Janine Beecken, Andreas Reinhardt: How Smooth is my Ride? Detecting Bikeway Conditions from Smartphone Video Streams. In: Proceedings of the 6th International Workshop on Crowd Assisted Sensing, Pervasive Systems and Communications (CASPer), pp. 65-70, March 2019.

How Smooth is my Ride? Detecting Bikeway Conditions from Smartphone Video Streams

Janine Beecken
Department of Informatics
Technische Universität Clausthal
Clausthal-Zellerfeld, Germany
janine.beecken@tu-clausthal.de

Andreas Reinhardt
Department of Informatics
Technische Universität Clausthal
Clausthal-Zellerfeld, Germany
reinhardt@ieee.org

***Abstract*—In many countries, bicycling has emerged as a viable alternative to motorized means of transport. Citizens rely on bicycles to commute to their workplaces, transport goods, and use them for sports and leisure activities. Available maps are, however, often scarce of information with relevance for cyclists. Besides the presence of tracks, their intersections, and approximations of their inclinations (through contour lines), little further annotations are available. In particular, the surface type of a track (e.g., asphalt, cobbled paving, or soil) is rarely provided, despite the fact that it determines how easily the track can be passed in diverse weather conditions. Cyclists will often only discover the exact track conditions by the time they pass it (or are unable to pass due to it being washed out or flooded by rain). In this work, we present SURF, a pervasive computing application which allows to detect a track's surface type using an opportunistic bicycle-centric sensing system. SURF relies on the processing of images (collected using a handlebar-mounted smartphone) by means of machine learning tools. We evaluate SURF using more than 67,000 training images collected during actual bicycle rides, and show how the system can determine five major surface types of bikeways at an accuracy of 99.51%.**

I. Introduction

Bicycles have established themselves as a flexible, sustainable, and pollution-free means of private transport. A general societal trend towards the increased use of bicycles for everyday activities can be observed in literature [1], [2]. Their adoption is furthermore accelerated by the construction of designated bikeways in and between cities [3]–[5], which provide safe and fast options for the daily commute. Routing cyclists along the network of cycleways and passable off-road tracks, however, is not a feature widely supported by navigation applications. Quite the contrary, many existing solutions[1] focus on motorized vehicles, for which they even provide real-time delay forecasts. Cyclists are, however, not necessarily impacted by the automobile traffic conditions on the roads they use. They also often have difference routing objectives, e.g., giving more weight to a smooth elevation profile or the absence of motorized traffic than to finding the shortest distance. Aforementioned vehicular routing applications are thus only partially usable for cyclist navigation, particularly when considered in conjunction with their limited extent of information of relevance for cyclists.

This restriction is overcome by applications specifically tailored to cyclist navigation[2], the majority of which use data from the community-driven OpenStreetMap (OSM) project [6]. This allows unpaved roads, hiking tracks, and even ferry connections to be considered in the routing process. OSM also offers the possibility to annotate the surface type of its contained tracks. This information can be used to determine routes which are both shorter and cater to the needs of cyclists better. A current limitation of OSM in this regard is, however, the incomplete set of surface annotations to provide this functionality. For example, the OpenStreetMap data for the German state of Lower Saxony (approx. 8 million inhabitants on an area of almost $50\,000\,\mathrm{km}^2$) contains more than one million "ways", yet less than $40\,\%$ of them are provided with surface annotations. We target to overcome this limitation by proposing SURF, a system to crowdsource the collection of track surface information with the help of computer vision and machine learning. SURF can differentiate between the five most prevalent surface types of tracks passable by bicycles, based on photographs from a front-facing camera. The key contributions of this paper can be summarized as follows:

- We present our concept for SURF, the data processing system used to categorize the surface of bicycle tracks based on photos taken from a handlebar-mounted camera.
- We introduce a data set containing $67\,000$ images from actual bicycle rides and discuss ways to augment the data in order to yield a larger number of input samples for SURF's machine learning component.
- We evaluate SURF using the aforementioned data set and measure its classification accuracy as well as discussing sources of confusion.

We present these contributions as follows. First, we discuss work related to the vehicle-based collection of data in Sec. II. We subsequently motivate the viability of camera-based surface recognition in Sec. III, before describing SURF's model training process in Sec. IV. The evaluation of SURF using more than $67\,000$ actual pictures collected during bike rides on several types of terrain is presented in Sec. V, and we conclude this work in Sec. VI.

[1] E.g., Google Maps (http://maps.google.com) or Waze (http://www.waze.com)

[2] Such as CycleStreets (http://www.cyclestreets.net), naviki (http://www.naviki.org), cycletravel (http://cycle.travel/map), and many others.

(a) Asphalt track. (b) Gravel track. (c) Cobblestone track. (d) Grass track. (e) Forest track.

Fig. 1. Sample images captured from a smartphone's built-in camera mounted in different positions on a bicycle's handlebar.

II. Related Work

A wide range of sensor data can be collected from bicycles and their riders. Through the combination of dedicated sensing devices (to measure, e.g., CO_2 and NO_x concentrations) and the sensing functionalities provided by smartphones, a comprehensive situational picture can be captured [7]–[9]. This allows bicyclists to retrieve data on their cycling performance, but also permits them to opportunistically collect environmental information while riding. In comparison to operating sensors in motorized vehicles, the limited extent of suspension and the absence of bodywork even allows for a much more precise tracking of environmental conditions.

The *BikeNet* system [10] is one example that integrates multimodal sensors to capture such parameters. Combined with location information, the system permits its users to create maps that show, e.g., experienced CO_2 levels, noise levels, and acceleration/braking data. A camera is also part of the system, yet only used to capture photographs when remotely triggered to do so. In a similar fashion, bicyclists can retrieve exercise-related information and post-exercise analysis when running the *smartphone-based real-time information feedback system* presented in [11]. Road inclination, uneven road surfaces ("bumpiness"), and the bicycle rider's body tilt are collected and used as map annotations in order to be shared with other cyclists. Other applications that promote the sharing of encountered environmental conditions with other cyclists include the *WeatherBike* [12], a bicycle that has been equipped with the required sensing devices to collect meteorological data, as well as *SmartBike* [13], which uses sensors to monitor the air conditions in a city. An offline analysis of the collected data was used in both aforementioned works, overlaying collected sensor readings on trajectory maps.

In contrast to these comparably holistic approaches of capturing a bicycle's environment, other contributions that focus on individual sensing modalities were also presented in literature. Gu et al. propose a smartphone-based system to detect cyclist behavior based on acceleration and angular rotation measurements [14]. Through the analysis of this data, dangerous bike-riding practices can be identified, and pinpointed to locations at which such riding behavior is exhibited. Another use of acceleration readings is presented in [15]–[17], whose objective is to determine the surface type of the road segment currently driven on. The authors of aforementioned works, however, note that signal processing is required to separate cyclist-induced acceleration and tilt signals from the vibration patterns induced by the road surface, thus increasing the risk of erroneous surface type detection. An application with a similar scope has been presented for car drivers, too: The *Pothole Patrol* [18] uses acceleration data to distinguish between different types of road anomalies, targeting to provide municipalities with information about necessary roadworks.

The use of camera sensors on bicycles has seen less scientific interest, despite the wide availability of handlebar mounting options for smartphones. The *PetrolWatch* system [19], even though tailored for its use in cars, however demonstrates the use of front-facing cameras to autonomously record petrol prices when passing filling stations. In a similar fashion, camera-based systems have been designed to determine the sources of traffic jams, as presented in [20]. Approaches to recognize road surface types based on pictures taken from a dashboard camera are presented in [21] and [22]. While the latter employs deep learning techniques, it is only evaluated in the context of car-based sensing and disregards surface types relevant to cyclists. Conversely, the former publication [21] focuses on the differentiation between road surface conditions (e.g., dry, wet, or snow-covered), yet does not distinguish between different types of track surfaces. Hence, we investigate the concept of using bicycle-mounted cameras in conjunction with deep learning to classify track surfaces in this manuscript.

III. Viability of Camera-based Surface Detection

The key contribution of this paper is SURF, an automatic classification system to determine the surface type of tracks

navigable by bicycles. In contrast to related work [15]–[17], where such functionality is realized based on acceleration data, we have decided to use a front-facing camera for two key reasons. First, camera images are less impacted by the cyclist's personal bikeriding style. Thus, less compensation for user-induced motion (cf. [16]) is required. Second, handlebar-mounted smartphones are frequently used for bicycle navigation. Cyclists thus have an intrinsic incentive to mount their phones in this position, in which the smartphone's rear camera points at the ground ahead of the bicycle.

We have confirmed the suitability of this mounting location through preliminary experiments, in which test subjects were prompted to affix a smartphone holder to a bicycle's handlebar. Fig. 1 shows five representative photographs taken from these mounting points. With exception of the lower part of the pictures, in which the front wheel as well as braking and gear-shifting cables are visible, the front-facing camera always captures at least a fraction of the track ahead, from which the type of the track's surface can be identified by a human.

Based on this observation of the viability of this approach, we have decided to conduct an investigation to what extent a correct classification of track surfaces using image recognition tools is possible. We realize the track surface type detection in SURF by means of state-of-the-art object recognition techniques, based on deep learning. Despite the ongoing research on the recognition of objects in images (e.g. [23]), detecting the surface type of a track is not a frequently considered setting. Thus, a prior training of the machine learning model to the specifics of road surfaces is needed before a classifier can reliably determine the surface of a track. We elaborate on this training step, the *model generation*, in the next section.

IV. Surface Type Model Generation

SURF's training phase is based on the data flow shown in Fig. 2. We explain details for each step as follows in a bottom-up order, i.e., starting with the deep learning component.

A. Deep Learning

Bicycle tracks can be expected to exhibit different appearances, based on the prevailing weather conditions and the time of the year. Even two different tracks of the same type (e.g., cobblestone) may look substantially different from each other. Hence, we have decided to employ machine learning in order to adapt to such situations, aligned with the remarkable results reported for image classification tasks in related work [24].

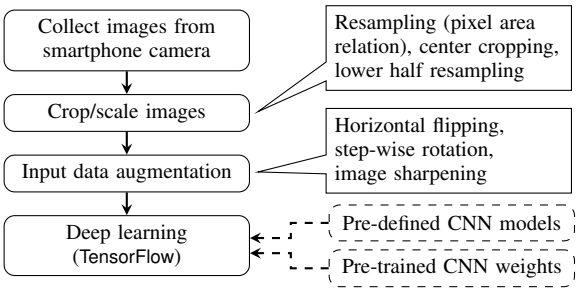


Fig. 2. Data flow of SURF's model creation phase.

SURF employs the TensorFlow deep learning framework [25] for the classification of input images. In order to accelerate the image recognition, TensorFlow can be supplied with pre-defined models and weights. For TensorFlow's configuration in SURF, we have relied on the Inception-v3 [26] and MobileNetV2 [27] models. The former is well-known for its reportedly high classification accuracy, whereas the latter model is renowned for its lightweight nature. To accelerate the training phase, TensorFlow's models were moreover pre-trained using weights resulting from the ImageNet Large Scale Visual Recognition Challenge 2012 [23].

We have substituted the classification layers of these pre-trained models by a custom layer implementation to discern between track surfaces. This classification layer is trained to differentiate five major types of track surfaces. Four of them were selected in accordance with existing work on vehicular applications [22]; the track type "forest" was added due to its relevance for bicycling. The classes "wet asphalt" and "snow" proposed in [22] were not used, given that they are seasonal occurrences. Thus, SURF supports the following surface types:

- **asphalt** (including any other paved surfaces such as concrete, tarmac, chipseal, etc),
- **gravel** (referred to as "dirt" in [22]),
- **cobblestone** (including setts),
- **grass** (as sometimes experienced when taking shortcuts across meadows), and
- **forest** tracks, which might be covered by soil, twigs, leaves, or other greens.

Sample images for each of these surface types, collected from a handlebar-mounted smartphone, are visualized in Fig. 1.

B. Input Data Augmentation

In deep learning applications, a sufficient amount of training data is required in order to attain good classification results. Due to the unavailability of well-annotated data sets of bicycle rides on different terrains, we had to rely on a data set collected ourselves (cf. Sec. V-A). In order to cope with the limited number of images in this data set, we assess the impact of *data augmentation* techniques in this work, applied in order to increase the number of training samples and prevent overfitting. Three input data augmentation techniques were employed in our analysis:

1) **Horizontal flipping**: We apply a left-right reversal to the input image.
2) **Step-wise rotation**: We tilt the captured images in steps of $3°$ from $-9°$ to $9°$. Pixels that were newly added as a result of the rotation process were colored black.
3) **Sharpening**: Through the application of a kernel, we have enhanced the contrast of edges.

By means of an example, we show the impact of image augmentation on the input image of a gravel track in Fig. 3.

C. Input Data Preparation

The choice of pre-defined CNN models also affects the required dimensions of the input data to be processed by TensorFlow. Given that we will be comparing the performances of

(a) Original. (b) Flipped. (c) Tilted $-9°$. (d) Tilted $-6°$. (e) Tilted $-3°$. (f) Tilted $3°$. (g) Tilted $6°$. (h) Tilted $9°$. (i) Sharpened.

Fig. 3. Sample images of a gravel track, demonstrating the effect of the different augementation techniques being applied to the input image.

MobileNetV2 and Inception-v3, we preprocess the input files such that they become the native input size for the CNNs. To attain pictures of 224x224 pixels (for MobileNetV2) and 299x299 pixels (for Inception-v3), we crop and/or scale the data according to the three mechanisms described as follows, all of which were realized using the OpenCV toolkit [28]:

1) **Resize**: We resample the input images to the dimensions required using the pixel area relation method.
2) **Crop center**: We cut out a fragment of the CNN model's required dimensions from the center of the input images.
3) **Lower half**: We use the lower half of the input images (assuming a portrait orientation of the smartphone on the handlebar) and resample it to fit the CNN model's input size, also using the pixel area relation method.

Examples for the three resizing operations are shown in Fig. 4 (224x224 pixels) and Fig. 5 (299x299 pixels), respectively.

V. Surface Type Model Evaluation

After having prepared the deep learning model for the identification of surface types, we assess its classification performance next. To this end, we have implemented SURF as described in Sec. IV and used it in conjunction with input data collected during a multi-day image collection campaign.

A. Collection of Input Data

The images used for SURF's evaluation were collected through bicycle rides in the areas of Clausthal-Zellerfeld and Lüneburg, Germany. A camera-equipped smartphone was attached to the bicycle handlebar using a mobile phone holder. As a result of this fixture point, most of the shots are slightly shifted to the left or to the right along the length of the handlebar (cf. Fig. 1). So that a larger amount of data could be collected, videos were recorded while riding the bicycle. From these videos, individual pictures were extracted using a video processing tool, and exported as JPEG images at the native video framerate of 30 pictures per second at a resolution of 1080x1920 pixels.

TABLE I
Statistics of collected input data.

Surface type	Number of input images
Asphalt	12,091
Gravel	13,829
Cobblestone	12,261
Grass	13,898
Forest	15,039
Total	**67,088**

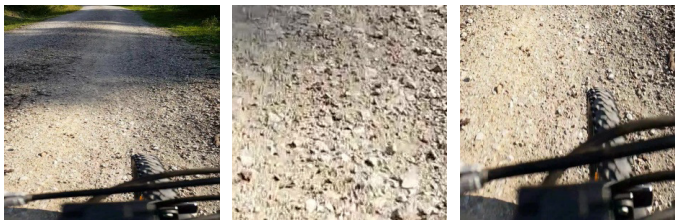
Only every sixth image from the video was used (i.e., the framerate was effectively reduced to five frames per second) and stored into the data set. The decision to omit intermediate images was the very limited extent of changes between successive video frames. Using this approach, more than 67 000 images were collected in total. A manual inspection process was conducted in order to remove images that did not fit into any of the categories or could not be unambiguously assigned. For each of the five categories considered (asphalt, cobblestone, grassland, gravel, forest soil), between 12 000 and 15 000 images remained, with their exact numbers provided in Table I. It needs to be remarked at this point that existing data sets (e.g., KITTI [29] or RobotCar [30]) were not suitable for our evaluation, because they commonly focus on a single type of environment, predominantly paved roads, and lack the required diversity.

B. Evaluation Setup

All results presented in this section are top-1 accuracy levels (i.e., true positive rates), as reported by TensorFlow. We have selected a 20/70/30 split for our cross-validations, i.e., 20 % of the input were separated for testing (with results shown as follows) and not used in the training phase. Out of the remaining data, 70 % were used for training, and 30 % for validation, in accordance with machine learning best practices.

(a) Bilinear resizing. (b) Center cropping. (c) Lower half resize.

Fig. 4. Visual comparison of the resizing approaches for a 224x224 pixel output using an input image containing the "grass" surface type.



(a) Bilinear resizing. (b) Center cropping. (c) Lower half resize.

Fig. 5. Visual comparison of the resizing approaches for a 299x299 pixel output using an input image containing the "gravel" surface type.

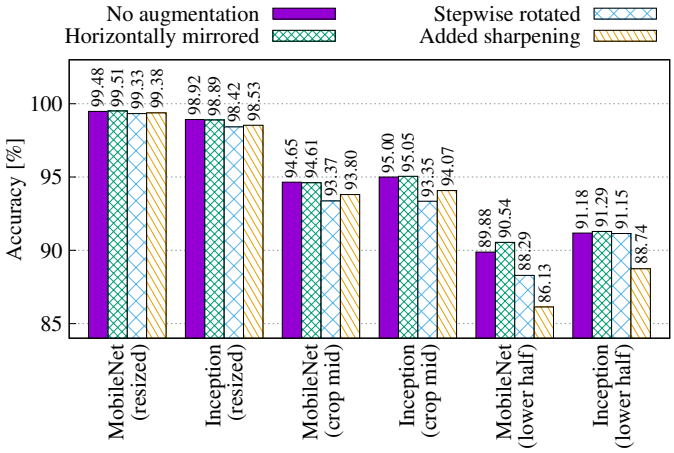


Fig. 6. Classification accuracy results for all combinations of CNN model, cropping method, and augmentation technique.

C. Impact of the CNN Model and Image Cropping

Our first evaluation is targeted at determining the accuracy achievable by the two CNN model types used by SURF, and to what extent image cropping influences these results. We thus visualize the classification accuracy values for all combinations of the considered CNN models and image cropping methods in Fig. 6. Note that we have trained, validated, and tested the deep learning models with input images that were all cropped/resized in the same way.

The highest overall classification accuracy level (99.4 % on average) is observed when using MobileNetV2 in conjunction with resized full images, i.e., pictures without any cropping applied to them. For the same type of images, Inception-v3 yields a slightly lower, yet nonetheless remarkable, classification performance (98.7 %). This not only confirms the general viability of recognizing surface types from photos taken by handlebar-mounted smartphones, but also shows that both classifiers are suitable choices to accomplish this task.

Much lower classification accuracy levels result from the application of the two other image cropping alternatives: A degradation by 4.82 percentage points on average is observed when the center-cropping method is employed, and a further 4.59 percentage points of classification accuracy are lost when using the lower half of the input images only.

D. Impact of Data Augmentation

Supplementally, we assess how an augmentation of the training and validation data impacts the classification accuracy. Each of the augmentation mechanisms described in Sec. IV-B was applied and analyzed separately. Results are shown in Fig. 6 and discerned by their color shadings. Note that we have used both the original images and their processed derivatives for training and validation. In the testing phase, only non-augmented images were being used to ensure realistic results.

The results show that none of the augmentation methods has a consistently positive impact on the classification accuracy. In some cases, adding horizontally flipped images to SURF's training leads to slight accuracy improvements. The remaining two augmentation techniques consistently have a negative impact on the overall results. The best overall results were thus reported when MobileNetV2 was used with horizontal mirroring used for the data augmentation, with a reported accuracy of 99.51 %. This result is closely tracked by the same setup without input data augmentation, at 99.48 % accuracy.

TABLE II
CONFUSION MATRIX FOR SURFACE TYPE CLASSIFICATIONS (IN PERCENT).

Actual surface type	Classified as: asphalt	cobble	gravel	grass	forest
asphalt	**99.25**	0	0.65	0	0.1
cobblestone	0.2	**99.65**	0.1	0	0.05
gravel	0.4	0	**99.3**	0.3	0
grass	0	0	0.1	**99.75**	0.15
forest	0.05	0	0.15	0.2	**99.6**

E. Insights and Discussions

We gained several other insights during the experiments, selected ones of which we reports as follows.

Confusion: Observed confusion was not equally distributed but almost always experienced between the *gravel* and *asphalt* types (and vice versa). This can be observed well from the confusion matrix, shown in Table II. Confusion was mainly observed at points where surface types change (shown, e.g., in Fig. 7a), or when multiple surface types are visible in the same image (see Figs. 7b and 7c).

Processing requirements: While we have performed the evaluation on a desktop computer, MobileNetV2 has been documented to be sufficiently lightweight to be executed on smartphones [31]. Moreover, promising approaches towards embedded machine learning are currently being proposed, e.g. in [32]. Executing the image analysis locally on a bicycle-mounted device would significantly reduce the incurred network traffic: Only extracted surface information would need uploading into the map provider's database instead of sending captured images or even video streams to a processing server.

Comparison to related work: In published works on road surface detection, consistently lower classification accuracy levels are reported (75.051 % in [15], 88 % in [21], and 92 % in [22]). Our approach outperforms these results, which we partially attribute to the following reasons:

- Less variability in the camera mounting position,
- Less vehicular traffic in the collected images of bicycling tracks, i.e., better view of the track ahead,
- Training and testing data were collected from almost the same environment (little geographical diversity), and

(a) Asphalt surface, classified as cobblestone. (b) Asphalt surface, classified as grass. (c) Asphalt surface, classified as gravel.

Fig. 7. Selected instances of surface types falsely classified by SURF.

- The methodological assessment of the parameter space in conjunction with the choice of the best-suited parameters.

VI. Conclusions and Outlook

In this paper we have presented SURF, a system based on convolutional neural networks to detect the surface types of tracks navigable by bicycle riders. Its required input images can be opportunistically collected from handlebar-mounted smartphones, which can simultaneously serve as a navigation aid. This crowdsourcing approach to classify track surfaces can be used to opportunistically complement the information in online maps, such as OpenStreetMap. We have demonstrated the efficacy of SURF with the help of a data set composed of 67,000 images we have collected during bicycle rides. With an overall accuracy of $99.51\,\%$, SURF has demonstrated its viability and readiness for practical use.

In the future, we plan to optimize SURF for its operation on mobile devices and build a crowdsensing system that collects track surface information at scale and improves online maps therewith. Moreover, we consider to extend SURF by supplementary sensing modalities (e.g., acceleration data) to improve the accuracy of the surface classification even more, and possibly also provide further map data annotations.

Acknowledgments

This work was financially supported by the Simulationswissenschaftliches Zentrum Clausthal-Göttingen (SWZ) as part of the *HerMes* project.

References

[1] J. Pucher and R. Buehler, "Walking and Cycling for Healthy Cities," *Built Environment*, vol. 36, no. 4, 2010.

[2] J. Mason, L. Fulton, and Z. McDonald, "A Global High Shift Cycling Scenario: The Potential for Dramatically Increasing Bicycle and E-bike Use in Cities Around the World, with Estimated Energy, CO2, and Cost Impacts," 2012, Institute for Transportation & Development Policy and the University of California, Davis.

[3] Mairie de Paris, "Paris: Capitale de Vélo 2020," Online: https://api-site.paris.fr/images/78376, 2015.

[4] Chicago Department of Transportation, "Chicago Streets for Cycling Plan 2020," Online: https://www.cityofchicago.org/content/dam/city/depts/cdot/bike/general/ChicagoStreetsforCycling2020.pdf, 2012.

[5] Ruhr Regional Association (Regionalverband Ruhr), "Feasibility Study Radschnellweg Ruhr (RS1)," Online: http://www.radschnellwege.nrw/fileadmin/user_upload/projekte/rs1/downloads/RS1_abstract_w_21.11.2016.pdf, 2016.

[6] M. Haklay and P. Weber, "OpenStreetMap: User-Generated Street Maps," *IEEE Pervasive Computing*, vol. 7, no. 4, 2008.

[7] M. Dozza and A. Fernandez, "Understanding Bicycle Dynamics and Cyclist Behavior From Naturalistic Field Data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, 2014.

[8] J. A. Afonso, F. J. Rodrigues, D. D. R. Pedrosa, and J. L. Afonso, "Automatic Control of Cycling Effort using Electric Bicycles and Mobile Devices," in *Proc. World Congress on Engineering*, 2015.

[9] P. Menaspà and C. R. Abbiss, "Considerations on the Assessment and Use of Cycling Performance Metrics and their Integration in the Athlete's Biological Passport," *Frontiers in Physiology*, vol. 8, 2017.

[10] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "The BikeNet Mobile Sensing System for Cyclist Experience Mapping," in *Proc. 5th International Conference on Embedded Networked Sensor Systems (SenSys)*, 2007.

[11] Y. Zhao, Y. Su, and Y. Chang, "A Real-Time Bicycle Record System of Ground Conditions Based on Internet of Things," *IEEE Access*, vol. 5, 2017.

[12] J. J. Cassano, "Weather Bike: A Bicycle-Based Weather Station for Observing Local Temperature Variations," *Bulletin of the American Meteorological Society*, vol. 95, no. 2, 2014.

[13] F. Corno, T. Montanaro, C. Migliore, and P. Castrogiovanni, "SmartBike: an IoT Crowd Sensing Platform for Monitoring City Air Pollution," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 6, 2017.

[14] W. Gu, Y. Liu, Y. Zhou, Z. Zhou, C. J. Spanos, and L. Zhang, "BikeSafe: Bicycle Behavior Monitoring via Smartphones," in *Proc. ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2017.

[15] M. Hoffmann, M. Mock, and M. May, "Road-quality Classification and Bump Detection with Bicycle-mounted Smartphones," in *Proc. 3rd International Conference on Ubiquitous Data Mining (UDM)*, 2013.

[16] J. Takahashi, Y. Kobana, Y. Tobe, and G. Lopez, "Classification of Steps on Road Surface Using Acceleration Signals," in *Proc. 12th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous)*, 2015.

[17] R. Suzuki, J. Takahashi, and Y. Tobe, "Effectivity Evaluation of YKOB in Actual City Road," in *Proc. IEEE/SICE International Symposium on System Integration (SII)*, 2017.

[18] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring," in *Proc. 6th International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2008.

[19] Y. Dong, S. S. Kanhere, C. T. Chou, and N. Bulusu, "PetrolWatch: Using Mobile Phones for Sharing Petrol Prices," in *Proc. 7th Annual International Conference on Mobile Systems, Applications, and Services (MobiSyS)*, 2009.

[20] X. Zhang, H. Gong, Z. Xu, J. Tang, and B. Liu, "Jam Eyes: A Traffic Jam Awareness and Observation System Using Mobile Phones," *International Journal of Distributed Sensor Networks*, vol. 8, no. 12, 2012.

[21] Y. Qian, E. J. Almazan, and J. H. Elder, "Evaluating Features and Classifiers for Road Weather Condition Analysis," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2016.

[22] M. Nolte, N. Kister, and M. Maurer, "Assessment of Deep Convolutional Neural Networks for Road Surface Classification," *CoRR*, 2018. [Online]. Available: http://arxiv.org/abs/1804.08872

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, 2015.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proc. 25th International Conference on Neural Information Processing Systems (NIPS)*, 2012.

[25] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: A System for Large-Scale Machine Learning," in *Proc. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2016.

[26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *CoRR*, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[27] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *CoRR*, 2018. [Online]. Available: http://arxiv.org/abs/1801.04381

[28] K. Pulli, A. Baksheev, K. Kornyakov, and V. Eruhimov, "Real-time Computer Vision with OpenCV," *Communications of the ACM*, vol. 55, no. 6, 2012.

[29] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets Robotics: The KITTI Dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, 2013.

[30] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000 km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, 2017.

[31] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *CoRR*, 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[32] S. Venkataramani, K. Roy, and A. Raghunathan, "Efficient Embedded Learning for IoT Devices," in *Proc. 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2016.