GPS2Vec: Pre-trained Semantic Embeddings for Worldwide GPS Coordinates

Yifang Yin, Ying Zhang, Zhenguang Liu, Sheng Wang,
Rajiv Ratn Shah, and Roger Zimmermann, Senior Member, IEEE

***Abstract*—GPS coordinates are fine-grained location indicators that are difficult to be effectively utilized by classifiers in geo-aware applications. Previous GPS encoding methods concentrate on generating hand-crafted features for small areas of interest. However, many real world applications require a machine learning model, analogous to the pre-trained ImageNet model for images, that can efficiently generate semantically-enriched features for planet-scale GPS coordinates. To address this issue, we propose a novel two-level grid-based framework, termed GPS2Vec, which is able to extract geo-aware features in real-time for locations worldwide. The Earth's surface is first discretized by the Universal Transverse Mercator (UTM) coordinate system. Each UTM zone is then considered as a local area of interest that is further divided into fine-grained cells to perform the initial GPS encoding. We train a neural network in each UTM zone to learn the semantic embeddings from the initial GPS encoding. The training labels can be automatically derived from large-scale geotagged documents such as tweets, check-ins, and images that are available from social sharing platforms. We conducted comprehensive experiments on three geo-aware applications, namely place semantic annotation, geotagged image classification, and next location prediction. Experimental results demonstrate the effectiveness of our approach, as prediction accuracy improves significantly based on a simple multi-feature early fusion strategy with deep neural networks, including both CNNs and RNNs.**

***Index Terms*—GPS, semantic embedding, neural networks, geo-aware applications**

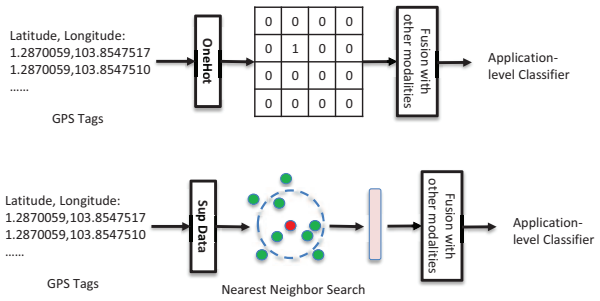


Fig. 1: Illustration of existing GPS encoding techniques.

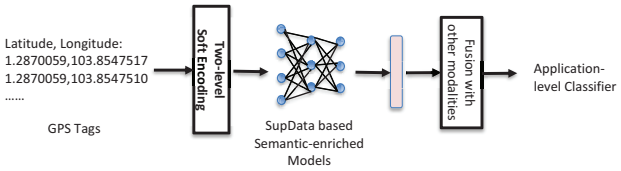


Fig. 2: Illustration of our proposed GPS2Vec approach.

I. Introduction

With the ubiquity of sensor-equipped smartphones it is common that multimedia documents, uploaded to the Internet from all over the world, have GPS coordinates associated with them. Such geotags can provide rich contextual information that is crucial for, *e.g.*, information understanding, document retrieval, and personalized recommendations [1]–[3]. For instance, a photo of clouds and a photo of a field of snow can be quite similar in their visual appearances. If a geotag is available, it could tell us that the photo was taken at a location where it never snows and so it is more likely to depict clouds [1]. Despite the significant importance of geotags, previous research has focused on extracting hand-crafted features from GPS locations that are tailored for specific applications, often targeting small areas of interest. A global method that can handle worldwide geotags is preferred as it can be applied to both global and regional applications. Conversely, extending a regional method to global applications is usually difficult. Moreover, constructing an end-to-end pre-trained model for geospatial feature extraction would be superior to existing hand-crafted feature engineering techniques. First, such an approach can provide a lightweight mobile solution to extract geo-aware features in real-time for locations worldwide and second, it would enable a more straightforward fusion between GPS coordinates and features from other modalities (*e.g.*, visual and textual features) in machine learning applications.

Figure 1 illustrates the existing two types of GPS encoding techniques which both generate manually crafted GPS embeddings in small areas of interest. The first type, as shown in Figure 1, is the OneHot encoding approach [4]. This type typically segments the 2D space into grid cells, thus recording into which cell each GPS coordinate falls into. These methods have been widely adopted due to their simplicity. However, they do not encode any semantic information of the locations and only work well for small areas of interest. A difference of one degree in latitude or longitude at the equator equals approximately 110 km. Therefore, segmenting

Y. Yin and R. Zimmermann are with the National University of Singapore (e-mail: idsyin@nus.edu.sg; rogerz@comp.nus.edu.sg).

Y. Zhang is with the Northwestern Polytechnical University, Xi'an, China (e-mail: yingz118@gmail.com).

Y. Yin and Y. Zhang contributed equally to this work.

Z. Liu is with the Zhejiang Gongshang University, China (e-mail: liuzhenguang2008@gmail.com). He is the corresponding author.

S. Wang is currently with the Alibaba Group, Singapore (e-mail: sh.wang@alibaba-inc.com). This work was done while he was at the National University of Singapore.

R. Shah is with the IIIT-Delhi, India (e-mail: rajivratn@iiitd.ac.in).

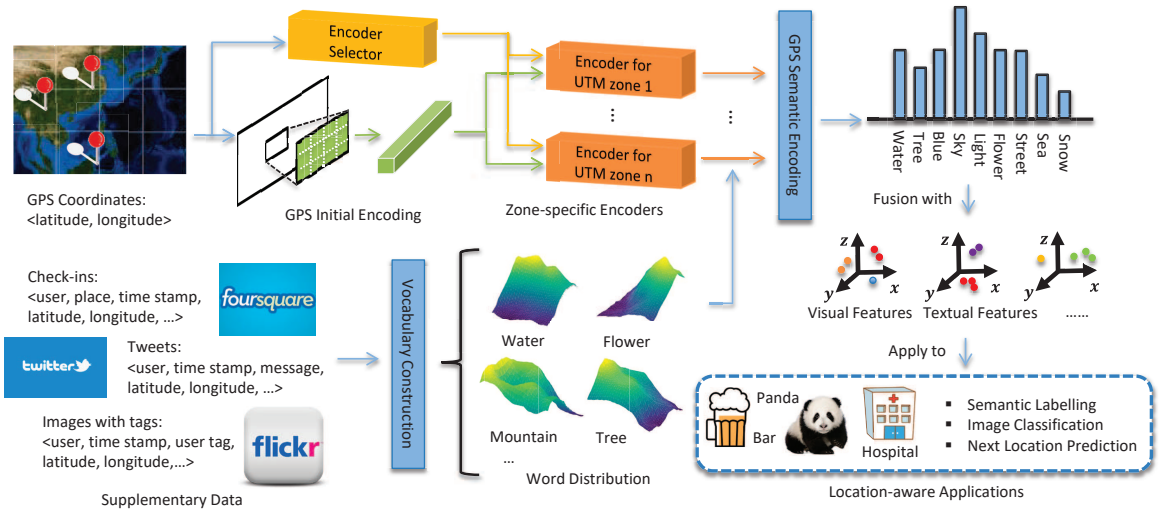


Fig. 3: The system overview of our proposed GPS semantic feature extraction and its use in location-aware applications.

the entire planet with a coarse-grained grid of $0.1° \times 0.1°$ cells (*i.e.*, a cell length at the equator of approximately 11 km) would result in a large 1800×3600 encoding. This would be significantly larger than the common input size of neural networks, *e.g.*, 224×224 pixels for images, and this might exhaust computational resources such as system or GPU memory.

The second type of GPS encoding technique uses supplementary data. Such additional data refers to large-scale datasets of geotagged documents or objects, from which we can extract surrounding information of a given GPS coordinate. For example, Joshi and Luo [5] proposed to utilize GeoNames [6] to retrieve nearby place entities at a specific location. Tang *et al.* [1] proposed to utilize geotagged images to extract both hashtag and visual context features for locations within the United States. To encode a GPS coordinate, these methods perform a nearest neighbor search to obtain a list of nearby objects from the supplementary data, from which a semantic vector is generated as the GPS representation. Though spatial indexing methods such as the KD-tree and the ball-tree [7] can be adopted to perform real-time spatial queries in 2D space, their efficiency declines when handling high dimensional features in the visual space [2]. Moreover, GPS encoding still relies on the supplementary dataset even with spatial indexing. To encode worldwide locations, the supplementary dataset must densely cover the whole world. The costs to maintain such a large-scale dataset are prohibitive. For mobile applications, the system efficiency will be further degraded by network communication delays between mobile devices and a server.

To address these issues we present our novel lightweight GPS2Vec approach which is able to extract geo-aware semantic features in real-time during inference for large-scale areas without dependence on supplementary datasets. As shown in Figure 2, we first propose a two-level soft encoding approach to extend the idea of one hot encoding to work smoothly with worldwide locations. Next, we use the semantic information extracted from supplementary data as labels to train neural networks that map the initial soft encoding (input) to a semantically-enriched encoding (output). We refer to these neural networks as the *SupData*-based semantically-enriched models. These models need to be trained only once for them to capture and summarize the semantic information of the supplementary data. Thus they can be used as a replacement to generate semantically-enriched GPS encodings. Moreover, our proposed models are lightweight and can be deployed in mobile devices, which enables true real-time response even without Internet connection.

Figure 3 illustrates the overall architecture of our system. For our method to handle GPS coordinates worldwide, we introduce a two-level grid-based approach to strike a balance between information loss and computation cost. On the first level, we adopt the Universal Transverse Mercator (UTM) coordinate system to divide the Earth's surface into 60 longitude zones and 20 latitude bands. On the second level, we encode GPS coordinates into grid-based features and train a neural network to learn the semantic embeddings separately in each UTM zone. While one hot encoding is widely used in previous work [1], [4], we present a new soft GPS encoding method that relaxes the requirements on the cell size, and thus is able to generate descriptive encoding features using a grid with fewer cells. As choice for the supplementary data source we can utilize any large-scale geotagged dataset such as tweets, check-ins, or images that are available from social media sharing platforms. Given a specific location, we generate a normalized histogram in its vicinity by calculating the weighted sum of the semantic words in a pre-defined vocabulary. For example, semantic words can be keywords in tweets, user tags associated with Flickr images, or venue types in Foursquare check-ins. Intuitively, we aim to use neural networks to predict the distribution of semantic words at a specific location, which can be a good semantically-enriched representation for GPS coordinates. The key contributions of this work are summarized as follows:

- We present the first machine-learning-based lightweight solution that generates semantically-enriched embeddings

for worldwide GPS coordinates. It achieves real-time responses with minimal computational resources that can be easily satisfied on mobile devices.
- We propose a novel two-level grid-based approach to learn global GPS semantic embeddings. Considering the surface area of the Earth, earlier methods have struggled to achieve a balance between computational cost and information loss during GPS encoding.
- The GPS2Vec models, once trained on supplementary data, can be used as pre-trained models for feature extraction from GPS coordinates. Transfer learning can be applied even when the domain of the supplementary data is different from that of an application.
- Extensive experiments have been conducted on place semantic labeling, image classification, and next location prediction. Our method significantly improves the prediction accuracy, and overall provides new insights into the challenges and opportunities in GPS encoding tasks.

The rest of this paper is organized as follows. Section II reports important related work. Section III introduces our proposed GPS2Vec system framework. Section IV presents the utilization of our GPS semantic embeddings in three location-aware applications. Experimental comparisons with state-of-the-art methods are reported in Section V. Finally, Section VI concludes and suggests future work.

II. Related Work

Geo-clustering and Geo-filtering: With the ubiquity of sensor-equipped cellphones, it is common for multimedia documents such as tweets and images posted on the Internet to be associated with GPS coordinates (*i.e.*, latitude and longitude) [8], [9]. The availability of geo-location context has opened up new opportunities in a variety of applications such as landmark recognition [10]–[12], image classification [1], [2], semantic annotation [13], [14], *etc.* The early utilization of geo-location context associated with documents can be roughly classified into the following two scenarios: geo-clustering and geo-filtering. These methods compute and leverage the geographical distance between documents calculated based on their associated geotags, rather than analysing the geotags directly. For example, Zheng *et al.* [10] proposed to discover landmarks worldwide by first performing geo-clustering. Moxley *et al.* [15] proposed to annotate an image based on its k-nearest geo neighbors. Kleban *et al.* [16] further proposed to weigh the geo neighbors of an image based on their visual similarity.

GPS Encoding without Supplementary Data: Since a GPS coordinate is a tuple of only two values, namely latitude and longitude, its integration with existing high dimensional textual and visual features in geo-based application is difficult. In recent years, several efforts have attempted to encode GPS coordinates at the feature level based on one hot encoding [1], [17]. Tang *et al.* [1] proposed to divide an area of interest (AOI) into 25×25 km^2 square cells and construct an indicator vector that records into which grid cell a GPS coordinate falls. Yao *et al.* [17] further proposed to transform the sparse indicator vector into a dense embedding vector by introducing an embedding layer in their system architecture. However, the number of cells to encode GPS coordinates is always limited by the computational cost and memory. In an extreme example, geotagged documents are spread all over the world [4]. The authors adopted UTM zones for GPS encoding, resulting in significant information loss, as the granularity of UTM is too coarse.

GPS Encoding with Supplementary Data (Regional): When supplementary data sources are available, it is possible to encode GPS coordinates into feature vectors with semantics. However, under many circumstances, such a geotagged supplementary dataset is only available in a limited number of regions around the world [1], [18]. For example, Tang *et al.* [1] proposed to extract geographic map features and ACS features for a given GPS coordinate using Google Maps [19] and American Community Survey (ACS) [20], respectively. Joshi and Luo [5] proposed to encode GPS coordinates by retrieving nearby place entities from GeoNames [6], which is a freely available geographic information system (GIS) database. Recently, Spruyt [21] presented a triplet network to learn a metric space that captures semantic similarity between different geographic location coordinates. Given a location and a radius, they queried their GIS database to obtain a large set of geographic information, and then rasterized it into image tiles for the triplet network training. As mentioned earlier, one major drawback of these approaches lies in their difficulties to generalize to worldwide applications, due to the limited availability of the required supplementary data sources in less populous areas around the world.

GPS Encoding with Supplementary Data (Global): One promising data source to generate worldwide GPS embeddings can be large-scale user generated and geotagged documents that are available online (*e.g.*, check-ins, images, and tweets) [22], [23]. Towards this direction, Liao *et al.* [2] proposed to extract geo-aware tag features by tag propagation from both the geo and visual neighbors of a given geotagged image. Tang *et al.* [1] proposed a hashtag context feature by capturing the distribution of hashtags associated with Instagram images in the vicinity of a given image. However, the aforementioned methods are tailored for geotagged image classification only. Moreover, it is time-consuming to query geo and visual neighbors from a large collection of supplementary images, which significantly hinders the use of such methods in real-time applications.

Image Geolocalization: Image geolocalization aims to predict the geotag of an image either by matching it to a large-scale georeferenced image dataset [24] or by directly classifying it to a pre-defined set of geographic cells [25]. A key component of image geolocalization is to model the distance between the query image and worldwide locations. To obtain a better distance metric, Salem *et al.* proposed to construct a global-scale, dynamic map of visual appearance attributes using geotagged images [26]. The distance can thus be computed by comparing the visual attributes of a query image with the visual attributes of a location predicted by their model. Similarly, though our GPS2Vec model was originally designed for geotag encoding, it has the potential to be utilized to support image geolocalization as well. The distance metric

could possibly be improved by additionally considering the difference between the semantic features of a query image and the semantic features of a location predicted by our model.

Multimodal Feature Fusion: The semantic embedding features extracted from GPS coordinates aim to capture high-level semantics that are complementary to the raw GPS coordinates and the existing textual/visual descriptors. Therefore, multimodal fusion techniques can be applied to obtain performance gains by combining different types of features [27]. To incorporate location features into existing neural networks, a concatenation layer is usually introduced to combine multiple features before passing them to a fusion network for more robust predictions [1]. More advanced fusion techniques, *e.g.*, bilinear CNN models [28] and multiplicative fusion methods [29], have also been proposed recently and significant improvements were reported.

III. Learning Worldwide GPS Semantic Embeddings

We propose a general solution that encodes a GPS coordinate into a semantic descriptor based on a neural network (NN) and is capable of handling locations worldwide rather than focusing on specific small areas of interest. Leveraging neural networks enables the extraction of GPS semantic embeddings in real-time. More importantly, our generated GPS semantic embeddings can easily be applied in a variety of location-aware applications to obtain significant performance gains. Next, we introduce the technical details of our proposed approach in the rest of this section and discuss its use in three location-aware applications in Section IV.

A. GPS Initial Encoding

It is difficult to directly use GPS coordinates as inputs to a neural network. Therefore, we transform the low-dimensional GPS coordinates into high-dimensional distributed vector representations before passing them to our proposed neural network to learn the semantic embeddings [30]. Additionally, none of the existing GPS encoding methods have ever been investigated for or extended to deal with locations worldwide. Thus we present a novel two-level grid based GPS encoding approach. On the first level, we adopt the Universal Transverse Mercator (UTM) coordinate system and divide the Earth into 60 longitude zones and 20 latitude bands. Each UTM zone is referenced by a longitudinal zone number (*i.e.*, 1 to 60) and a latitudinal zone letter (*i.e.*, C to X, omitting O). A location in a zone is represented by the projected easting and northing planar coordinate pair. Considering that the granularity of the UTM zones might be too coarse, we further divide each zone into $m \times m$ grid cells on the second level to perform the initial GPS encoding. Formally, let $Z = \{g_{ij} | i, j = 1, 2, ..., m\}$ denote the set of cells in zone Z. Let $c_{ij} = (x_{ij}, y_{ij})$ denote the center UTM coordinate of cell g_{ij}. Then for any GPS coordinate in the same zone Z, we first represent it by the corresponding UTM coordinate $l = (x, y)$. Next, we compute its initial encoding $E^l = \{e^l_{ij} | i, j = 1, 2, ..., m\}$ as,

$$e^l_{ij} = \exp(-\frac{\|l - c_{ij}\|_2}{\sigma}) \quad (1)$$

where $\|l - c_{ij}\|$ denotes the Euclidean distance between the UTM coordinates l and c_{ij}, and σ represents a constant attenuation coefficient.

Existing grid-based GPS encoding methods mostly use a one-level grid to construct an indicator vector that records which grid cell a GPS coordinate falls into, resulting in a sparse feature vector with only one entry set to 1 [1], [4], [17]. The grid granularity is required to be very fine in order to reduce the information loss as GPS coordinates that fall into the same cell will be assigned the same encoding feature. When it comes to the scale of the entire planet, the use of a single grid level may result in great information loss as the number of cells is always limited by the system's computational resources. Comparatively, our approach processes each UTM zone individually by introducing a second grid level to improve the system's scalability. The advantages of our proposed two-level grid-based GPS encoding method are twofold. First, multiple models can be more efficiently retrieved and updated locally in each UTM zone compared to a single model representing the entire planet. Second, the information loss during GPS encoding is majorly controlled by the cell size of the second-level grid. Therefore, the choice of the first-level grid in our method has more flexibility in terms of its granularity. Here we adopt the UTM coordinate system as the first-level grid, for ease of converting a latitude and longitude pair into a planar coordinate in meters. Other projection models such as the ECEF (Earth-centered Earth-fixed) [31] and Google's S2 geometry [25] can be adopted, but we choose UTM as it is a grid-based projection that is widely used in many geo-based applications [4].

Moreover, our soft encoding approach using Eq. 1 can better discriminate GPS coordinates since the distance $\|l - c_{ij}\|$ is more sensitive to the location change in the corresponding UTM coordinate l. With a proper setting of σ, we are able to generate a dense encoding E^l with few zero entries, leading to effective learning by the neural network presented in Section III-C. Later in the experiments, we will demonstrate that representative semantic embeddings can be extracted by neural networks from the initial GPS encoding generated with a simple 20×20 grid in each UTM zone.

B. Vocabulary-based Semantic Feature

Given GPS coordinates' initial encoding features, we aim to learn GPS semantic embeddings based on neural networks. The labels for training can be automatically generated by extracting semantic contexts from supplementary data sources such as Flickr, Twitter, and Foursquare. Formally, let $V = \{t_1, t_2, ..., t_n\}$ denote a vocabulary consisting of n words. The construction of vocabulary V will be discussed in Section III-D. Our goal is to automatically generate a vocabulary-based feature for a GPS coordinate based on vocabulary V. The resulting n-dimensional feature, denoted as $S^l = \{s^l_i | i = 1, 2, ..., n\}$, will be used as a set of labels for the training of the neural networks.

Let o be a multimedia document in the supplementary dataset, and $l(o)$ and $T(o)$ be the geotag and semantic words associated with sample o, respectively. For example, $T(o)$ can

be the user tags associated with an image, the texts of a tweet, or the venue type in a check-in record where o is an image, a tweet, or a check-in record, respectively. For each multimedia document, we compute its semantic encoding $S(o)$ based on vocabulary V as,

$$s_i(o) = \begin{cases} 1 & t_i \in T(o) \\ 0 & t_i \notin T(o) \end{cases} \tag{2}$$

where $s_i(o)$ is the i-th element in vector $S(o)$. This semantic encoding can be quite sensitive to both GPS noise and semantic keyword uncertainty, and therefore cannot be directly used as the vocabulary-based semantic feature. For instance, images that are geographically close to each other can sometimes have completely different user tags. This can be caused by human mistakes, or more commonly by the difference between the image content location and the camera location. To reduce noise, we smooth the geographic distribution of semantic words by taking the geo neighbors into consideration as well. Given a location l, we first retrieve the k-nearest geo neighbors $NN(l)$ of location l from the geotagged supplementary dataset in terms of the geographic distance. Next, we compute the weighted sum of the semantic encodings in the geo neighborhood,

$$\tilde{s}_i^l = \sum_{o \in NN(l)} w_i^l(o) \cdot s_i(o) \tag{3}$$

and apply l_1 normalization to obtain our vocabulary-based semantic feature S^l

$$s_i^l = \frac{\tilde{s}_i^l}{\sum_j \tilde{s}_j^l} \tag{4}$$

where j iterates over the words in the vocabulary V, and s_i^l represents the i-th element in S^l.

Weight $w_i^l(o)$ is formulated based on the geographic distance between locations l and $l(o)$ as [2],

$$w_i^l(o) = \exp(-\frac{\|l - l(o)\|_2}{\sigma_w}) \tag{5}$$

where $\|l - l(o)\|_2$ computes the Euclidean distance between the UTM coordinates l and $l(o)$, and σ_w is a constant attenuation coefficient.

The generated feature vector S^l captures the distribution of semantic words around location l, which provides rich contextual information about events that occur in the real world. The l_1 normalization is applied to reduce the impact caused by the unbalanced geographic distribution of the geotagged documents.

C. Neural Network Architecture

In each UTM zone we propose to train a neural network, which is able to transform the GPS initial encoding (introduced in Section III-A) into the vocabulary-based semantic feature (introduced in Section III-B). We map the zone-level GPS initial encodings into the shared global semantic embeddings by training separate models in each UTM zone. It is noteworthy that though the GPS coordinates in different UTM zones may have the same initial encodings, the extracted semantic embeddings will be different as they will be processed by different neural networks. Compared with traditional unsupervised methods, the advantages of our proposed machine based solution are twofold. First, our method divides the GPS encoding into two stages: offline training and online extraction. The offline training executes time-consuming nearest neighbor queries from large-scale supplementary datasets, while online extraction generates GPS semantic embeddings in real-time using only pre-trained models. Second, we adopt a relatively simple neural network architecture to prevent overfitting, based on experiments, which makes our solution more robust than unsupervised methods to both GPS noise and semantic keyword uncertainty. Though multiple models are trained to cover the entire Earth's surface, these models can be efficiently retrieved and updated locally in each UTM zone. The maintenance costs of our pre-trained models are much smaller than that of the supplementary dataset. Moreover, the inference of our model is performed simply by passing an input location to a neural network that outputs the semantic embedding, which can be executed highly efficiently in milliseconds.

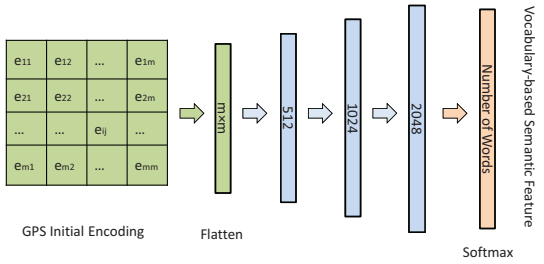


Fig. 4: Illustration of the proposed neural network for GPS semantic encoding.

As shown in Figure 4, we adopt a neural network that consists of three hidden layers and one output layer. A ReLU (Rectified Linear Unit) activation is applied to each hidden layer and a softmax activation is applied at the end to transform the network output into the prediction space. The size of the three hidden layers are set to 512, 1024, and 2048 neurons based on experiments. Algorithm 1 illustrates the preparation of a training dataset from a worldwide supplementary dataset, and the training of the GPS2Vec models in each UTM zone. The size of the output layer equals the size of vocabulary V, whose construction will be discussed in the next section. The input to our neural networks is the GPS initial encoding E^l as introduced in Section III-A. During learning, we aim to use the vocabulary-based semantic features S^l as labels to train our set of neural networks $f(E^l)$ to estimate the normalized word frequency in the vicinity of location l. Let θ be the model parameters to be learned, then the loss function for the network training is given as,

$$L(\theta) = \sum_l D_{KL}(S^l || f(E^l; \theta)) \tag{6}$$

where $D_{KL}(P||Q) = \sum_i P_i \log \frac{P_i}{Q_i}$ represents the Kullback Leibler (KL) divergence. As the semantic feature S^l can be interpreted as a distribution of the semantic words in vocabulary V, the KL divergence, which measures how one probability distribution is different from another, reference probability

Algorithm 1: GPS2Vec Model Training

Input: A worldwide large-scale supplementary dataset *SupData* and a vocabulary V consisting of n semantic words
Output: Pre-trained GPS2Vec models in each UTM zone
List $list_E$, $list_S$;
Dictionary $models$;
for *each* o, $l(o)$, $T(o)$ *in SupData* **do**
  /* o is a multimedia document, $l(o)$ and $T(o)$ are the geo and semantic tags associated with o */
  $l = l(o)$;
  compute the initial GPS encoding E^l using Eq. 1;
  add E^l into $list_E$;
  compute the normalized semantic feature S^l using Eq. 4;
  add S^l into $list_S$;
for *each utm in UTM zones* **do**
  $list'_E, list'_S$=filtering($list_E$,$list_S$);
  /* filter $list_E$ and $list_S$ based on locations */
  train a neural network $model$ as shown in Fig. 4 with $list'_E$ being the input and $list'_S$ being the output;
  $models[utm] = model$
return $models$;

Algorithm 2: Semantic Embedding Extraction

Input: GPS2Vec pre-trained models $models$ and a GPS coordinate l
Output: The semantic embedding of the GPS coordinate l
$model$=select($models$,l);
/* select the correct GPS2Vec model representing the UTM zone that contains location l */
compute the initial GPS encoding E^l using Eq. 1;
return $model(E^l)$;

distribution, can be a good choice for the loss function. During training, we optimize $\arg\min_\theta L(\theta)$ using stochastic mini-batch gradient descent based on back-propagation with momentum. The mini-batch size and the momentum were set to 32 and 0.9, respectively. The learning rate was set to 0.001.

Algorithm 2 illustrates the process of GPS semantic embedding extraction based on our pre-trained GPS2Vec models. Given a GPS coordinate l, we first compute its initial GPS encoding E^l using Eq. 1. Next, we pass E^l to the correct GPS2Vec model, selected according to l. Finally, the model output is returned as the semantic embedding of location l.

D. Vocabulary Construction

Ideally, the vocabulary construction should be application-specific, as applications in different domains may require different words. For example, Flickr, Twitter, Foursquare, Strava, *etc.*, record different user activities and therefore depict a place from inherently different perspectives. In this initial attempt to learn worldwide GPS semantic embeddings, we start by building the vocabulary from one data source. Fortunately, the extension to leverage multiple data sources for embedding learning is straightforward. For example, multiple vocabularies can be built with different data sources and the most appropriate one can be selected based on the application domain. Moreover, the embeddings learnt from one data source (*e.g.*, images) are also helpful when they are applied to other data sources (*e.g.*, tweets or check-ins) based on transfer learning. This is reasonable as correlations exist between words from different domains. For example, a user tag "beer" associated with an image may have a high correlation with venues of type "pub" and "bar", and a tag "food" tends to appear more frequently near "restaurants."

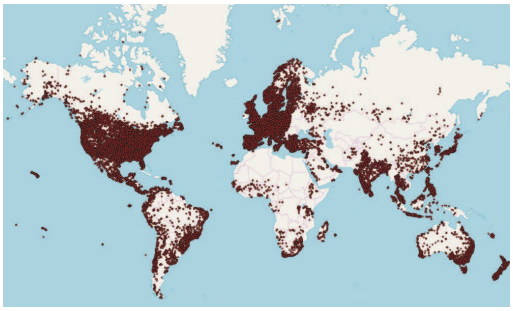


Fig. 5: Geographical distribution of the one million Flickr images.

By considering both the data availability and application popularity, for vocabulary construction and semantic embedding learning we chose geotagged images and leveraged the one million Flickr images collected by Li *et al.* [22], which were taken by 145,000 distinct users in over 100 countries. As seen in Figure 5, this dataset is quite diverse in terms of its geographical distribution. To ensure that high frequency tags are included in the semantic embedding learning, we constructed the vocabulary V by selecting the top 2,000 most frequent tags in the one million Flickr dataset, excluding stop words, camera brands, and non English words [2], [32]. Thus, both the vocabulary-based semantic feature and the output of our neural network have a dimensionality of 2,000. For feature normalization, in addition to the l_1 norm that we adopted in Eq. 4, it would also be possible to consider the tag frequency in the future [1].

One issue of the supplementary dataset that we used in this study might be the uneven distribution of the geotagged images. The GPS embeddings generated in regions with only a few geotagged images can be sparse. Fortunately, when used in applications, the GPS embeddings will be fused with features from other modalities such as the visual features in image classification and the check-in-based features in place annotation. Thus, locations with sparse embeddings can still be distinguished based on multi-modal feature fusion.

IV. Enriching Geo-applications with Semantics

Next we introduce the use of our proposed GPS semantic embeddings not only in image classification, but also in place semantic annotation and in next location prediction with transfer learning.

A. Semantic Annotation of Places

Semantic place annotation refers to the process of assigning a meaningful name to a location. For example, with GPS data from government diary studies, the semantic labels may be "home," "work," and "school," given to geographic locations where a person spends time [14]. With check-in data collected from location-based social networking (LBSN) services such as Foursquare, the semantic labels may be "restaurant," "hotel," and "hospital," given to places of interest (*i.e.*, venues) in LBSN [13]. Such semantic labels are important for people to

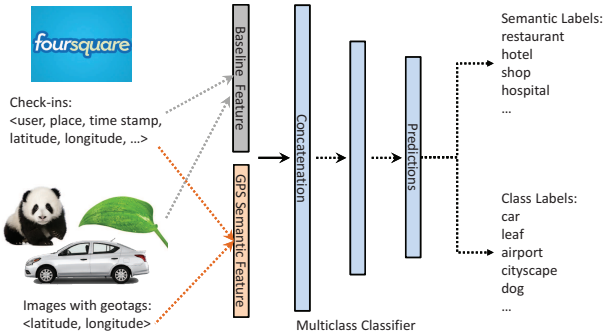


Fig. 6: Integration of our GPS semantic feature with other application-related features in place annotation and image classification.

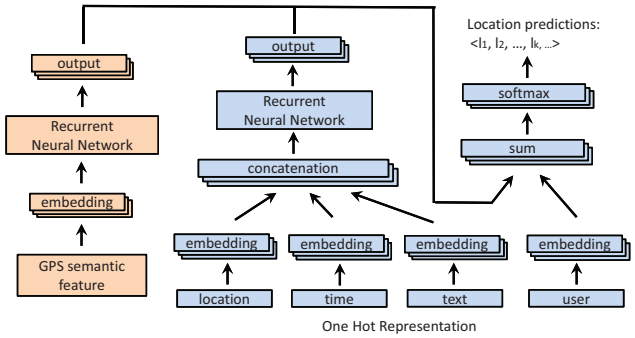


Fig. 7: Integration of our GPS semantic feature in the recurrent model for next location prediction.

infer activities, explore new places, and develop recommendation services [33], [34].

The semantic place annotation can be formulated as a multi-class classification problem based on features generated for each place. For example, in LBSN, users are likely to behave differently at different venues due to the nature of the services and functions offered by these places [13]. Therefore, different behavior patterns of visitors can be extracted from the check-in data at each venue to depict the place. Some baseline features that are proposed in previous work are as follows:

- total number of visits,
- total number of unique visitors,
- maximum number of check-ins by a single visitor,
- distribution of visit time in a week, and
- distribution of visit time in a day.

Such baseline features can be easily integrated with our proposed GPS semantic embeddings that are extracted based on the geo-coordinates of the places to be labeled. As illustrated in Figure 6, one way to achieve this goal is to concatenate different features to form a final feature vector through early fusion. The concatenated feature vector is next passed to a multi-class classifier for label prediction. The advantage of this early fusion strategy not only lies in its effectiveness, but also its generalization capability to any geo-aware classification problem with little modifications required.

B. Image Classification with Location Context

Nowadays it is common for online images to be associated with GPS coordinates, due to the availability of sensor-equipped smartphones. Such geotags can be leveraged to obtain rich contextual information that is of great importance to help predict what is captured in an image. Though several efforts have been made in this direction, drawbacks exist in some methods, *e.g.*, tailoring for specific AOIs [1], GPS encodings that are computationally expensive [2], and significant information loss with coarse-grained encodings [4]. Our work differs in that we are interested in efficient and fine-grained GPS semantic embeddings extraction that is not tailored to particular tasks or AOIs. Our method can be applied to a wide range of geo-aware applications with sufficient performance gains, in terms of both effectiveness and efficiency, achieved.

The use of our GPS semantic embeddings in image classification is illustrated in Figure 6. In this case, the baseline features refer to image visual features varying from hand-crafted low-level features such as HOG [35] and SIFT [36] to the most recent CNN-based deep features for object and scene recognition [37], [38]. Moreover, the class labels can cover a wide range of concepts without limitations to particular types of objects and scenes. For the classifier, we adopt a neural network with one hidden layer of 512 units and one output layer for both semantic annotation of places and image classification. Other classifiers such as SVM [39] would also suffice.

C. Next Location Prediction in Semantic Trajectories

Predicting the next location that a user tends to visit is a challenging and crucial task for applications such as tour recommendations and traffic planning [40]. For example, given a user's historical check-in data, we may be able to predict where the user will go next by jointly analyzing multiple factors including time, location, texts, and user preferences [17]. Recently, Yao *et al.* [17] proposed a unified framework, termed SERM, which is capable of jointly learning the embeddings of the aforementioned multi-factors and the transition parameters of a recurrent neural network (RNN) for next location prediction. The architecture of their proposed semantics-enriched recurrent model (SERM) is illustrated in Figure 7. First, one hot representations are generated for all inputs, where the category of each factor is defined as follows:

- **Time:** one hot vector generated based on timestamp by discretizing one week's time into 48 equal sized time slots
- **Location:** one hot vector generated based on location by discretizing the AOI into a $500\ \text{m} \times 500\ \text{m}$ grid
- **Text:** one hot vector generated based on bag-of-keywords
- **User:** one hot vector generated based on user id

Next, embedding vectors are generated and concatenated from one hot representations of time, location, and text to train a recurrent neural network, and fused with the user embedding that describes the user preferences, at the RNN's output layer. To make use of pre-trained models and reduce RNN training time, we propose to first train an individual RNN with the same architecture based on our GPS semantic embeddings, and then fuse its output with pre-trained SERM for fine-tuning as shown

TABLE I: Venue categories and their percentages (z%) in the evaluation dataset.

Category	z%	Category	z%
Restaurant & Food	40.4%	Shop	19.8%
Hotel	3.5%	Pub & Bar	13.1%
Store	20.7%	Hospital	2.5%

in Figure 7. Finally, we derive the probability distribution over all the locations at each step by applying the softmax function. In addition to CNN-based models such as the aforementioned image classification, this application demonstrates the use of our proposed GPS semantic embeddings in RNN-based models, and its effectiveness in sequential data modeling.

V. Experiments

We first evaluate the effectiveness of our proposed GPS semantic embeddings learning method in three location-aware applications. The venue semantic annotation and the geotagged image classification are global applications based on world-wide Foursquare check-in records and Flickr images, respectively. The next location prediction is a regional application based on city-scale semantic trajectory data (*i.e.*, check-ins and tweets) in New York and Los Angeles. We evaluate our proposed approach on both global and regional applications to verify its effectiveness. Throughout the experiments, the parameter m that controls the grid size in each UTM zone is set to 20. The attenuation coefficient σ is set to 20 km. The number of geo neighbors k is set to 150 as suggested by Liao *et al.* [2]. We adopt the output of the softmax layer in our proposed neural network as the generated GPS semantic embedding. Next, we perform an ablation analysis on the model parameter settings and discuss their impact in our proposed framework.

A. Evaluation on Venue Semantic Annotation

1) Experimental Setup: We conduct experiments on a global-scale check-in dataset collected from Foursquare [41], [42]. Each check-in records the user id, venue id, visit time, latitude, longitude, and venue category name. We choose six popular categories as shown in Table I to form an evaluation dataset. The semantic labeling of places can be formulated as a multi-label classification problem. For example, a place associated with tag "restaurant" can also be annotated with "bar." Therefore, we apply sigmoid activation at the output layer, and choose binary cross-entropy as the loss function. The network was trained using stochastic mini-batch gradient descent based on back-propagation. The learning rate and mini-batch size were set to 0.001 and 32, respectively.

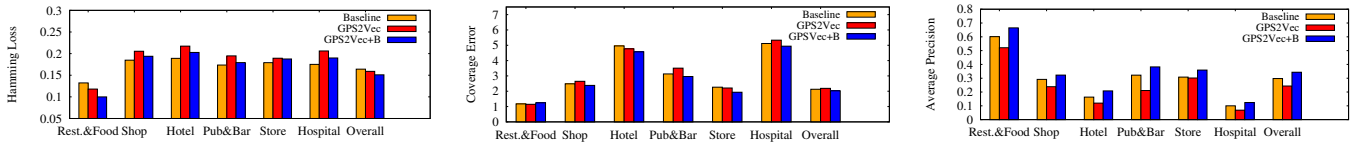
2) Performance Comparison: We adopt the following three metrics: hamming loss, coverage error, and average precision as the evaluation metrics. The hamming loss measures the fraction of labels that are incorrectly predicted, while the other two metrics measure the ranking performance of the predictors [43]. The coverage error evaluates how far we need to travers the ranked list of predicted tags in order to cover all the ground-truth tags associated with the venue. The average precision is a measure that combines recall and precision for ranked retrieval venues given a tag, and is computed as the mean of the precision scores after each ground-truth venue is retrieved. We extract the five baseline features as introduced in Section IV-A, and concatenate them into a feature vector, which we refer to as the baseline feature.

TABLE II: Performance comparison on place semantic labeling based on GPS embedding, baseline feature (B), and their fusion.

Method	Hamming loss	Coverage error	Average precision
Baseline [13]	0.159	2.129	0.298
OneHot [4]	0.165	2.235	0.210
HashTag [1]	0.164	2.200	0.249
GPS2Vec$_{onehot}$	0.165	2.224	0.220
GPS2Vec	0.164	2.189	0.243
OneHot [4] + B	0.152	2.045	0.337
HashTag [1] + B	0.153	2.060	0.335
GPS2Vec$_{onehot}$ + B	0.151	2.034	0.340
GPS2Vec + B	**0.151**	**2.029**	**0.344**

We report the overall hamming loss, coverage error, and average precision in Table II, with the **best** result highlighted in bold, and the place semantic labeling results per venue category in Figure 8. We compared our GPS2Vec to the OneHot encoding [4] and the GPS2Vec$_{onehot}$ which is a variation of GPS2Vec by using the one hot encoding instead of our proposed soft encoding approach in generating the initial GPS encoding. As is illustrated, GPS2Vec outperforms OneHot and GPS2Vec$_{onehot}$ by 3.3% and 2.3% in terms of the average precision, respectively. Please note that the GPS2Vec embeddings are learnt from geotagged images. By being applied to an application in a different domain (*i.e.*, check-in records), our proposed GPS2Vec+B is still able to achieve the best classification result, outperforming Baseline and OneHot+B by 4.6% and 0.7% in terms of the average precision. This indicates that the GPS2Vec embeddings learnt from one data source have the potential to work reasonably well with applications in different domains. Generally speaking, the baseline features depict different user behavior patterns at different places in terms of population and temporal duration. The vocabulary-based GPS2Vec embeddings, on the other hand, provide contextual information about events that occur in the real world at each place. The distribution of user tags in the vicinity can be closely related to the place category. For example, the tag "beer" can have a high correlation with places such as pubs and bars, and the tag "food" tends to appear more frequently near restaurants.

Next, we compare our proposed method to the HashTag context feature proposed by Tang *et al.* [1]. Given a GPS coordinate to be encoded, the authors define a set of radii R, and for each $r \in R$, they pool over a circle of radius r around that GPS coordinate and count the number of images tagged with tag t that falls within the radius. We normalize the generated feature vector and set $R = \{1,000, 2,000, ..., 10,000\}$ as described in their original paper. To perform a fair comparison, we generated a 2,000-D feature by setting the tag collection to be the top 50 most frequent tags in our one million Flickr dataset. As shown in Table II, GPS2Vec+B outperforms HashTag+B by 0.9%. Moreover, the computation of the HashTag context



(a) Hamming loss (b) Coverage error (c) Average precision

Fig. 8: Performance comparison per concept on place semantic labeling based on GPS embedding, baseline feature, and their fusion.

feature relies on the supplementary Flickr image dataset, while our proposed method generates GPS semantic embeddings based on pre-trained models during inference without relying on any large-scale supplementary dataset.

B. Evaluation on Image Classification

1) Experimental Setup: For image classification, we evaluate our method using the NUS-WIDE dataset [44], which is a benchmark dataset that is widely used for image annotation and classification. As the location context is required in our experiments, we use the geotagged images in NUS-WIDE and form a training set with 41,173 images and a test set with 27,401 images. Ignoring rare concepts, we test on 75 concepts covering objects, scenes, and events. In terms of the visual feature, we adopt the BovW representation based on SIFT descriptors that is used in previous work [2] to make it a fair comparison.

Since an image is mostly associated with more than one tag, we formulate this problem as a multi-label classification, apply sigmoid activation at the output layer, and choose binary cross-entropy as the loss function. The network was trained using stochastic mini-batch gradient descent based on back-propagation. The learning rate and mini-batch size were set to 0.001 and 32, respectively. We report the Average Precision (AP) per concept, and the mean Average Precision (mAP) as the evaluation criteria.

2) Performance Comparison: We first compare our proposed method to the state-of-the-art geo-based and fusion-based image classification systems in Table III with the **best** and second best results highlighted. Our GPS2Vec obtained the best mAP among the geo-based methods. Moreover, the GPS2Vec+V outperforms the OneHot+V, HashTag+V, and GPS2Vec$_{onehot}$+V by 6.2%, 3.9%, and 2.3%, respectively. The results verify the effectiveness of our proposed GPS2Vec embeddings and its complementarity to image visual features. When comparing to other fusion approaches, our method achieves significant improvements over the approaches that utilize the GPS coordinates in a traditional manner for geo neighbor search [8], [16], [22]. Wang *et al.* [32] and Liao *et al.* [2] both proposed to fuse a visual classifier with a textual classifier built upon tag features generated by conjunctively considering geo and visual neighbors from a supplementary image dataset. The method proposed by Liao *et al.* [2] was able to achieve the best classification result due to the following reasons. First, this method searches for visual neighbors of test images, which is tailored for image classification and cannot be applied to other geo-aware applications. Second, the authors leverage a much larger supplementary dataset that consists of 10 million geotagged images to generate a more descriptive tag-based feature. Comparatively, our method is more general, and at the same time, is able to obtain the second best mean average precision of 0.3 in image classification. Moreover, our method moves the time-consuming nearest neighbor queries to the offline training stage, in order to achieve a real-time response in the testing stage of extracting semantic embeddings from GPS coordinates.

TABLE III: Comparison with the state-of-the-art location-aware image classification approaches.

Method	Classifier	mAP
Visual	visual	0.234
OneHot [4]	geo	0.066
HashTag [1]	geo	0.163
GPS2Vec$_{onehot}$	geo	0.130
GPS2Vec	geo	0.182
OneHot [4] + V	fusion	0.238
HashTag [1] + V	fusion	0.261
GPS2Vec$_{onehot}$ + V	fusion	0.277
GPS2Vec + V	fusion	<u>0.300</u>
Kleban *et al.* [16]	fusion	0.080
Qian *et al.* [8]	fusion	0.113
Li *et al.* [22]	fusion	0.251
Wang *et al.* [32]	fusion	0.236
Liao *et al.* [2]	fusion	**0.347**

TABLE IV: Mean average precision comparison per UTM zone on image classification based on GPS embedding, visual feature, and their fusion.

UTM Zones	30U	31U	18T	10S	32T
Visual	0.254	0.238	0.256	0.283	0.291
GPS2Vec	0.166	0.189	0.156	0.197	0.174
GPS2Vec + V	0.293	0.282	0.278	0.351	0.322

Next, we compare the average precision per UTM zone and per concept obtained by different classifiers trained with GPS semantic embedding (GPS2Vec), image visual feature (Visual), and their fusion (GPS2Vec+V). The results are illustrated in Table IV and Figure 9, respectively. We select the top five most popular UTM zones based on the number of testing images each zone contains. Some UTM zones contain areas of only one country (*e.g.*, 10S is within the United States), while others contain more than one (*e.g.*, 32T overlaps with multiple European countries including Italy, Switzerland, and France). By fusing our proposed GPS2Vec embeddings with image visual features, a consistent performance gain was observed among different UTM zones. Similarly in Figure 9, the fusion method achieves the best average precision with most of the concepts. The geo method outperformed the visual method in

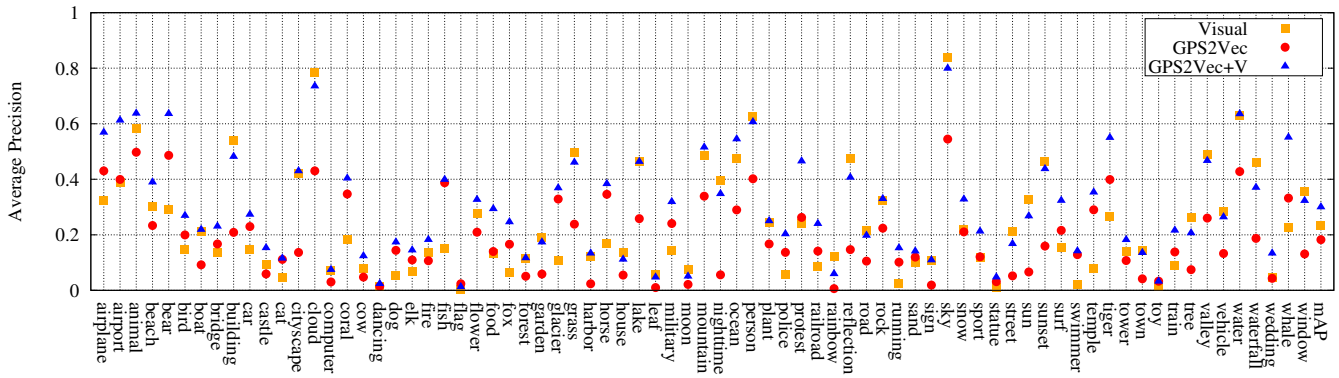


Fig. 9: The average precision comparison per concept on image classification based on GPS embedding, visual feature, and their fusion.

TABLE V: Comparison to the state-of-the-art next location prediction methods on the New York dataset.

Method	HR@1	HR@5	HR@10	HR@20	δ_d/m
NL	0.1630	0.2455	0.2998	0.4386	2903
MF [45]	0.1690	0.4326	0.5013	0.5358	1963
HMM [46]	0.1763	0.4298	0.5251	0.5518	1952
ST-RNN [40]	0.1942	0.4421	0.5381	0.6053	1602
SERM [17]	0.2535	0.4507	0.5433	0.6237	1457
HashTag + S	0.2575	0.4889	0.6036	**0.6861**	1253
GPS2Vec + S	**0.2716**	**0.5131**	**0.6097**	0.6841	**1193**

TABLE VI: Comparison to the state-of-the-art next location prediction methods on the Los Angeles dataset.

Method	HR@1	HR@5	HR@10	HR@20	δ_d/m
NL	0.3745	0.4516	0.4704	0.4911	6061
MF [45]	0.3646	0.5810	0.6354	0.6877	2647
HMM [46]	0.3921	0.5935	0.6331	0.6732	2521
ST-RNN [40]	0.4311	0.6013	0.6521	0.6980	2384
SERM [17]	0.4625	0.6265	0.6670	0.7026	2177
HashTag + S	0.4625	0.6343	0.6709	**0.7055**	2154
GPS2Vec + S	**0.4654**	**0.6344**	**0.6729**	0.7036	**2135**

concepts such as "coral," "whale," "surf," and "temple," that are sensitive to locations. For example, "coral," "whale," and "surf" can have a strong correlation with seas and oceans. On the other hand, the visual method outperforms the geo method in concepts such as "sun," "cloud," "sky," "grass," and "tree" that can be easily recognized based on their visual appearance.

C. Evaluation on Next Location Prediction

1) Experimental Setup: For next location prediction, we conducted experiments on two semantic trajectory datasets collected in two cities, namely New York City and Los Angeles. Records are segmented into semantic trajectories with a time gap constraint of $\Delta t = 10\ h$. Users with fewer than 50 records and trajectories with a length shorter than three are removed [17]. After data preprocessing, the New York dataset [47] consists of 3,103 trajectories from 235 users and the Los Angeles dataset [48] consists of 7,826 trajectories from 244 users. We apply softmax activation at the output layer, and choose categorical cross-entropy as the loss function. The two evaluation datasets are divided into 80%–20% splits for training and testing. Two metrics are adopted for method comparison: 1) the hit ratio at k, which examines whether the ground-truth location appears in the top-k predictions, and 2) the prediction error in the distance, which computes the minimum geographical distance from the ground-truth location to the top-5 predictions. We train the network architecture using stochastic mini-batch gradient descent based on back-propagation. The learning rate and mini-batch size were set to 0.001 and 200, respectively.

2) Performance Comparison: We compare our method to SERM, HashTag+S, and four other next location prediction methods. HashTag+S is a variation of our method GPS2Vec+S by using the HashTag context feature [1] instead of our proposed GPS2Vec feature. Here "S" refers to method SERM as we integrate the proposed GPS2Vec feature into the original SERM architecture. The results are reported in Tables V and VI, with the **best** results highlighted. The NL (Nearest Location) method chooses the nearest neighbor to the user's current location as the prediction. We show it as a straightforward baseline method for comparison.

As can be seen, our method outperforms its competitors on both datasets. The MF method casts the location prediction as a recommendation problem without modeling the transition of the sequential input. The HMM method considers the sequential transition but it only models the first-order dependency of the semantic trajectories. Comparatively, RNN-based methods, ST-RNN and SERM, are capable of capturing long-term dependencies of people's movements, and therefore turn out to be the strongest baseline methods. SERM outperforms ST-RNN as it captures the spatiotemporal dynamics by jointly learning the embeddings of location and time. Additionally, SERM leverages the textual information such as tweets in its modeling, while ST-RNN only focuses on the next location prediction for GPS trajectories without a user's textual messages. Next, we compare GPS2Vec and HashTag features in the next location prediction. As Tables V and VI show, GPS2Vec+S outperforms HashTag+S on both datasets, which indicates that the GPS semantic embeddings generated by

our pre-trained models are more descriptive than the HashTag context features generated from the supplementary data.

By integrating the GPS semantic embeddings, our method improves the SERM method in the following two aspects. First, the vocabulary-based semantic feature extracted from GPS makes it possible to measure the high-level similarity between places, which is crucial for the next location prediction problem. For example, people tend to frequent similar venues due to personal preferences. This may be one of the reasons why our method obtained more significant improvements with the New York check-in dataset compared to the Los Angeles tweets dataset. Compared with tweets, locations of check-ins more likely correlate with the semantic similarities of places. Second, the SERM method encodes locations into one hot representations by discretizing the area of interest into 500 m $\times$ 500 m grid cells. Comparatively, our GPS encoding method is able to generate a more generalized and semantically-enriched representation that can be applied to locations anywhere in the world.

D. Parameter Tuning

We first study the impact of the grid size controlled by parameter m for the GPS initial encoding. As mentioned before, each UTM zone is divided into an $m \times m$ grid. Let m be $\{10, 20, 30\}$, respectively. We report the mean average precision on venue semantic labeling and image classification in Table VII with the **best** result highlighted in each row. The GPS2Vec method uses our GPS semantic embeddings as the only feature. The GPS2Vec+B method concatenates the GPS semantic embeddings and the base line features extracted from the check-in data, while the GPS2Vec+V method concatenates the GPS semantic embeddings and the image visual features to train a classifier. As Table VII shows, GPS2Vec obtains the best mean average precision with $m = 30$ on both venue semantic labeling and image classification. Generally speaking, the GPS semantic embeddings tend to be more descriptive with a smaller grid cell size, *i.e.*, larger m. This is because a grid with fine granularity has a better resolution to discriminate GPS coordinates that are located close to each other. However, with the number of grid cells increasing, the input size of the neural network will rise quadratically, leading to high computational costs and delay. On the other hand, GPS2Vec+B and GPS2Vec+V obtain competitive classification results with $m = 20$ and 30, which indicates that the GPS2Vec embeddings and the baseline/visual features are complementary to each other so that the influence of the grid cell size on the system effectiveness has been reduced. We consider that $m = 20$ can be a good trade-off, and use this as the optimal setting in the rest of the experiments.

Next, we study the impact of the attenuation coefficient in Eq. 1 on the classification results in venue semantic labeling and image classification. We set σ to 10 km, 20 km, and 30 km, respectively, and report the mean average precision in Table VIII with the **best** result highlighted in each row. As can be seen, all three methods obtain their best mean average precision with $\sigma = 20$ km. The calculation of the GPS initial encoding is parameterized together by m and σ. With a moderate-sized grid where $m = 20$, a small change in σ is unlikely to have a significant impact on the classification results. Therefore, we set σ to 20 km throughout the experiments.

The results shown indicate that our proposed method has the advantage of not being sensitive to the parameters m and σ within a large range. This is a good property as the parameters of our proposed method can be effectively tuned when being applied to different applications.

E. Ablation Analysis

1) Comparison of layers: We analyze the discriminative performance of each layer in our proposed neural network. In the previous experiments, we adopted the output of the softmax layer in our proposed neural network as the GPS semantic embedding. However, more generally, the output of intermediate layers can also be used as GPS semantic embeddings in downstream applications. We compare the GPS semantic embeddings extracted from different layers and report the mean average precision in Table IX. As illustrated in Figure 4, the feature dimensions extracted from fc1, fc2, fc3, and output layers are 512, 1,024, 2,048, and 2,000, respectively. For all three methods, the mean average precision increases when using features extracted from layers closer towards the output. Therefore, we adopt the features extracted from the output layer, which tend to encode more semantic information, as the GPS semantic embedding in our proposed framework.

2) Comparison of first-level grid: In our current work, we have adopted the UTM coordinate system as the first-level grid, due to its popularity and the ease of converting a latitude and longitude pair into a planar coordinate in meters. We now compare it to a density-based grid partitioning method using Google's S2 geometry library [25]. More specifically, starting at the root S2 cell, we recursively descend each quad-tree and subdivide cells until no cell contains more than 15,000 images. Subsequently, sparsely populated areas are covered by larger cells and densely populated areas are covered by finer cells. We further divide each cell into a 16×16 second-level grid[1] for the GPS initial encoding. The results are reported in Table X. Recall that m controls the granularity of the second-level grid. Therefore, $m = 16$ in Google's S2 cells causes a slight mAP decrease compared to $m = 20$ in UTM zones (please refer to Table VII). Google's S2 geometry has the advantage of partitioning the Earth's surface into fewer cells in a balanced way. However, the UTM grid can perform competitively well in terms of accuracy as long as the model capacity is sufficient to capture the data characteristics in each zone.

3) Comparison of training sample distribution: In the training phase of our proposed GPS2Vec method, we use the geotags of the images in the supplementary dataset as the training locations in each UTM zone. Table XI shows the comparison of our sampling strategy to a uniform sampling strategy where the training samples are evenly distributed in

[1]As Google's S2 follows a quadtree partitioning, a second-level grid of 16×16 is the closest to the one of 20×20 used in each UTM zone.

TABLE VII: Mean average precision comparison on venue semantic labeling and image classification based on GPS semantic embeddings generated with different grid sizes.

Task	Method	10×10	20×20	30×30
Venue Semantic Labeling	GPS2Vec	0.2349	0.2435	**0.2441**
	GPS2Vec+B	0.3399	**0.3435**	0.3431
Image Classification	GPS2Vec	0.1754	0.1824	**0.1853**
	GPS2Vec+V	0.2971	**0.2999**	0.2987

TABLE VIII: Mean average precision comparison on venue semantic labeling and image classification based on GPS semantic embeddings generated with different σ settings in Eq. 1.

Task	Method	10 km	20 km	30 km
Venue Semantic Labeling	GPS2Vec	0.2422	**0.2435**	0.2405
	GPS2Vec+B	0.3408	**0.3435**	0.3415
Image Classification	GPS2Vec	0.1779	**0.1824**	0.1747
	GPS2Vec+V	0.2947	**0.2999**	0.2986

TABLE IX: Mean average precision comparison of extracting GPS embeddings at different layers on venue semantic labeling and image classification.

Task	Method	fc1	fc2	fc3	output
Venue Semantic Labeling	GPS2Vec	0.2319	0.2332	0.2409	**0.2435**
	GPS2Vec+B	0.3336	0.3341	0.3397	**0.3435**
Image Classification	GPS2Vec	0.1708	0.1718	0.1762	**0.1824**
	GPS2Vec+V	0.2853	0.2950	0.2960	**0.2999**

TABLE X: GPS2Vec mAP comparison of first-level grid choices: UTM zones vs. Google's S2 cells.

Task	UTM	Google's S2
Venue Semantic Labeling	0.243	0.238
Image Classification	0.182	0.179

TABLE XI: GPS2Vec mAP comparison of training sample distribution: data-driven vs. uniform.

Task	data-driven	uniform
Venue Semantic Labeling	0.243	0.229
Image Classification	0.182	0.149

TABLE XII: GPS2Vec mAP comparison of distance metric: Euclidean vs. Haversine.

Task	Euclidean	Haversine
Venue Semantic Labeling	0.243	0.243
Image Classification	0.182	0.185

TABLE XIII: GPS2Vec mAP comparison of supervision: supervised vs. unsupervised.

Task	supervised	unsupervised
Venue Semantic Labeling	0.243	0.247
Image Classification	0.182	0.161

each UTM zone[2]. As can be seen, our data-driven sampling outperforms the uniform sampling by a significant margin. The vocabulary-based semantic features generated at geotag locations can be more accurate as they are directly associated with user tags. Moreover, the image distribution in the supplementary dataset tends to be consistent with the distribution of the testing images. Thus, an improved mAP can be obtained by paying more attention to densely populated areas.

4) Comparison of distance metric: We use the Euclidean distance in this work as the UTM coordinate system naturally converts a latitude and longitude pair into a planar coordinate in meters. Table XII shows the mAP comparison by using the alternative Haversian distance. On one hand, the calculation of the Euclidean distance is more efficient. On the other hand, the Haversian distance is more accurate when computing the distance between two geo-coordinates on the spherical Earth's surface. Moreover, the Euclidean distance can only be used within UTM zones, while the Haversian distance should be adopted together with other projection models such as the Google's S2 cells.

[2]The training locations are obtained by dividing each UTM zone into 1000×1000 cells.

5) Comparison of supervision: Finally, we compare our proposed supervised method with existing unsupervised geotag encoding approaches. For example, the vocabulary-based semantic feature introduced in Section III-B can be used directly as an unsupervised geotag encoding approach. The advantages of the supervised method are twofold. First, unsupervised methods require access to the supplementary dataset during inference, leading to potential computational costs and processing delays. Second, as Table XIII shows, the supervised method outperforms the unsupervised method as the latter requires hand-crafted features that are sensitive to data noise. Comparatively, our supervised method generates learnt features that are more smooth and robust to data noise.

VI. Conclusions and Future Work

We have presented a novel framework, GPS2Vec, to learn GPS semantic embeddings in support of location-aware applications. The generated semantic embeddings can be easily integrated with existing high dimensional descriptors, *e.g.*, image visual features, by early fusion, based on which a new classifier can be trained to obtain more robust predictions. To divide the Earth's surface into smaller areas of manageable scale, we adopt the UTM coordinate system and train a neural

network for each UTM zone to generate location semantic embeddings. We have conducted extensive experiments using tweets, check-ins, and images with three location-aware applications. Our generated GPS semantic embeddings are complementary to the textual and visual features in existing systems. The location-aware applications demonstrate the effective use of our proposed GPS semantic embeddings in both CNN- and RNN-based systems. In the future, we plan to increase the data scale by utilizing different types of data sources for vocabulary construction and semantic embeddings learning. If the accuracy level of a geotag is available, it is also possible to filter out noisy samples to generate more accurate semantic features for GPS coordinates.

ACKNOWLEDGMENT

This research is supported by Singapore Ministry of Education Academic Research Fund Tier 2 under MOE's official grant number MOE2018-T2-1-103.

REFERENCES

[1] K. Tang, M. Paluri, L. Fei-Fei, R. Fergus, and L. Bourdev, "Improving Image Classification with Location Context," in *IEEE International Conference on Computer Vision*, 2015, pp. 1008–1016.

[2] S. Liao, X. Li, H. T. Shen, Y. Yang, and X. Du, "Tag Features for Geo-Aware Image Classification," *IEEE Transactions on Multimedia*, vol. 17, no. 7, pp. 1058–1067, 2015.

[3] Y. Zhang and R. Zimmermann, "Efficient Summarization From Multiple Georeferenced User-Generated Videos," *IEEE Transactions on Multimedia*, vol. 18, no. 3, pp. 418–431, 2016.

[4] G. Christie, N. Fendley, J. Wilson, and R. Mukherjee, "Functional Map of the World," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[5] D. Joshi and J. Luo, "Inferring Generic Activities and Events from Image Content and Bags of Geo-tags," in *International Conference on Content-based Image and Video Retrieval*, 2008, pp. 37–46.

[6] GeoNames, http://www.geonames.org/.

[7] M. Dolatshah, A. Hadian, and B. Minaei-Bidgoli, "Ball*-tree: Efficient Spatial Indexing for Constrained Nearest-Neighbor Search in Metric Spaces," *arXiv preprint arXiv:1511.00628*, 2015.

[8] X. Qian, X. Liu, C. Zheng, Y. Du, and X. Hou, "Tagging Photos Using Users' Vocabularies," *Neurocomputing*, pp. 144–153, 2013.

[9] A. Zubiaga, A. Voss, R. Procter, M. Liakata, B. Wang, and A. Tsakalidis, "Towards Real-Time, Country-Level Location Classification of Worldwide Tweets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 9, pp. 2053–2066, 2017.

[10] Y. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua, and H. Neven, "Tour the World: Building a Web-scale Landmark Recognition Engine," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1085–1092.

[11] Y. Yin, B. Seo, and R. Zimmermann, "Content vs. Context: Visual and Geographic Information Use in Video Landmark Retrieval," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 11, no. 3, pp. 39:1–39:21, 2015.

[12] Y. Yin, Z. Liu, and R. Zimmermann, "Geographic Information Use in Weakly-supervised Deep Learning for Landmark Recognition," in *IEEE International Conference on Multimedia and Expo*, 2017, pp. 1015–1020.

[13] M. Ye, D. Shou, W.-C. Lee, P. Yin, and K. Janowicz, "On the Semantic Annotation of Places in Location-based Social Networks," in *ACM International Conference on Knowledge Discovery and Data Mining*, 2011, pp. 520–528.

[14] J. Krumm and D. Rouhana, "Placer: Semantic Place Labels from Diary Data," in *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2013, pp. 163–172.

[15] E. Moxley, J. Kleban, and B. Manjunath, "SpiritTagger: A Geo-Aware Tag Suggestion Tool Mined from Flickr," in *ACM International Conference on Multimedia Information Retrieval*, 2008, pp. 24–30.

[16] J. Kleban, E. Moxley, J. Xu, and B. S. Manjunath, "Global Annotation on Georeferenced Photographs," in *ACM International Conference on Image and Video Retrieval*, 2009, pp. 12:1–12:8.

[17] D. Yao, C. Zhang, J. Huang, and J. Bi, "SERM: A Recurrent Model for Next Location Prediction in Semantic Trajectories," in *ACM International Conference on Information and Knowledge Management*, 2017, pp. 2411–2414.

[18] B. Yan, K. Janowicz, G. Mai, and S. Gao, "From ITDL to Place2Vec: Reasoning About Place Type Similarity and Relatedness by Learning Embeddings From Augmented Spatial Contexts," in *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2017, pp. 35:1–35:10.

[19] Google Maps, https://maps.google.com/.

[20] American Community Survey, http://www.census.gov/acs/www/.

[21] V. Spruyt, "Loc2Vec: Learning Location Embeddings with Triplet-loss Networks," https://www.sentiance.com/2018/05/03/loc2vec-learning-location-embeddings-w-triplet-loss-networks/, 2018.

[22] X. Li, C. G. M. Snoek, M. Worring, and A. W. M. Smeulders, "Fusing Concept Detection and Geo Context for Visual Search," in *ACM International Conference on Multimedia Retrieval*, 2012, pp. 4:1–4:8.

[23] H. Mousselly-Sergieh, D. Watzinger, B. Huber, M. Döller, E. Egyed-Zsigmond, and H. Kosch, "World-wide Scale Geotagged Image Dataset for Automatic Image Annotation and Reverse Geotagging," in *ACM Multimedia Systems Conference*, 2014, pp. 47–52.

[24] S. Workman, R. Souvenir, and N. Jacobs, "Wide-area Image Geolocalization with Aerial Reference Imagery," in *ICCV*, 2015, pp. 3961–3969.

[25] T. Weyand, I. Kostrikov, and J. Philbin, "Planet-photo Geolocation with Convolutional Neural Networks," in *ECCV*, 2016, pp. 37–55.

[26] T. Salem, S. Workman, and N. Jacobs, "Learning a Dynamic Map of Visual Appearance," in *CVPR*, 2020, pp. 12 435–12 444.

[27] R. R. Shah, Y. Yu, A. Verma, S. Tang, A. D. Shaikh, and R. Zimmermann, "Leveraging Multimodal Information for Event Summarization and Concept-level Sentiment Analysis," *Knowledge-Based Systems*, vol. 108, no. C, pp. 102–109, 2016.

[28] T. Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN Models for Fine-Grained Visual Recognition," in *IEEE International Conference on Computer Vision*, 2015, pp. 1449–1457.

[29] E. Park, X. Han, T. L. Berg, and A. C. Berg, "Combining Multiple Sources of Knowledge in Deep CNNs for Action Recognition," in *IEEE Winter Conference on Applications of Computer Vision*, 2016, pp. 1–8.

[30] Y. Yin, Z. Liu, Y. Zhang, S. Wang, R. R. Shah, and R. Zimmermann, "GPS2Vec: Towards Generating Worldwide GPS Embeddings," in *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2019, pp. 416—-419.

[31] W. Dongfang, P. Baojun, X. Weike, and P. Keke, "GEO Objects Spatial Density and Collision Probability in the Earth-centered Earth-fixed (ECEF) Coordinate System," *Acta Astronautica*, vol. 118, pp. 218–223, 2016.

[32] G. Wang, D. Hoiem, and D. Forsyth, "Building Text Features for Object Image Classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1367–1374.

[33] C. Keßler, K. Janowicz, and M. Bishr, "An Agenda for the Next Generation Gazetteer: Geographic Information Contribution and Retrieval," in *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2009, pp. 91–100.

[34] H. Wang, M. Terrovitis, and N. Mamoulis, "Location Recommendation in Location-based Social Networks Using User Check-in Data," in *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2013, pp. 374–383.

[35] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[36] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[37] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *CoRR*, vol. abs/1409.1556, 2014.

[38] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 Million Image Database for Scene Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452–1464, 2018.

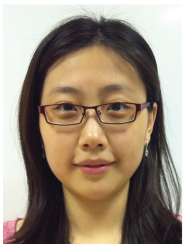
[39] S. Maji, A. C. Berg, and J. Malik, "Classification using Intersection Kernel Support Vector Machines is Efficient," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[40] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts," in *AAAI Conference on Artificial Intelligence*, 2016, pp. 194–200.

[41] D. Yang, D. Zhang, L. Chen, and B. Qu, "NationTelescope: Monitoring and Visualizing Large-scale Collective Behavior in LBSNs," *Journal of Network and Computer Applications*, vol. 55, pp. 170–180, 2015.

[42] D. Yang, D. Zhang, and B. Qu, "Participatory Cultural Mapping based on Collective Behavior in Location based Social Networks," *ACM Transactions on Intelligent Systems and Technology*, vol. 7, no. 3, pp. 30:1–30:23, 2016.
[43] X. Wu and Z. Zhou, "A Unified View of Multi-Label Performance Measures," in *International Conference on Machine Learning*, 2017, pp. 3780–3788.
[44] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A Real-world Web Image Database from National University of Singapore," in *ACM International Conference on Image and Video Retrieval*, 2009, pp. 48:1–48:9.
[45] N. Duong-Trung, N. Schilling, and L. Schmidt-Thieme, "Near Real-time Geolocation Prediction in Twitter Streams via Matrix Factorization Based Regression," in *ACM International on Conference on Information and Knowledge Management*, 2016, pp. 1973–1976.
[46] W. Mathew, R. Raposo, and B. Martins, "Predicting Future Locations with Hidden Markov Models," in *ACM Conference on Ubiquitous Computing*, 2012, pp. 911–918.
[47] C. Zhang, J. Han, L. Shou, J. Lu, and T. La Porta, "Splitter: Mining Fine-grained Sequential Patterns in Semantic Trajectories," *Proceedings of the VLDB Endowment*, vol. 7, no. 9, pp. 769–780, 2014.
[48] C. Zhang, K. Zhang, Q. Yuan, L. Zhang, T. Hanratty, and J. Han, "GMove: Group-Level Mobility Modeling Using Geo-Tagged Social Media," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1305–1314.

Zhenguang Liu is currently with Zhejiang Gongshang University. He was a research fellow in National University of Singapore and Singapopre Agency for Science, Technology and Research (A* STAR) from 2015 to 2018. He respectively received his Ph.D. and B.E. degrees from Zhejiang University and Shandong University, China, in 2010 and 2015. His research interests include multimedia data analysis, deep learning, and blockchain. Various parts of his work have been published in first-tier venues including TKDE, CVPR, AAAI, ACM MM, TMM, TOMM. Dr. Liu has served as technical program committee member for conferences such as ACM MM and MMM, and a reviewer for IEEE Transactions on visualization and computer graphics, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), Multimedia Tools and Applications, Sensors, etc.



Yifang Yin received the B.E. degree from the Department of Computer Science and Technology, Northeastern University, Shenyang, China, in 2011, and received the Ph.D. degree from the National University of Singapore, Singapore, in 2016. She is currently a senior research fellow with the Grab-NUS AI Lab at the National University of Singapore. She worked as a Research Intern at the Incubation Center, Research and Technology Group, Fuji Xerox Co., Ltd., Japan, from October, 2014 to March, 2015. Her research interests include machine learning, spatiotemporal data mining, and multimodal analysis in multimedia.

Sheng Wang received the B.E. degree from the Department of Computer Science and Technology, Harbin Institute of Technology, Harbin, China, in 2011, and the Ph.D. degree in Computer Science from National University of Singapore in 2016. He is currently a Research Scientist in Database and Storage Lab, DAMO Academy, Alibaba Group. His research interests are mainly in large-scale data management and processing, including OLTP/OLAP/NoSQL database engines, distributed systems, cloud data security, and machine learning platforms.

Ying Zhang received the B.E. degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2009, and the Ph.D. degree in computer science from the National University of Singapore, in 2014. She was a Senior Research Fellow with NUS from 2019-2021. Before that, she was the Head of Data Security Unit and Research Scientist with the department of Cyber Security & Intelligence, Institute for Infocomm Research (I2R), A*STAR, Singapore. Her research interests include machine learning, image and video processing, multimedia.

Rajiv Ratn Shah currently works as an Assistant Professor in the Department of Computer Science and Engineering (joint appointment with the Department of Human-centered Design) at IIIT-Delhi. He received his Ph.D. in computer science from the National University of Singapore, Singapore. Before joining IIIT-Delhi, he worked as a Research Fellow in Living Analytics Research Center (LARC) at the Singapore Management University, Singapore. Prior to completing his Ph.D., he received his M.Tech. and M.C.A. degrees in Computer Applications from the Delhi Technological University, Delhi and Jawaharlal Nehru University, Delhi, respectively. He has also received his B.Sc. in Mathematics (Honors) from the Banaras Hindu University, Varanasi. Dr. Shah is involved in organizing and reviewing of many top-tier international conferences and journals. His research interests include multimedia content processing, natural language processing, image processing, speech processing, multimodal computing, data science, and social media computing.

Roger Zimmermann (M'93–SM'07) received his M.S. and Ph.D. degrees from the University of Southern California (USC) in 1994 and 1998, respectively. He is currently an Associate Professor with the Department of Computer Science at the National University of Singapore (NUS), Singapore. He is also a Deputy Director with the Smart Systems Institute (SSI), and previously co-directed the Centre of Social Media Innovations for Communities at NUS. He has coauthored a book, seven patents, and more than 250 conference publications, journal articles, and book chapters. His research interests include streaming media architectures, distributed systems, mobile and geo-referenced video management, applications of machine/deep learning, and spatial data management. He is an associate editor for IEEE MultiMedia, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), Springer Multimedia Tools and Applications (MTAP), and IEEE Open Journal of the Communications Society (OJ-COMS). He is a distinguished member of the ACM and a senior member of the IEEE. Further information can be found at http://www.comp.nus.edu.sg/ rogerz/.