as shown in Figure 4. If the width of the target road is *l*, the length of a triangle edge will be $l\sqrt{3}$. In the subsequent experiments, we take *l* as 5 *m* for simplicity, but it can be altered. This figure describes also coordinate mapping of the target area. Namely, S_1, S_2 and S_3 are mapped to $(1+5/2\sqrt{3}, 5)$, $(1, 0)$, and $(1+5\sqrt{3}, 0)$, respectively, with 1 *m* margin at both ends. With this coordinate assignment, the simulation process can make objects virtually distribute with simple arithmetic operations. Our experiment further assumes that the target area is flat.

After all, the detection area is a virtual 5 *m* × $(2+5\sqrt{3})$ *m* square. For simulation, 100 distributions are generated within the target space for the number of objects from 1 to 10, respectively, making the total number of learning patterns 1,000. For a distribution associated with a specific number of objects, three sensor node agents calculate the distance to each object and select the smallest one with the straightforward mathematical calculation, respectively. The operation of a sensor node is simulated in this way and we obtain 1,000 learning patterns. Then, the simulator generates another 200 distributions, 20 each for the respective cases of 1 to 10 objects. The experiment evaluates the performance of the proposed pedestrian volume investigator in terms of exact estimation, average error, and maximum error.

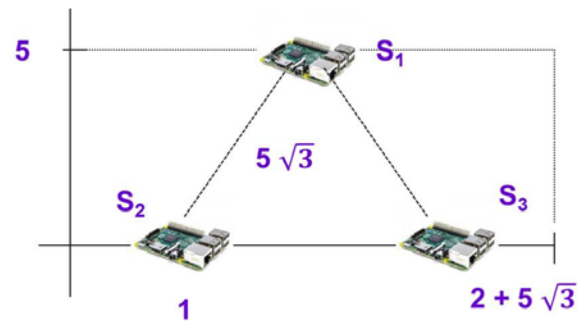

Figure-4. Simulation environment.

To begin with, Figure-5 plots the fitting error according to the number of hidden nodes. The experiment changes the parameter from 5 to 30. The FANN learning process displays the fitting error in terms of the root mean square when it completes thesufficiently large number of iterations [13]. As shown in the figure, the fitting error is not so much affected by the number of hidden nodes, the maximum gap being just 1.4 %. The fitting error becomes the smallest when we take 20 nodes and this model is used for the subsequent estimation.

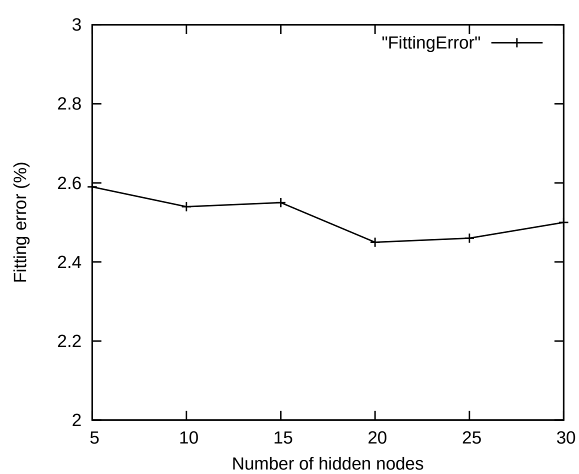

Figure-5. Fitting error vs. number of hidden nodes.

Figure-6 plots the success ratio that the investigator exactly estimates the number of objects. In case of 1 or 2 objects, the investigator estimates without any error on 11 out of 20 distributions. In this case, the estimation is different from the actual value mostly by just one. However, the ratio gets reduced and no correct estimation is obtained on 10 objects. It is obvious that with more objects, they are more likely to overlap. More sensor deployment would improve the possibility of detecting interim objects. In addition, more exact estimates are observed in the case of 6 objects than in the case of 5 objects. This situation is a little bit anomalous.

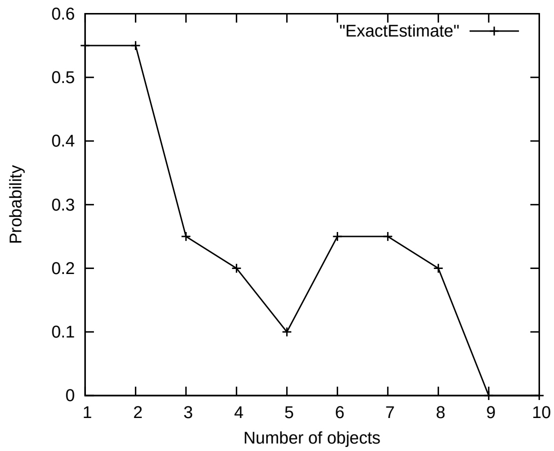

Figure-6. Probability of exact estimation.

Figure-7 plots the average error size, which is counted by the number of objects. As shown in the curve, the error is not so large when the number of objects is 8 or less. However, in case of 9 or 10 objects, the error size grows as hidden objects are highly likely not to be identified by our scheme. We can see that the error size hardly exceeds 1 person when there are fewer than or equal to 8 objects.

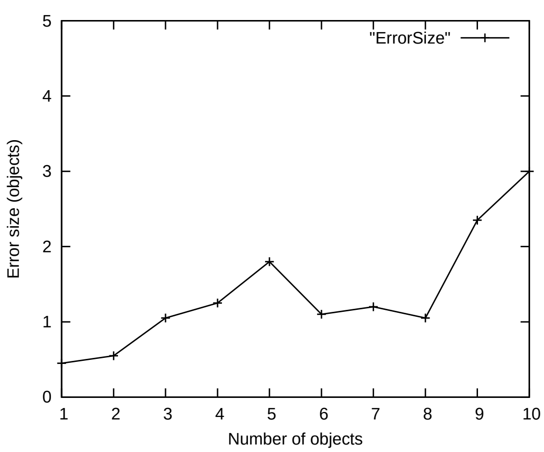

Figure-7. Error size.

Finally, Figure-8 plots the maximum error when the number of objects ranges from 1 to 10. In cases of 8 or fewer objects, the difference between the maximum and average error is large, indicating that the maximum error (or errors) contributes to the increase in the average error size. In most cases, our scheme can reasonably detect the number of objects in the target space. After all, our scheme, built upon low-cost sensor nodes and network connectivity easily available these days, shows a meaningful performance until the number of objects is less than or equal to 8 in our simulation.

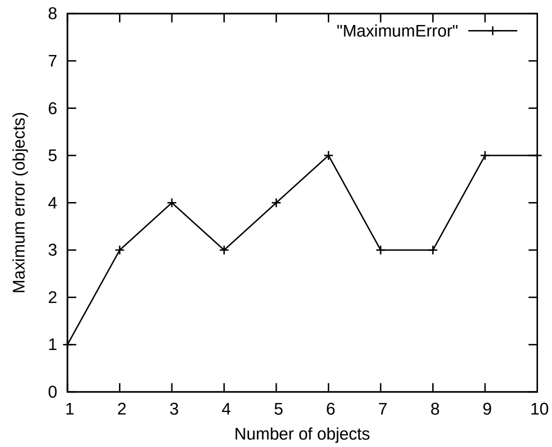

Figure-8. Maximum error.

4. CONCLUSIONS

Raspberry Pi is ceaselessly extending its application area with the development of new sensors, the integration of computational intelligence, and the availability communication networks. In this paper, we have designed a low-cost volume investigator, exploiting Raspberry Pi sensor nodes and machine learning technologies, specifically, an artificial neural network. Consisting of 3 sensor nodes and a coordinator, this system first acquires sample patterns by which we make an artificial neural network learn. A learning pattern is made up of 3 sensor readings, namely, the distance from a sensor to the closest object in the target space, and the number of objects in the target space. Then, the coordinator periodically and automatically collects sensor readings from 3 nodes while the number of objects is extracted by injecting them to the neural network. The analysis module will find the dynamics of such series with sufficient computing power. The simulation study has discovered that the error size hardly exceeds 1 object when the number of objects is 8 or less, indicating that it is possible to build a low-cost and less constrained pedestrian volume investigator.

The accuracy can be continuously improved by incrementally upgrading a constituent module with a better one, fully benefiting from the open hardware and software architecture. Hence, as future work, we will survey the performance of available building blocks and we are also planning to test diverse sensor distributions. It is promising to combine another machine learning techniques including fuzzy logic and different neural network configurations [14]. In addition, our final target is detecting the flow of electric vehicles on a road segment. Hence, we will search for sensors suitable for this purpose and look into the problem in handling such fast-moving objects. In case of electric vehicles, the spatial effect must be considered simultaneously [15].

ACKNOWLEDGEMENTS

Prof.Cheol Min Kim is the corresponding author.

This research was supported by the 2016 scientific promotion program funded by Jeju National University.

REFERENCES

[1] Nguyen H.-Q., Loan T., Mao B., Huh E.-N. 2015. Low cost real-time system monitoring using Raspberry Pi. International Conference on Ubiquitous and Future Networks. pp. 857-859.

[2] Ramchrun S., Vytelingum R., Rogers A., Jennings N. 2012. Putting the 'smarts' into the smart grid: A grand challenge for artificial intelligence. Communication of the ACM. 55(4):89-97.

[3] Hainen A., Remais S., BullockD.2001. Collection and analysis of multi-modal airport land side probe data from Bluetooth enabled mobile devices. IEEE Annual Conference on Intelligent Transportation Systems. pp. 1304–1309.

[4] Ding Z., Yang B., Chi Y., Guo L.2016. Enabling smart transportation systems: A parallel spatio-temporal database approach. IEEE Transaction on Computers. 65(5):1377-1391.

[5] Sager H., Hoff W. 2014. Pedestrian detection in low resolution videos. IEEE Winter Conference on Applications of Computer Vision. pp. 668-673.

[6] Wagner R., Gabb M., Forster J., Schweiger R., Rothermel A. 2012. Improving detector performance by learning from compressed samples. IEEE International Conference on Consumer Electronics.

[7] Ma Y., Lee Y., Yuen R. 2016. An artificial intelligence-based approach for simulating pedestrian movement. IEEE Transactions on Intelligent Transportation Systems.

[8] Burstdde C., Klauck K., Schadschneider A., Zittartz J. 2001. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. Physica A. 295(3/4):507-525.

[9] Cocorullo G., Corsonello P., Frustaci F., Guachi L., PerriS. 2015. Embedded surveillance system using background subtraction and Raspberry Pi. AEIT International Annual Conference.

[10] Lee J., Park G., Cho Y., Kim S., Jung J. 2015. Spatio-temporal analysis of state-of-charge streams forelectric vehicles. 4th ACM/IEEE International Conference on Information Processing in Sensor Networks. pp. 368-369.

[11] Nissen S. 2005. Neural Network Made Simple. http://fann.sourceforge.net/fann_en.pdf Software 2.0.

[12] Brunsdon C., Comber L. 2015. An Introduction to Rfor Spatial Analysis & Mapping. SAGE Publication Ltd.

[13] Lee J., Park G., Kim Y., Kim E., Lee I. 2012. Wind speed modelling based on artificial neural networks for Jeju area. International Journal of Control and Automation. 5(2):73-80.

[14] Lee J., Park G. 2016. Vote-based wind speed forecast scheme built upon different artificial neural network models. International Journal of Control and Automation. 9(5): 99-110.

[15] Park C., Lee J., Kim S., Hyun J., Park G. 2013. Spatio-temporal correlation analysis of wind speed and sunlight for prediction power generation in Jeju. International Journal of Multimedia and Ubiquitous Engineering. 8(4):273-282.