Sensors and Materials, Vol. 32, No. 1 (2020) 1–12
MYU Tokyo

Invited

S & M 2083

NeuralIO: Indoor–Outdoor Detection via Multimodal Sensor Data Fusion on Smartphones

Long Wang,* Lennard Sommer, Yexu Zhou, Yiran Huang, Jingsi Wang, Till Riedel, and Michael Beigl

TECO, Karlsruhe Institute of Technology,
Vincenz-Prießnitz-Straße 1, 76131 Karlsruhe, Germany

(Received August 31, 2019; accepted November 5, 2019)

Keywords: indoor–outdoor detection, multimodal data fusion, neural network model

The indoor–outdoor (IO) status of mobile devices is fundamental information for various smart city applications. In this paper, we present NeuralIO, a neural-network-based method for dealing with the IO detection problem for smartphones. Multimodal data from various sensors on a smartphone are fused through neural network models to determine the IO status. A data set containing more than one million labeled samples is then constructed. We test the performance of an early fusion scheme in various settings. NeuralIO achieves an accuracy above 98% in 10-fold cross-validation and an accuracy above 90% in a real-world test.

1. Introduction

The past decade has witnessed the flourishing of the Internet of Things (IoT) and its applications in urban spaces. The widespread deployment of IoT devices and the rise of smart cities are giving birth to an increasing number of smart applications.(1–4) Context status is critical and fundamental information for ubiquitous computing systems and context-aware IoT applications.(5,6) "Context" consists of a wide range of aspects such as location, time, surrounding environment, and so on. The rapid growth of smartphones is driving the increasing interest in context-aware applications.(7–9)

One of the most fundamental items of contextual information is whether the device is in an indoor or outdoor environment, because it makes a significant difference if the user is standing in front of a shopping mall or inside a shopping mall. Furthermore, the availability and capabilities of different technologies vary considerably between these two environments. The knowledge about the indoor–outdoor (IO) status enables the use of appropriate technologies, which leads to a better user experience. For instance, a device can trigger a reminder, change the working mode, and switch between GPS-based navigation and indoor navigation schemes when the user enters or leaves an indoor environment. Furthermore, the device can save energy by turning off the GPS module in indoor environments such as a metro station. Existing IO detection approaches commonly use a GPS signal,(10–13) a wireless signal(16,23,26,28) and other sensor data(15,17,20,21,27) to determine the IO status. Owing to the rich characteristics

*Corresponding author: e-mail: wanglong@teco.edu

https://doi.org/10.18494/SAM.2020.2586

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

of natural phenomena, it is rare that a single modality provides comprehensive knowledge of the phenomenon of interest.(18) The increasing availability of multiple sensing modalities on smartphones offers us more freedom to recognize the context. The capability of neural network models has been proven superior in solving increasingly complex machine learning problems that often involve multiple data modalities.(22)

In this paper, we propose NeuralIO to detect the IO status of smartphones through multimodal sensor data fusion using neural network models. We create a data set containing more than one million labeled samples involving nine users. Nine different sensing modalities, which are acceleration, GPS, light, magnetic field, proximity, cellular signal strength, sound level, temperature, and WiFi, are covered in the data set. We test the performance of an early fusion scheme in various settings.

To summarize, the contributions of this study are as follows.

1. We apply neural network models to the IO detection problem and perform a comprehensive analysis.
2. We implement an Android app for data collection and conduct experiments to collect data samples in various real daily scenarios. A data set containing more than one million labeled data samples is constructed.
3. We evaluate the performance of an early fusion scheme based on the data set through cross-validation and a real-world test. An accuracy above 98% is achieved in across validation and an accuracy above 90% is achieved in the real-world test.

The rest of the paper is organized as follows. Section 2 presents related works. Different fusion schemes are introduced in Sect. 3. The experiment and data collection are described in Sect. 4 and evaluation results are presented in Sect. 5. We conclude our work in Sect. 6.

2. Related Works

2.1 GPS

GPS signals are highly dependent on the line-of-sight (LOS) paths between the device and GPS satellites. It is well known that GPS signals are poor in indoor environments as the LOS paths of GPS signals are blocked. On the other hand, the LOS paths are not blocked in most outdoor scenarios. On the basis of these facts, the localization accuracy of GPS or the availability of GPS signals has been exploited to determine whether a device is in an indoor or outdoor environment.(10–13)

Despite the intuitive nature and easy implementation of GPS-based methods, they suffer from several disadvantages. Radu *et al.* identified the GPS chipset as the sensor with the highest power consumption among the evaluated sensors.(21) The battery capacity is still limited in state-of-the-art mobile phones and most users dislike applications that drain the battery. Secondly, the intuition behind these methods is not always reliable. For instance, GPS signals are reasonably strong if a device is in an indoor environment with large windows. In contrast, GPS signals can be blocked by surrounding mountains if a device is in a valley. Under these circumstances, GPS-based methods may give misleading results. A third disadvantage is that it normally takes around one minute to launch the GPS module, making GPS-based methods unsuitable for real-time applications.

2.2 Wireless signals

Shtar *et al.*(23) presented a method of continuous IO environment detection on mobile devices based solely on WiFi fingerprints assumed no prior knowledge of the environment. The model trained with the data collected for only a few hours on a single device was applicable to unknown locations and new devices. WifiBoost(16) used a machine learning meta-algorithm that combined an adequate ensemble of simple classifiers (so-called weak learners) to improve the overall performance. An average error rate of around 2.5% was achieved in the evaluation. However, a classifier should be created for each building and the surrounding area through measurements and labeling of each measurement point, especially in those cases where there was no previous fingerprinting database. Building such a database is not a trivial task.

Wang *et al.*(26) applied a machine learning algorithm to classify the signal strengths of neighboring cellular base stations in different environments and identified the current context by signal pattern recognition. An accuracy of 100% was reported for the identification of open outdoors, semi-outdoors, light indoors, and deep indoors.

In Ref. 28, low-power iBeacon technology was leveraged to develop an accurate, fast-response and energy-efficient scheme for IO detection. The transitions between outdoors and indoors were detected by comparing the received signal strengths of two predeployed Bluetooth beacons on two sides of each entrance.

2.3 Multiple sensors

Since a single sensor might not be able to tackle all application scenarios, data from multiple sensors such as accelerometers, proximity and light sensors, wireless receivers, and magnetometers were exploited for IO detection.(15,17,20,21,27) IODetector(27) combined data from three lightweight sensors (light, cell tower signal strength, and magnetic sensors) to develop an extensible IO detection framework that did not require a training phase. Although acceptable error rates were achieved, Radu *et al.*(21) criticized IODetector for its hard-coded thresholds that might not work with new devices and environments. As an alternative, they proposed a semi-supervised training method to improve IO detection accuracy across different devices and environments.

2.4 Other methods

In Ref. 19, the embedded digital camera on a mobile phone was utilized for IO detection. The developed gentle boosting classifier achieved error rates of 1.7% for indoor scenes and 10.8% for outdoor scenes. In addition, a feed forward neural network was trained with the gist feature of images to address the IO detection problem.(25) These methods help to generate semantic IO labels for images but do not work for tracking and in other real-time application cases.

Sung *et al.*(24) developed a sound-based IO detection method using a chirp signal. A simple classifier was developed with a static threshold. However, this work was rather simple and straightforward, and no comprehensive analysis was performed. Wang *et al.* conducted a comprehensive study on an audio-based IO detection method. The method was evaluated in various scenarios with different probing signals (MLS and chirp), noise levels, and device types.

3. Fusion Scheme

Neural networks offer the flexibility of implementing multimodal sensor fusion either as early, late or intermediate fusion.(22)

As shown in Fig. 1(a), the early fusion scheme data from multiple sources are integrated into a single feature vector to serve as the input of one machine learning model. In contrast, the late fusion scheme aggregates decisions from multiple models that are trained separately on their own modality as shown in Fig. 1(b). This fusion architecture is often favored because errors from multiple classifiers tend to be uncorrelated and the method is feature-independent.(22) For traditional machine learning methods, it is typically necessary to manually extract features from each modality, which is not only time-consuming but also challenging. Neural networks are known for being able to learn features automatically. In this paper, we use the feedforward neural network (FNN) model to conduct early fusion for the IO detection problem.

4. Experiment and Data Collection

4.1 App design and implementation

We have developed an Android app for data collection. The Android app is implemented with Android Studio. The target version of the application is 27 with a minimum version of 19. This covers the Android smartphones of all participants in this study. The app needs to access multiple sensors on the smartphone and save the sensor readings to a database. The collected data comprise the battery temperature, luminance, magnetic flux density, proximity, cellular signal strength, and cellular network bit error rate, an abstract level for the overall signal strength ranging from one to four, the number of WiFi networks around the user, the highest signal strength of the WiFi networks around the user, the number of GPS satellites, the GPS

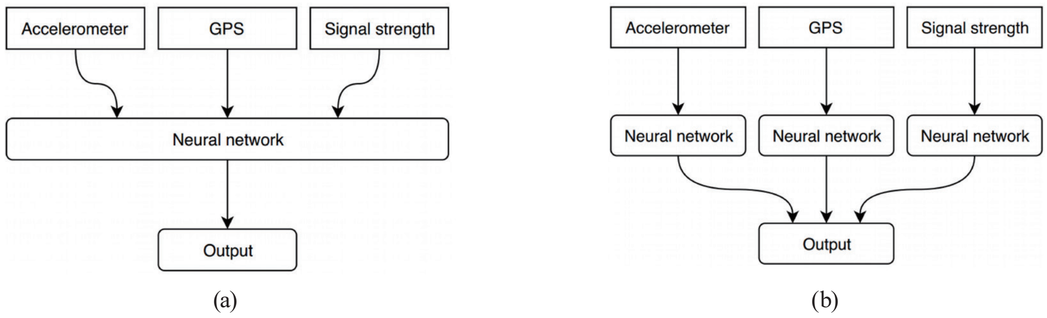


Fig. 1. Schema of early fusion and late fusion schemes based on neural networks. (a) Early and (b) late fusion.

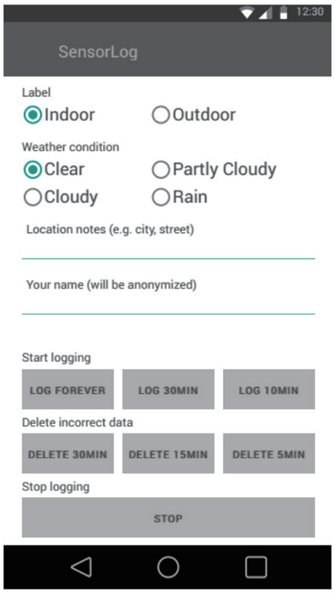
accuracy in meters, the GPS signal-to-noise ratio, and the ambient noise level. Additionally, some anonymous information about the device is also recorded to distinguish different data traces.

It is crucial for the user interface of the smartphone application to ensure that the user can modify data labels or remove the collected data since they may make mistakes when they log data. Also, the process of starting and stopping the data collection should be fast and simple for the user.

Figure 2 shows a screenshot of the developed app. The user specifies whether he/she is indoor or outdoor and inputs the current weather condition. Then, he/she has the option to provide notes on the location and his/her name. The user starts the logging period for either 10 min, 30 min, or an unlimited amount of time. If, for example, the user walks indoors while logging data labeled outdoors, he/she has the option to invalidate the last 5, 15, or 30 min of the collected data. The user can stop the logging process at any time. The application collects the specified information every 200 ms as one JavaScript Object Notation (JSON) object. The data is then sent to an instance of the Firebase Realtime Database (DB).[14] This ensures that every user directly writes to the same database and no data is saved locally on user's device. From there, the data can be downloaded for further processing. This process is displayed in Fig. 3.

4.2 Data collection

The smartphone application was handed out to multiple participants for data collection. The users were instructed about the application and how to use it. The data collection ran for four weeks and users were free to choose the time and environment for data logging. Nine users



12:30

SensorLog

Label
● Indoor ○ Outdoor

Weather condition
● Clear ○ Partly Cloudy
○ Cloudy ○ Rain

Location notes (e.g. city, street)

Your name (will be anonymized)

Start logging
LOG FOREVER | LOG 30MIN | LOG 10MIN

Delete incorrect data
DELETE 30MIN | DELETE 15MIN | DELETE 5MIN

Stop logging
STOP

Fig. 2. (Color online) Screenshot of the developed Android app.

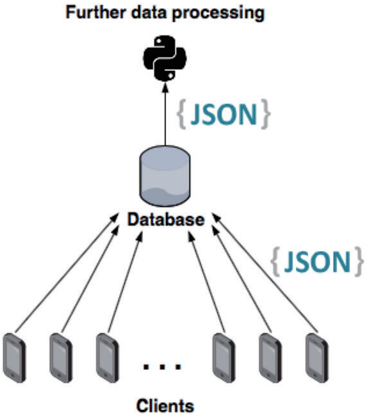


Fig. 3. (Color online) Data flow diagram.

participated in the data collection campaign and various models of smartphones were used for data collection. The users collected the data in their daily life in both urban and rural areas. This ensures the diversity of the data set. Figure 4 shows typical data logging scenarios.

The resulting data set contain 1038678 samples, which correspond to 58 hours of data. 99.49% of the data are collected by four users. The remaining 0.51% of the data are collected by five other users. Overall, the distribution of indoor to outdoor samples ranges from 57.61 to 42.39%. Table 1 shows an example of collected data instance.

(a) (b)

Fig. 4. (Color online) Data logging process. The picture on the left shows a user configuring the data logging session and the picture on the right depicts data logging inside a pocket.

Table 1
Example of collected data instance.

```
" sensor_data ":
            " accelerometer_x ": -3.39139986038208,
            " accelerometer_y ": 0.39469999074935913,
            " accelerometer_z ": 1.4085999727249146,
            " gps_accuracy ": 8,
            " gps_lat ": 49.00587403419052,
            " gps_long ": 8.418945486346862,
            " gps_satellites ": 17,
            " gps_snr_list ": {"0": 22,   "1": 41,  "2": 18,         "3": 34,    "4": "23",  ...},
            " is_indoor ": false ,
            " light ": 1295,
            " magnetic_x ": 8.9375,
            " magnetic_y ": -10,
            " magnetic_z ": -43.4375,
            " proximity ": 5,
            " signal_bit_error_rate ": -2145384446,
            " signal_level ": 4,
            " signal_strength ": -113,
            " sound_level ": 0,
            " temperature ": 26,
            " time ": " Mon Oct 29 08:16:21 GMT +01:00 2018",
            " wifi_list ": {"0": -78,   "1": -81,   "2": -71,  "3": -73,  "4": "-75",  … }
" user_data ":
            " delete_last_minutes ": 0,
            " location_notes ": " Home to university ",
            " phone_model ": " HUAWEI HUAWEI GRA -L09",
            " username ": " **** ",
            " weather ": " Cloudy "
```

4.3 Overview of data set

The distribution before cleaning for different smartphones is illustrated in Fig. 5. Different smartphones also represent different users.

By removing the samples invalidated by the users themselves, 1019091 samples are left; this number of samples is equivalent to about 56.5 h of data. However, not every collected sample is completed for various reasons. Figure 6 illustrates how many samples are missing for each sensing modality. After removing the incomplete samples, the resulting data set includes 623320 samples, which correspond to around 34.5 h of data. The balance between indoor and outdoor samples is now 43.98 to 56.02%. Figures 7–9 show the data distribution regarding

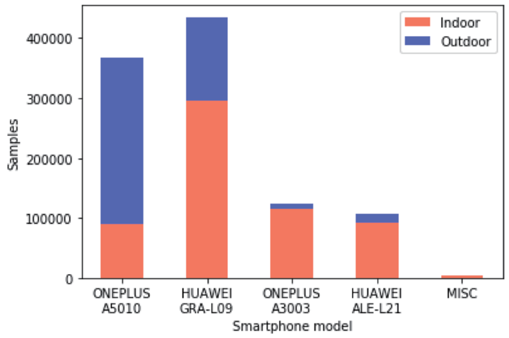


Fig. 5. (Color online) Distribution before cleaning.

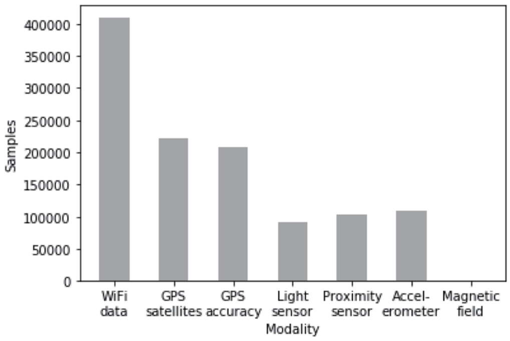


Fig. 6. Number of samples with missing values.

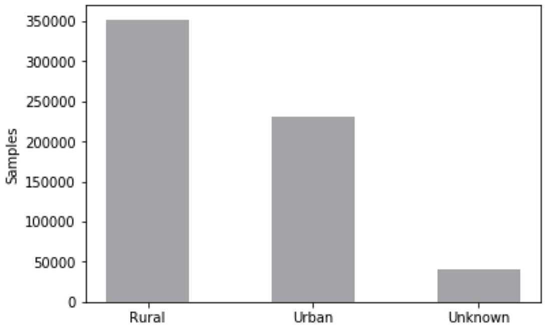


Fig. 7. Rural/urban environment.

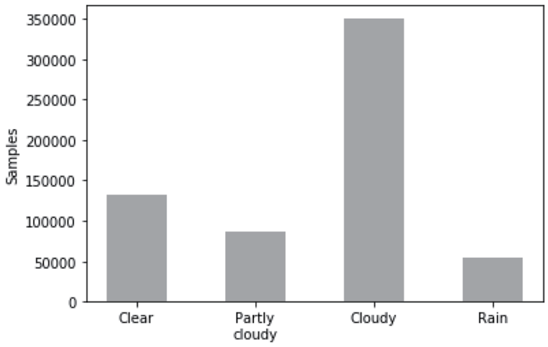


Fig. 8. Weather conditions.

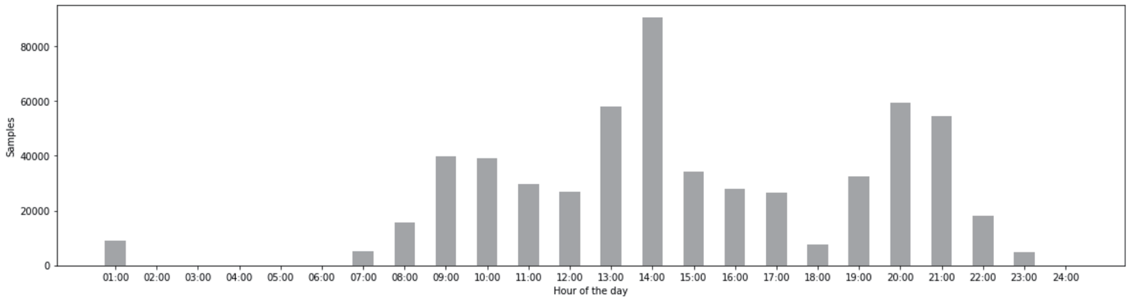


Fig. 9. Time distribution.

rural/urban environments, weather conditions, and time, respectively. We can see that the data set contains various data samples with a balanced distribution.

5. Evaluation

5.1 Cross-validation

We used 10-fold cross-validation to evaluate the performance of the constructed model with various numbers of hidden units and layers. Finally, we obtained good balance between performance and model complexity by using the architecture in Fig. 10. The input layer with 24 input nodes is omitted owing to the limited space. There are four hidden layers with 10, 5, 4, 3 hidden units with the Relu function as the activation function. The output unit uses the sigmoid function as the activation function. As shown in Table 2, the results of 10-fold cross-validation demonstrate that the model performs very well in nine out of 10 folds; in the fifth fold, the model only achieves an accuracy of 0.73. This is probably due to the loss function becoming trapped at a local minimum.

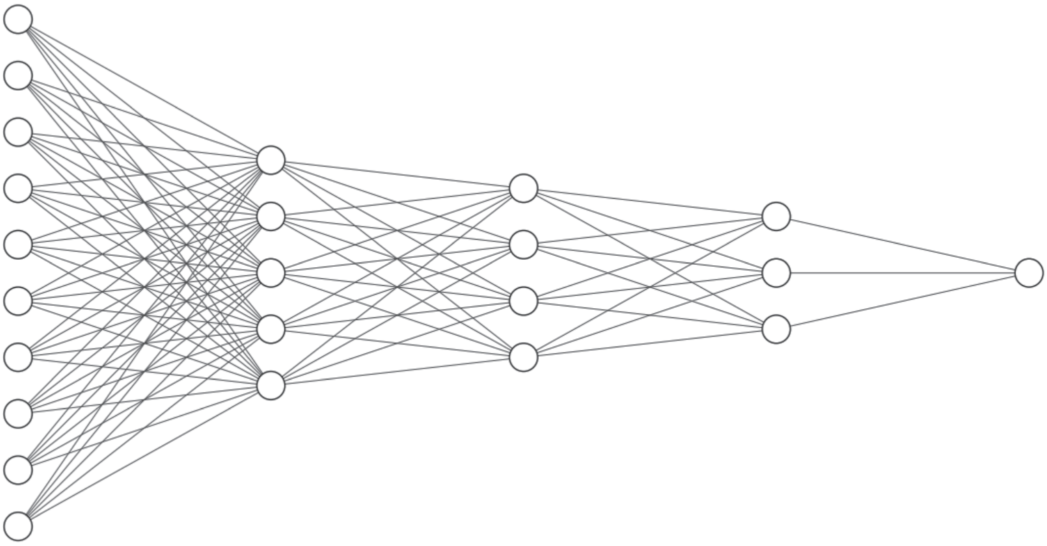


Fig 10. Feedforward neural network (FNN) model architecture with four hidden layers and one output layer. Note that the input layer (with 24 input nodes) is not displayed owing to the limited space. The activation function of all hidden units is the Relu function. The activation function for the output node is the sigmoid function.

Table 2
Results of 10-fold cross-validation. Precision and recall are for the outdoor label.

	1	2	3	4	5	6	7	8	9	10
Accuracy	0.98	0.99	0.99	0.99	0.73	0.98	0.99	0.99	0.99	0.98
Precision	0.98	0.99	0.99	0.99	0.73	0.99	0.99	0.99	0.99	0.99
Recall	1	0.99	0.99	0.99	1	0.99	0.99	0.99	0.99	0.98

5.2 Real-world test

To verify the performance of the model in the real world, we tested the trained FNN model on the real-world data set recorded around two months later than the training data set. During the collection of the data set, the user walked through the city as depicted in Fig. 11. The trace covers indoor environments, such as campus buildings and shopping malls, and outdoor environments, such as streets.

The confusion matrix is shown in Fig. 12. Generally, the model performed well in the real-world test with an overall accuracy of 91%. Specifically, the model can recognize indoor cases with a precision of 96% with 4% falsely classified as outdoors. The model achieves a precision of 88% in outdoor cases with 12% of all outdoor cases falsely classified as indoors. The model shows good generalizability on the new data set. To investigate the cause of the misclassification of the model, we plot the labels of all data entries against the index in Fig. 13. As shown in Fig. 13, there are some isolated misclassifications in both indoor and outdoor cases. Making the common-sense assumption that it is very rare for people to switch between indoor and outdoor states in a short time period (for instance, 2 s), we can use a majority voting strategy with a sliding window to filter out the isolated misclassification cases. The basic idea is that the IO state is not only determined by the input data, but also depends on the previous predicted labels in the sliding window. As shown in Fig. 14, there are fewer isolated misclassification cases after applying the majority voting strategy with a sliding window of 10. The confusion matrix in Fig. 14 also shows an increase in precision in both indoor and outdoor cases.

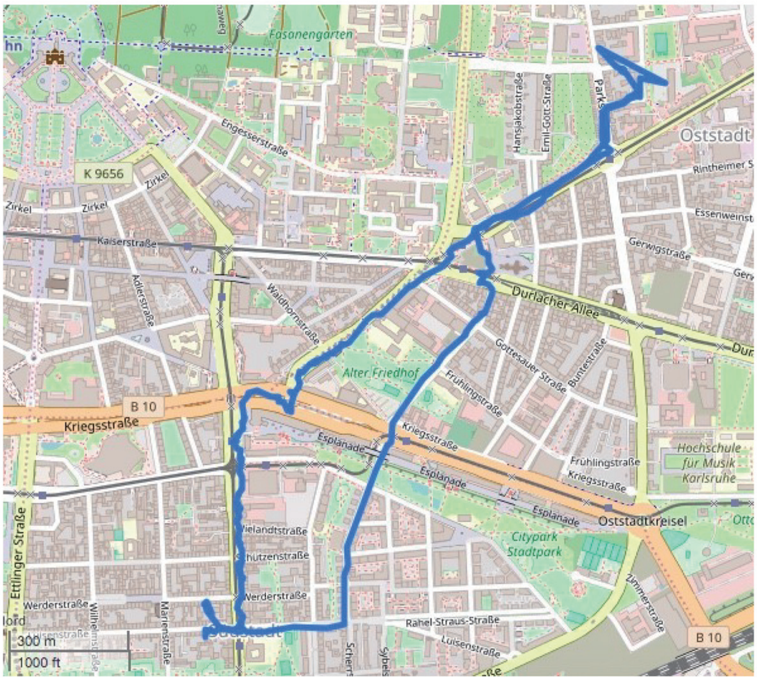


Fig. 11. (Color online) Real-world test path.

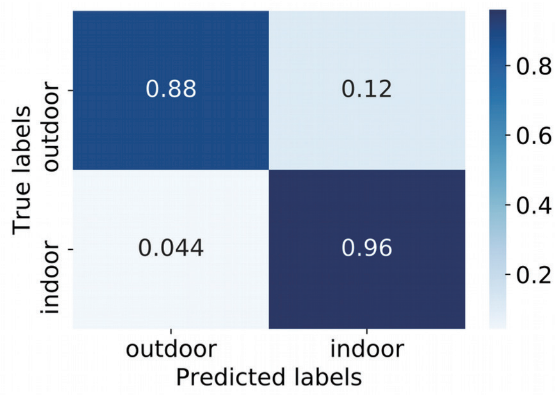


Fig. 12. (Color online) Confusion matrix.

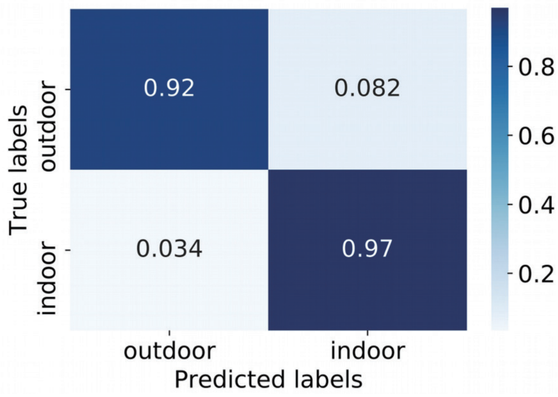


Fig. 13. (Color online) Confusion matrix for model with majority voting strategy.

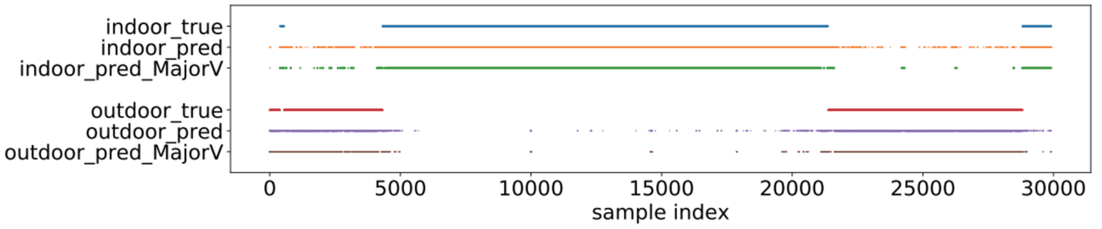


Fig. 14. (Color online) Ground truth and prediction results without/with majority voting strategy.

6. Conclusions

We developed NeuralIO, a neural-network-based multimodal fusion method for the IO detection problem on smartphones. A data set containing more than 1 million data samples was constructed. Nine different sensing modalities were covered in the data set. We built a feed forward neural network model for the early fusion of all available raw data. Cross-validation and a real-world test have shown the feasibility of our developed method for indoor-outdoor detection and generalizability on a new data set.

References

1 O. Amft and K. Van Laerhoven: IEEE Pervasive Comput. **16** (2017) 80. https://doi.org/10.1109/MPRV.2017.3971124

2 Y. Hen, T. Yonezawa, J. Nakazawa, and H. Tokuda: Proc. 2017 10th Int. Conf. Mobile Computing and Ubiquitous Network (ICMU) (IEEE, 2017) 1. https://doi.org/10.23919/ICMU.2017.8330071

3 T. Franke, P. Lukowicz, and U. Blanke: J. Int. Services Appl. **6** (2015) 27. https://doi.org/10.1186/s13174-015-0040-6

4 A. E. Khaled, A. Helal, W. Lindquist, and C. Lee: IEEE Access **6** (2018) 24048. https://doi.org/10.1109/ACCESS.2018.2825295

5 Y. Ishida, N. Thepvilojanapong, and Y. Tobe: Proc. 10th Int. Conf. Mobile Data Management: Systems, Services and Middleware (IEEE, 2009) 508. https://doi.org/10.1109/MDM.2009.89

6 Y. Uehara, M. Mori, N. Ishii, Y. Tobe, and Y. Shiraishi: Proc. 4th Int. Conf. Embedded Networked Sensor Systems (ACM, 2006) 379. https://doi.org/10.1145/1182807.1182864

7 H. Maeda, Y. Sekimoto, and T. Seto: Proc. Second Int. Conf. IoT in Urban Space (ACM, 2016) 111. https://doi.org/10.1145/2962735.2962738

8 H. Maeda, Y. Sekimoto, and T. Seto: Proc. 5th ACM SIGSPATIAL Int. Workshop Mobile Geographic Information Systems (ACM, 2016) 37. https://doi.org/10.1145/3004725.3004729

9 M. Perttunen, O. Mazhelis, F. Cong, M. Kauppila, T. Leppänen, J. Kantola, J. Collin, S. Pirttikangas, J. Haverinen, T. Ristaniemi, and J. Riekki: Proc. Int. Conf. Ubiquitous Intelligence and Computing (Springer, 2011) 64. https://doi.org/10.1007/978-3-642-23641-9_8

10 K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan: Proc. 16th Annu. Int. Conf. Mobile Computing and Networking (ACM, 2010) 173. https://doi.org/10.1145/1859995.1860016

11 H. Cho, J. Song, H. Park, and C. Hwang: Proc. 11th Int. Symp. Location-based Services (2014).

12 M. Okamoto and C. Chen: Adjunct Proc. 2015 ACM Int. Joint Conf. Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM Int. Symp. Wearable Computers (ACM, 2015) 257. https://doi.org/10.1145/2800835.2800939

13 H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury: Proc. 10th Int. Conf. Mobile Systems, Applications, and Services (ACM, 2012) 197. https://doi.org/10.1145/2307636.2307655

14 Firebase Realtime Database Documentation: https://firebase.google.com/docs/database/ (accessed June 2019).

15 M. Ali, T. ElBatt, and M. Youssef: IEEE Sens. J. **18** (2018) 3684. https://doi.org/10.1109/JSEN.2018.2810193

16 O. Canovas, P. E. Lopez-de Teruel, and A. Ruiz: IEEE Sens. J. **17** (2017) 1443. https://doi.org/10.1109/JSEN.2016.2640358

17 M. Jia, Y. Yang, L. Kuang, W. Xu, T. Chu, and H. Song: Proc. Trustcom/BigDataSE/ISPA (IEEE, 2016) 1114. https://doi.org/10.1109/TrustCom.2016.0183

18 D. Lahat, T. Adali, and C. Jutten: Proc. IEEE **103** (2015) 1449. https://doi.org/10.1109/JPROC.2015.2460697

19 U. Lipowezky and I. Vol: Proc. 2010 IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition-Workshops (IEEE,2010) 31. https://doi.org/10.1109/CVPRW.2010.5543754

20 Y. Nakamura, M. Ono, M. Sekiya, K. Honda, and O. Takahashi: Proc. 2015 Int. Workshop Informatics (2015).

21 V. Radu, P. Katsikouli, R. Sarkar, and M. K. Marina: Proc. the 12th ACM Conf. Embedded Network Sensor Systems (ACM, 2014) 280. https://doi.org/10.1145/2668332.2668347

22 D. Ramachandram and G. W. Taylor: IEEE Signal Process. Mag. **34** (2017) 96. https://doi.org/10.1109/MSP.2017.2738401

23 G. Shtar, B. Shapira, and L. Rokach: Springer Wireless Netw. **25** (2018) 1341. https://doi.org/10.1007/s11276-018-1753-9

24 R. Sung, S. H. Jung, and D. Han: ICT Express **1** (2015) 106. https://doi.org/10.1016/j.icte.2016.02.001

25 W. Tahir, A. Majeed, and T. Rehman: Proc. 12th Int. Conf. High-capacity Optical Networks and Enabling/Emerging Technologies (HONET) (IEEE, 2015) 1. https://doi.org/10.1109/HONET.2015.7395428

26 W. Wang, Q. Chang, Q. Li, Z. Shi, and W. Chen: Sensors **16** (2016) 1563. https://doi.org/10.3390/s16101563

27 P. Zhou, Y. Zheng, Z. Li, M. Li, and G. Shen: Proc. 10th ACM Conf. Embedded Network Sensor Systems (ACM, 2012) 113. https://doi.org/10.1145/2426656.2426668

28 H. Zou, H. Jiang, Y. Luo, J. Zhu, X. Lu, and L. Xie: Sensors **16** (2016) 268. https://doi.org/10.3390/s16020268

About the Authors

Long Wang received his B.S. degree from Tsinghua University, China, in 2010 and his M.S. degree from PLA University of Science and Technology, China, in 2013. Since 2015, he has been a Ph.D. student and research assistant at the Karlsruhe Institute of Technology (KIT), Germany. His research interests are in human-computer interaction, context recognition, and data analytics. (wanglong@teco.edu)

Lennard Sommer received his B.S. degree from the Karlsruhe Institute of Technology (KIT), Germany, in 2019. Since 2019, he has been a master student at KIT. His research interests are in context recognition and machine learning. (sommer@teco.edu)

Yexu Zhou received his B.S. and M.S. degrees from the Karlsruhe Institute of Technology (KIT), Germany, in 2016 and 2018, respectively. Since 2018, he has been a scientific assistant at Telecooperation Office (TECO) Institute of KIT. His research interests are in automated machine learning and neural architecture searches. (zhou@teco.edu)

Yiran Huang received his B.S. degree from Zhengzhou University, China, in 2013 and his M.S. degrees from the Karlsruhe Institute of Technology(KIT), Germany, in 2018. Since 2018, he has been a scientific staff at KIT. His research interests are in reinforcement learning, AutoML, and data mining. (ihuangyiran@icloud.com)

Jingsi Wang received her B.S. degree from Tianjin University, China, in 2012 and her M.S. degree from the Karlsruhe Institute of Technology, Germany, in 2019. Her research interests are in image processing and deep learning. (jwang@teco.edu)

Till Riedel received his diploma in computer science from the University of Karlsruhe, Germany in 2005 and his Ph.D. degree from the Karlsruhe Institute of Technology (KIT), Germany, in 2012. Since 2010, he has been the research (co-)director at Telecooperation Office (TECO), Karlsruhe Institute of Technology. His research interests are in data analytics, context-sensitive systems, and pervasive computing. (Riedel@teco.edu)

Michael Beigl has been a professor at the Karlsruhe Institute of Technology (KIT, formerly University of Karlsruhe) since 2010 and holds the chair position for Pervasive Computing Systems. He has also been the head of the TECO research group since 2010, the spokesperson and co-founder of the Smart Data Innovation Lab (SDIL), and the co-director of the Smart Data Solution Center, Baden-Wuerttemberg (SDSC-BW). His research interests evolve around people and services at the center of communication and information technology in the field of ubiquitous/pervasive/mobile computing. In his industry projects, he is interested in the Internet of Things (IoT), Big Data, and Industry 4.0 applications. His current research focuses on the combination of human computer interaction, sensing systems and appliances, and sensor analytics using methods and systems based on artificial intelligence. (beigl@teco.edu)