Viewing Flowers at their Most Beautiful Moments: A Crowd Sensing Application

Weifeng Xiong
College of Computer and Data Science
Fuzhou University
Fuzhou, China
211027177@fzu.edu.cn

Fangwan Huang
College of Computer and Data Science
Fuzhou University
Fuzhou, China
hfw@fzu.edu.cn

Zhiyong Yu
College of Computer and Data Science
Fuzhou University
Fuzhou, China
yuzhiyong@fzu.edu.cn

Xianwei Guo
College of Computer and Data Science
Fuzhou University
Fuzhou, China
xwguo111@163.com

Binwei Lin
College of Computer and Data Science
Fuzhou University
Fuzhou, China
2292639773@qq.com

Qiquan Cai
College of Computer and Data Science
Fuzhou University
Fuzhou, China
cq_diarmuid@yeah.net

***Abstract*—To assist people's itinerary planning for viewing flowers, it is very meaningful to visualize the different stages of specific flowers with high spatio-temporal resolution. To achieve this goal, this paper realized a crowdsensing application called *Hanami*, which means 'flower viewing'. The implementation of this application contains three modules: data sensing, flower classifier, and visualization. Particularly, the flower classification module utilized a residual network to identify the types and stages of flowers from crowdsensed photos. For the visualization module, a bilayer clustering view method was designed to aggregate the points on the map, which can be further clustered by different features of flowers. Experimental evaluation showed that *Hanami* can help users view flowers at their most beautiful moments.**

Keywords—crowdsensing, flower viewing, ResNet, bilayer clustering view

I. Introduction

After viewing flowers online during the COVID-19 epidemic, the outdoor experience has become an aspiration for tourists and flower lovers. Choosing the right time and place becomes a key issue as people make travel arrangements. To address this issue, some people manually collect information on the locations and times of flowers in small areas and then visualize them on maps. However, this method is significantly limited by delayed updates and high costs, which results in difficulty in obtaining fine-grained distribution maps for large areas.

Mobile crowdsensing (MCS) offers an unprecedented opportunity to collect sensing data on a large scale from mobile devices. The application of MCS can provide a wide range of services for people's daily life, such as environmental monitoring [1] and traffic management [2]. In these applications, a fine-grained overview of environmental phenomena can be obtained by aggregating large amounts of geographically distributed measurements collected by participants with their mobile phones.

Existing applications for viewing flowers only focus on identifying types of flowers, but ignore the impact of the spatio-temporal distribution of flowers at different stages. This problem is highly difficult due to several challenges in the process of implementation. First, how to get information about flowers at high spatio-temporal resolution. The stages of flowers may be different with time and space. For example, the blooming time for the same species is always different in each city. Fortunately, CrowdOS, a mobile computing platform for crowd sensing, is recommended to cope with this problem [3]. Specifically, photos and locations generated by persons at different times are collected through participatory sensing to obtain spatio-temporal data with high resolution [4]. Second, how to guarantee the accuracy of classifying the type and stage of flowers under limited data. Residual network (ResNet) widely used in image classification has shown good generalization in several domains (e.g., garbage classification [5] and plant identification [6]). However, the limited datasets for training could bring many undesirable consequences [7]. To deal with the impact of limited data on models, this paper used transfer learning to implement a classification module. Third, how to visualize the spatio-temporal distribution of flowers briefly. It is easy to cause problems such as overlap while displaying large-scale data. A traditional solution is to cluster by location. However, location clustering cannot effectively demonstrate the distribution and their division within clusters. Thus, a bilayer clustering view was designed to visualize the collected data, while the outer-layer cluster was based on location, and the intra-layer cluster was based on the types and stages of flowers.

To deal with the above problems, this paper innovatively applied mobile crowdsensing to plant identification and implemented an application called Hanami. Our application collected both photos of flowers and their spatio-temporal information during the sensing cycle. Then, the collected fine-grained data were used to map floral spatio-temporal distribution. The main contributions of this paper were summarized below:

- CrowdOS was innovatively applied to an application of plant identification, which allowed us to obtain high-resolution data by participatory sensing.
- A bilayer clustering view was designed to visualize floral spatio-temporal distribution. Compared with the traditional location clustering methods, our method can provide more intra-cluster information.
- The application was evaluated on real data set, which proved that it can help improve the rationality and efficiency of flower viewing.

The remainder of the paper is structured as follows. Section II presents a review of related work. In Section III, the overall structure of the ***Hanami*** application is introduced. Sections IV and V discuss the classification model and the bilayer clustering view. Section VI shows the implementation

This work is supported by the Research Project for Young and Middle-aged Teachers of Fujian Province under Grant No. JAT210007, and the Fujian Engineering Research Center of Big Data Analysis and Processing.

978-1-6654-6457-4/22/$31.00 ©2022 IEEE
DOI 10.1109/MSN57253.2022.00149

and evaluation details of the ***Hanami*** application. Finally, some conclusions are presented in Section VII.

II. Related Work

In this section, a review of related work on plant identification and data visualization is given.

A. Plant Identification

In recent years, applications for plant identification have gradually emerged. In terms of functionality, Leafsnap [8] is one of the first mobile applications for plant identification. It uses a computer vision component to segment the leaves, which makes it possible to achieve plant recognition. However, the leafless plants will be discarded directly in this application. In contrast to Leafsnap, Flower Partner [9] is allowed to identify leafy and leafless plants and shows a location distribution map. However, it does not provide more information about plants, ignoring the impact of spatio-temporal information on the users. For classifiers, deep neural networks achieve remarkable success. For example, Qin et al. [10] used a pre-trained VGG-16 network to learn the features of flowers, which achieved an accuracy of 87.6%. However, there are some weaknesses in VGG-16, such as long training time, large storage capacity, and poor deployability of applications. Tian et al. [11] introduced a DarkNet-19 network, which is a good solution to the problems on VGG-16, but its internal implementation is discouraging. To deal with the above problems, we adopted the ResNet50 network [12] as a classifier in our application.

B. Data Visualization

Data visualization, a key technique of data analysis, has been used in different areas (weather, electricity, aviation, etc.). It consists of three components: data, marks, and the mapping between them [13]. The collected raw data are typically pre-processed, such as outlier processing, missing data completion, and data normalization. So a high-quality dataset can be obtained to ensure that the data can better serve the data analysis or data mining work. As these three elements are defined, the data can be rendered.

For collected spatial data, maps are one of the common ways of data visualization. With available map tools, presenting these spatial data on a map makes it easier to find points or areas of interest. The most popular map-based visualization techniques are heat maps, dot maps and flow maps, area maps. Purahoo et al. [14] extracted the features of spatial data and then labeled parking spots on the map. Furthermore, they clustered the parking spots within limits, which decreased the time to find parking spots. Fig. 1 shows the distribution of available parking spots. Fig. 2 describes a customized marker represented by photos [15], and these photographs are of local attractions. These methods have effectively enhanced the quality of system services, for example, capabilities of data processing and retention rates. However, some other features of data cannot be visualized using the techniques discussed above. Spatial-based feature visualization without comparing to any other features makes it difficult to highlight points of interest. In particular, the formed clusters cannot present their intra-clusters division. In this paper, a bilayer clustering view was designed to address this problem, which was evaluated on real data set.

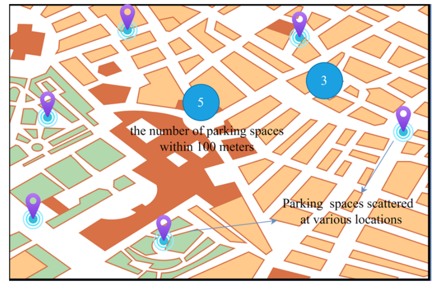


Fig. 1. Map visualization of parking spots.

Fig. 2. Customized markers at each attraction on the map.

III. Module Design

The application ***Hanami*** implemented in this paper contained three modules: data sensing, flower classifier, and visualization, whose structure is shown in Fig. 3.

Data Sensing. This module was used to collect spatio-temporal data with high resolution. We formulated the task structure in the following format: {‘Flowers’, ‘Latitude’, ‘Longitude’, ‘UploadTime’, ‘UpdateTime’}, and then published a crowdsensing task on the platform. Among these features, ‘flowers’ denoted the uploaded photos; ‘latitude’ and ‘longitude’ uniquely determined the location of flowers; ‘uploadtime’ defaulted to the time of uploading, which was also used to classify the flowering stages as mentioned later. However, if participants delay the submissions, this property will be affected. To this end, we introduce another feature called ‘updatetime’ to reduce this error. It originates from the metadata [17] recorded in the photos and allows users to identify the real-time capture. Participants are asked to collect and submit the sensing information mentioned above to the MCS server. Although privacy and security issues [18] threaten MCS systems, many works have been done to analyze and address them, such as a trusted data collection method proposed in [19]. Currently, our work mainly focused on the application of data collection, but the issue mentioned above will also be added to future work to improve the reliability of the system.

Flower Classifier. Deep learning is a powerful classifier with good performance, especially in image recognition tasks. But the number of parameters and the practical performance in the production environment are issues that we should consider. With this in mind, this paper introduces a residual network as a classifier, which is still popular because of its simple network and stable performance. With the classifier, uploaded images are recognized and then stored in the database.

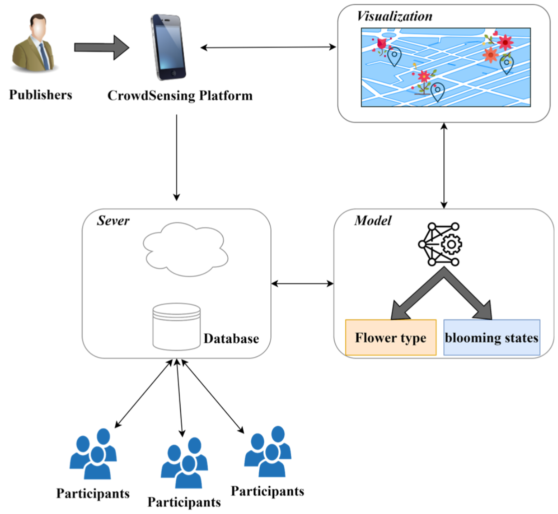


Fig. 3. System Architecture of ***Hanami***.

Visualization. The collected data can be visualized on a map by groups of geographical coordinates. Here, this paper used a popular visualization tool, Dot Maps, to visualize these data. Given the large amount of data gathered in an area, it can be difficult for users to find the messages they are looking for. Therefore, an interactive data visualization technique was adopted that aims to provide faster responsiveness and more intuitive visualization. In this paper, a bilayer clustering view was designed to visualize data on the map (as described in Section V) and display them at different granularities by adjusting the map level.

IV. Flower Classifier

The goal of ***Hanami*** is to be able to classify and display flower photos sensed by participants, which assists people's itinerary planning for viewing flowers. However, how to accurately identify the type and stage of flowers? The process of implementation is described below.

A. Data collection and Processing

The goal of data collection is to find datasets that can be used to train classifiers. Currently, two sources of data collection are available. One is the data sensing module mentioned in the previous section, where participants can receive rewards for completing the tasks. Meanwhile, 12 volunteers were hired to take pictures of flowers in each region of Fuzhou. In the end, about 1400 usable images were collected.

Next, our work focuses on data preprocessing. First, this paper classified the dataset into 8 groups by species. Based on the stages of flowers, each group was further divided into four classes, including flowerless, budding, blooming, and withering. Fig. 4 shows the four stages of Lotus. For classifiers, the quality of datasets greatly affects their performance. Thus, the original images were processed. The goal of image processing is to reduce useless information and augment the available parts. This paper adopted some common approaches, such as denoising, averaging, and image enhancement. The data was expanded to reduce overfitting. Instead of a larger image dataset, random transformations were applied to the training images (e.g. rotation or horizontal flip) to artificially introduce sample diversity, which contributed to making the

Fig. 4. Four stages of Lotus.

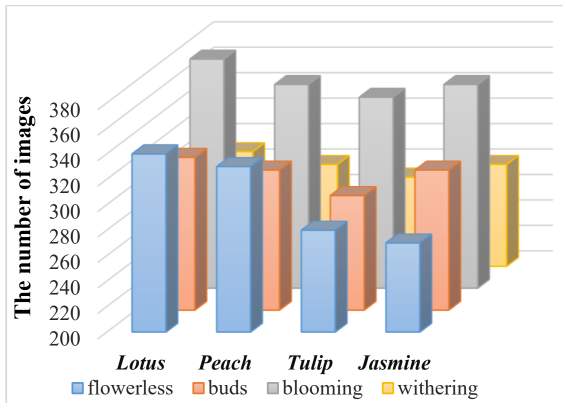


Fig. 5. Data distribution of several types of flowers in the dataset.

model exposed to different aspects of the training data and reduced overfitting. Fig. 5 shows the distribution of the four flowers in the dataset at different stages.

B. Flower Classifier

Considering that there are 32 categories for the combination of floral types and stages, and a limited dataset is difficult to train a model with high accuracy in this multi-category classification. At the same time, users may only require one of the results instead of all of them to satisfy their demands. Therefore, the two classifiers were trained separately for those two classification tasks.

The dataset was divided randomly and set the ratio of the training set to the test set at 8:2. During training, the original images were center cropped and resized to the target input size by scaling. To initialize our network, This paper used TensorFlow to load the parameters of the resnet50 model pre-trained on the imagenet dataset. During the fine-tuning process, two steps were executed. The first step was to initialize the network with pre-trained network parameters. Instead of discarding the original classification layer, a new fully connected classification layer and softmax layer were added after the original structure to adapt to the new tasks [20]. We trained all the networks using the RMSProp optimizer and set the batch size to 64, the initial learning rate to 0.001, and the epoch to 50. Rather than training two classifiers in parallel, we first trained a model for floral type classification. Then that classifier was used as a pre-trained model to train the stage classifier.

For the purpose of this work, the trained models were deployed on the server side. Docker containers provide a solution for multi-model deployment. The trained models were saved as onnx model files, which means that only the topology and weights of their networks were saved. The final results can be obtained by loading different onnx model files. For communication, the Flask framework was used to implement the RESTful API service, which allowed to easily access to the functional interface.

V. Bilayer Clustering View

To obtain a visual map of the spatio-temporal distribution of flowers, this paper designed a bilayer clustering view. Moreover, an interactive approach was adopted to provide a distinct visualization for users. Our method incorporated three constraints, namely, latitude and longitude, floral type, and stages, which were used to partition the data. Similar to hierarchical clustering, a bottom-up hierarchical approach was applied to group the data into two layers. Initially, we divided the map into 50-meter grid regions and then mapped all the data to the grids based on latitude and longitude. During clustering, the outer layer aggregates those data by grid density G_ρ and is calculated as:

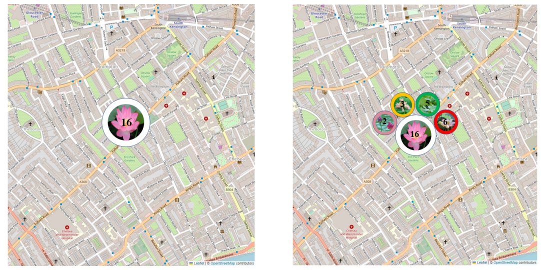
$$G_{\rho_i} = \frac{N_i}{\sum_{i=1}^{n} N_i} \tag{1}$$

As (1) shows, the map is divided into n girds, and G_{ρ_i} represents the density of each grid, N_i denotes the amount of data in this grid.

This paper introduced a threshold denoted by $\delta(0<\delta<1)$. If there are some regions whose density is higher than the given threshold, then the clustering starts in that region, otherwise, it remains unchanged. Intra-layer clustering is mainly dependent on data features. As mentioned above, the classifier returns the results to the server, which contains the types and stages of flowers while the client requests the interface. Therefore, both features were used to further divide the data in the outer layer. In this setup, this paper set the weight of each feature to fine-tune the visualized results. Considering that users tend to seek flowers in their blooming period, the weight of blooming was increased and the other three stages were decreased to match their requirements. To minimize the response loss caused by aggregation, an interactive method was introduced to present the bilayer clustering. While the mouse is hovering over the outer layer, the intra-layer begins to aggregate and the obtained clusters will surround the outer layer. After mouse removals, the original stage is restored. Fig. 6(a) shows the visualization of the outer-layer cluster, which turned into Fig. 6(b) after triggering the aggregation of the intra-layer clustering. This approach demonstrates the division at different granularities with minimal operations and minimal infoboxes, as well as accelerates the identification of features contained in the clusters.

VI. Implementation And Evaluation

In this section, the implementation of Hanami is first described. Then, the system performance is evaluated from classification and visualization, which demonstrates that ***Hanami*** is an effective application for viewing flowers.



(a) (b)

Fig. 6. The examples of the clustering view, (a) and (b) are respectively the clustering results of the outer-layer cluster and intra-layer cluster.

A. System Implementation

The ***Hanami*** application was developed for the web platform, which included both client and server. The server contained data mapping, data processing, and web services. Data mapping was to map the spatial data to the corresponding grids; data processing had already been described in Section IV; a key requirement for the web service was that the server needed to support a certain amount of concurrency. Therefore, this paper used a caching mechanism to reduce the pressure on the server caused by high access volumes with Redis. The client provided a visual interface and interacts with users. This application is available at www.fzu-urbansensing.com. Fig. 7 shows the components related to the interface below.

Search tab. The search tab was designed to provide a search service that included four options, including floral name, location, time, and floral stages, which allowed users to efficiently find the required information. In particular, the stage search service was supported by the classifiers in Section IV. Time-based search provided a date selection, and the server parsed the selected date into the corresponding flowering period, then returned the filtered data. With different choices, users can view the current or future floristic distribution.

Main panel. The main panel contained a visualization capability that allowed the user to view the spatio-temporal distribution of the data on an interactive map. For point markers, each marker was marked by an uploaded image. Their borders were marked with different colors, which indicated different floristic stages, and the corresponding legend was displayed on the side. Clicking on the map markers displayed a label with a general description, its flowering period, location, and current stage. The application also provided a Wikipedia for the specific flowers, and clicking on the link would open a new browser tab. For clusters, we made some distinctions from point markers. For example, a number was placed in the middle of clusters to denote the number of points in the cluster. The border color of the outer-layer clusters had been changed to white, while the intra-layer was attached around the outer layer. This panel was controlled by the external input, and the visualization would be updated if the input changes. Users can also adjust the map layers to show the distribution of flowers and their info at different resolutions.

Sidebar. The sidebar contained all the categories returned by the server and the input options that controlled the visualization of the main panel. The main panel was reactive to the sidebar, and content in the main panel was updated immediately as changes were made to the options. The input in the sidebar allowed users to choose floral types, as they searched for a certain flower by the search tab, only one result would be presented in the sidebar.

B. *Evaluation*

In this section, we evaluate the components of ***Hanami***. This paper first describes the performance of the classifiers on a real dataset, then evaluates the effectiveness of bilayer clustering on the system by questionnaires.

Flower Classifier. The proposed classification model was evaluated on the test set. The accuracy of the model reached 92.4%, which indicated that the model could accurately classify the types and stages of flowers. Furthermore, the model was encapsulated as an API interface, which reduced the interdependence of system components, and promoted the extensibility of ***Hanami***.

Bilayer Clustering View. Questionnaires were used to evaluate our bilayer clustering schemes in terms of their effectiveness in motivating users to the application. This paper divided the 54 volunteers into 6 groups, and they were asked to use the system with and without the bilayer clustering scheme. They were also asked to give feedback, and whether they would like to use the application. With the collected data, the usage of each group was counted. Fig. 8 shows the probability of each group, it is obvious that the participation probabilities of bilayer clustering are higher than those of normal location clustering. Among their feedback, the bilayer clustering scheme is more intuitive than the location-based clustering, which effectively visualizes the distribution of flowers and reduced the number of operations.

VII. Conclusion

The application ***Hanami*** solves the problem that existing applications for viewing flowers cannot provide the stages of flowers at higher spatio-temporal resolutions, and assists people's itinerary planning for viewing flowers. There are some advantages of ***Hanami***:

- Utilize the sensing capabilities of mobile devices to obtain the distribution of flowers with high spatio-temporal resolution at a lower cost.

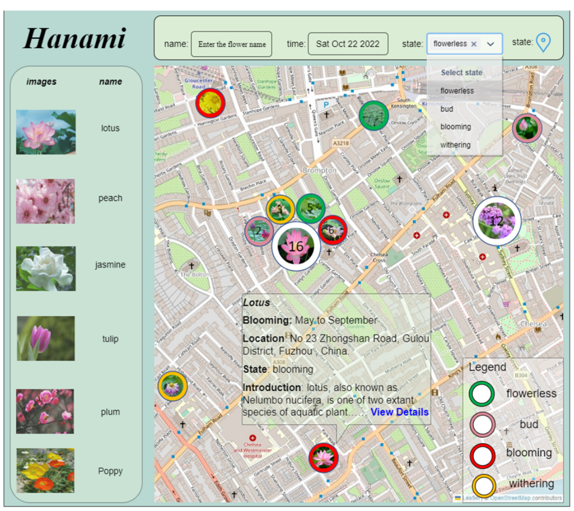


Fig. 7. Map visualization of components in ***Hanami***.

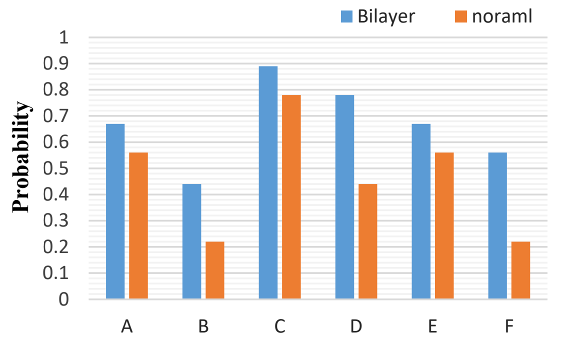


Fig. 8. Probabilities among users who used the application with two different solutions.

- Fine-tuning the pre-trained network with only limited data results in a highly accurate model that reduces the reliance on labeled data.
- The bilayer clustering view concisely demonstrates the distribution of flowers, which not only optimizes the display of images but also reduces operations for users.
- By combining fine-grained spatio-temporal data, ***Hanami*** provides users with a date-specific search service and a high-precision location service. Therefore, it is convenient for users to choose the time and place to enjoy flowers.

In summary, ***Hanami*** provides a good experience for users while ensuring performance, and contributes to users' itinerary planning for viewing flowers.

References

[1] Y. Liu, X. Ma, L. Shu, Q. Yang, Y. Zhang, Z. Huo, et al. "Internet of things for noise mapping in smart cities: state of the art and future directions," IEEE Network, 2020, 34(4): 112-118.

[2] X. Zhu, Y. Luo, A. Liu, W. Tang, and M. Z. A. Bhuiyan. "A deep learning-based mobile crowdsensing scheme by predicting vehicle mobility", IEEE Transactions on Intelligent Transportation Systems, 2020, 22(7): 4648-4659.

[3] Y. Liu, Z. Yu, B. Guo, Q. Han, J. Su, and J. Liao. "CrowdOS: A ubiquitous operating system for crowdsourcing and mobile crowd sensing". IEEE Transactions on Mobile Computing, 2020, 21(3): 878894.

[4] Salil S. Kanhere. "Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces". International Conference on Distributed Computing and Internet Technology, 2013, 7753: 19-26.

[5] Z. Kang, J. Yang, G. Li, and Z. Zhang. "An automatic garbage classification system based on deep learning", IEEE Access, 2020, 8: 140019-140029.

[6] N. H. Krishna, M. Rakesh, and R. RamKaushik. "Plant Species Identification Using Transfer Learning-PlantCLEF 2020", CLEF, 2020, 2696: 139-144.

[7] J. Barbedo. "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification", Computers and electronics in agriculture, 2018, 153: 46-53.

[8] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. Soares. "Leafsnap: A computer vision system for automatic plant species identification", European conference on computer vision, 2012, 7573: 502-516.

[9] M. Li. "flower partners: A new medium of knowledge service in the era of artificial intelligence", Publishing Reference, 2018, 8: 23-24.

[10] M. Qin, Y. Xi, and F. Jiang. "A new improved convolutional neural network flower image recognition model", 2019 IEEE Symposium Series on Computational Intelligence (SSCI), 2019: 3110-3117.

[11] M. Tian, H. Chen, and Q. Wang. "Flower identification based on Deep Learning", Journal of Physics: Conference Series. IOP Publishing, 2019, 1237(2): 022060-022070.

[12] Z. Wu, C. Shen, and A. Van Den Hengel. "Wider or deeper: Revisiting the resnet model for visual recognition", Pattern Recognition, 2019, 90: 119-133.

[13] X. Qin, Y. Luo, N. Tang, and G. Li. "Making data visualization more efficient and effective: a survey", The VLDB Journal, 2020, 29(1): 93-117.

[14] Z. Purahoo, S. Cheerkoot-Jalim. "SenseAPP: An IoT-Based Mobile Crowdsensing Application for Smart Cities", 2020 3rd International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering (ELECOM), 2020: 47-52.

[15] P. Fränti, R. Mariescu-Istodor, and L. Sengupta. "O-Mopsi: Mobile orienteering game for sightseeing, exercising, and education", ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 2017, 13(4): 1-25. http://cs.uef.fi/o-mopsi.

[16] M. Cristani, A. Perina, U. Castellani, and V. Murino. "Content visualization and management of geo-located image databases", CHI'08 extended abstracts on Human factors in computing systems, 2008: 2823-2828.

[17] C. Riggs, T. Douglas, and K. Gagneja. "Image Mapping through Metadata", 2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC), 2018: 1-8.

[18] H. Tao, M. Z. A. Bhuiyan, M. A. Rahman, G. Wang, T. Wang, M. M. Ahmed, and J. Li. "Economic perspective analysis of protecting big data security and privacy", Future Generation Computer Systems, 2019, 98: 660-671.

[19] T. Wang, Y. Li, W. Fang, W. Xu, J. Liang, Y. Chen, et al. "A comprehensive trustworthy data collection approach in sensor-cloud systems", IEEE Transactions on Big Data, 2018, 8(1): 140-151.

[20] T. Shermin, S. W. Teng, M. Murshed, G. Lu, F. Sohel, and M. Paul. "Enhanced transfer learning with imagenet trained classification layer", Pacific-Rim Symposium on Image and Video Technology. Springer, Cham, 2019, 11854: 142-155.