FedTour: Participatory Federated Learning of Tourism Object Recognition Models with Minimal Parameter Exchanges between User Devices

Shusaku Tomita[1,3], Jose Paolo Talusan[1], Yugo Nakamura[2], Hirohiko Suwa[1,3], [1], Keiichi Yasumoto[1,3]
[1]Nara Institute of Science and Technology, Nara 630-0192, Japan
[2]Kyushu University, Fukuoka 819-0395, Japan
[3]RIKEN Center for Advanced Intelligence Project AIP, Tokyo 103-0027, Japan
Email: http://ubi-lab.naist.jp

***Abstract*—In this paper, we propose FedTour, a federated learning-based method for training tourism object recognition models, which utilizes short-distance direct communication between user devices and maximizes the model performance within a limited number of updates. In FedTour, whenever two user devices are within range, they first exchange metadata including the learning degree (e.g., recognition accuracy) of their models, and determine whether it is effective to integrate the peer model by using a regressor trained with various pairs of models with different accuracy to predict the accuracy of the merged model. Once it is deemed effective, the model parameters are exchanged and the model is updated using FedAvg (averaging weights of two models of user devices). By carefully setting the threshold of whether FedAvg is applied or not, model performance is improved within a limited number of model parameter exchanges resulting in lower power consumption of user devices. We conducted a simulation using mobile phone trace data of actual users in a real sightseeing area and evaluated the improvement in accuracy of a CNN model that recognizes 10 objects while limiting the number of model parameter exchanges to only 40. Results show FedTour increased the initial model accuracy by 112%, while the baseline gossip-based method achieved 69%.**

***Index Terms*—Federated learning, Tourism object detection, Participatory learning, Direct communication**

I. Introduction

In recent years, AI technologies have been utilized for tourism services. There have been many AI-based systems [1], [2] that recommend sightseeing spots from photos uploaded to SNS. "Deaps," a smartphone application that recommends tourist spots and provides tourist information based on information posted by users on SNS and their behavioral history, has been developed. Smart tourism relies on adopting emerging technologies such as social media and AI to create new value propositions [3]. The White Paper on Tourism in Japan, 2021* by the Japan Tourism Agency, Ministry of Land, Infrastructure, Transport, and Tourism, reports that new and diverse travel styles such as staycation, decentralized travel, nearby travel, and online tours have been gaining popularity. Thus, the development and utilization of tourism AI will accelerate in the future to adapt to the diversity of tourists/tourism styles.

*https://www.mlit.go.jp/kankocho/news02_000447.html

One of the main functions of tourism AI is context recognition of tourist attractions (e.g., congestion level, availability of events, weather, scenery, etc.) in real-time. A promising method is the use of object recognition models to comprehensively estimate the context from a set of objects in photos and videos. A tourist spot contains unique attractions ranging from wild animals such as deer, unique architecture such as temples and shrines, and scenery such as cherry blossoms or autumn leaves. Building a recognition model for a wide variety of objects is non-trivial. A model has to learn vast amounts of data which is often held by large companies or stored on users' devices. Privacy and security concerns make collecting data difficult.

Federated Learning [4] has attracted a great deal of attention in recent years because it has the potential to train models while addressing privacy and security concerns. In this method, local models are trained on local datasets, and only parameters (e.g., model weights or gradients) are exchanged between clients to achieve a global model. Aggregation servers orchestrate the entire training process and update the global model without the need to access local client data. The process makes it hard to leak private information. When Federated Learning is applied to tourism object recognition models, photographs taken by tourists are used as training data. The more tourists participate, the more diverse the tourist object recognition models that can be obtained. However, in addition to the maintenance and operation cost of the aggregation server, there are also additional communication costs (communication fee, power consumption, etc.) because of the frequent large-capacity wide-area wireless communication with the server. Therefore, to create a tourism object recognition model based on federated learning, it is necessary to solve the following challenges: (1) avoid using wide-area wireless communication and aggregation servers, and (2) select a communication target with a model that will effectively maximize the accuracy of the merged model within a limited number of model updates (limiting the communication cost).

Several methods have been proposed to solve these challenges. Lee et al. [5] trained models by directly communicating with nearby devices instead of relying on servers. Chen et

al. [6] reduced federated learning's dependency on aggregation servers by having each device execute a process similar to that of the aggregation server. However, these are insufficient to solve the above problems because they either require more frequent communication to achieve higher accuracy or still require the existence of an aggregation server.

In this paper, we propose FedTour, a participatory federated learning scheme for tourism object recognition model, that solves the challenges (1) and (2). For challenge (1), we employ short-distance direct communication between user devices as [5]. For challenge (2), we develop a novel method allowing each user device to know how effective it is to integrate with the peer model while only exchanging a small amount of metadata with the encountered user device.

To develop the method, we collected accuracy data of object recognition models created by applying FedAvg [4] to combinations of two models with different accuracies and trained a model to predict the accuracy of the merged model from only the accuracy information of the two original models. We use this predicted accuracy to determine if the local model would be improved in the resulting update. The model parameters would be exchanged only if the predicted accuracy is greater than the local model by a predetermined threshold. Thus, it is expected to reduce communication costs and increase the possibility of improving the accuracy within a limited number of parameter exchanges between user devices.

Contributions: This paper presents, *FedTour*, a participatory federated learning algorithm for training models with minimal parameter exchanges between devices. This paper's main contributions are as follows:

- Train models without the need for aggregation servers, using instead short-range direct communication (WiFi Direct, BLE, etc.) between tourist devices.
- A model update algorithm that takes into account both the movement of tourists and the predicted accuracy of the resulting model before any parameter exchange occurs. This improves local model accuracy while limiting the number of model exchanges required for training.
- A simulation that shows how the proposed algorithm performs against a gossip protocol-based method in which models are exchanged randomly, without regard for the resulting model accuracy. It was found that the proposed method improves upon the initial recognition model by 112% compared to the gossip-based method which only showed a 69% improvement.

II. Related Work

Federated Learning is one of the machine learning methods proposed by Google. It is capable of learning on private data such as personal photos and search histories, which are difficult to train a global model on. The first Federated Learning method proposed was Centralized Federated Learning, which consists of an aggregation server and edge devices. Later, Decentralized and Distributed Federated Learning methods were proposed.

A. Decentralized Federated Learning

Decentralized Federated Learning places an intermediate server or device as a relay point between the central server and the edge devices, thus reducing the direct load on the aggregation server. In the methods [6]–[9] using Decentralized Federated Learning, the number of accessible devices is increased by setting up a relay point, so that a wider range of edge devices can participate in federated learning. There is another approach [10] that uses multiple central servers and updates their global model obtained from each local network while communicating between servers.

B. Distributed Federated Learning

Distributed Federated Learning mainly utilizes connections between edge devices. This learning method is categorized into two types: one targets the fixed network topology (e.g., links between devices always exist like Peer-to-peer network) between devices [11]–[14], and the other does the topology which dynamically changes (e.g., existence of links between devices is opportunistically changed) [5], [15], [16]. The distributed federated learning eliminates the need to install servers and allows an unspecified number of devices to participate in the training. However, due to the push-only nature of these protocols, a device may end up merging its fresher model with an outdated one. In this study, we consider the devices possessed by tourists' (e.g., smartphones) as edge devices, and since we assume that many tourists come and go in the tourist spots, we assume the latter network.

Among the existing studies in this category, Opportunistic Federated Learning [5] is the only other approach that implements encounter-based pairwise collaborative learning. They also use short-distance direct communication between devices, making learning possible without using wide-area communication. Model integration only occurs when it is beneficial and feasible, increasing training efficiency. However, while their approach modifies the number of training rounds per encounter based on the duration of the encounter, it does not put a ceiling on the total number of parameter exchanges and model integration. This may cause the user device to consume more power due to continuous integration as the user encounters other devices. Our approach can limit the number of exchanges a device will make, lowering the power consumption on the user device, while maintaining high model performance.

III. Problem Statement

Our primary goal is to implement a tourism object recognition model based on Federated Learning with user participation. In this section, first, we show a use case scenario and identify challenges to solve. Then, we formulate the problem.

A. Use Case Scenario

Suppose tourists will be visiting, Nara Park, a well-known tourist area in Japan. There are three famous sightseeing spots, Nara Park, Todaiji Temple, and Kasuga Taisha Shrine. Each of these tourist spots features a set of objects that uniquely

characterize the area. These objects are denoted as O_{NP}, O_{TT}, and O_{KT}, respectively.

$$O_{NP} = \{\text{male deer, female deer, fawn,...}\}$$
$$O_{TT} = \{\text{statue of Buddha, pond, carp, fawn,...}\}$$
$$O_{KT} = \{\text{wisteria, torii, pond, carp, shrine,...}\}$$

We assume that each of these spots also has a signage device installed, c_{NP}, c_{TT}, and c_{KT}. We assume that these have no training data, but have the baseline models (trained by public data or given from tourist devices) that can recognize the objects within their areas.

Let there be nine tourists who will travel around these sightseeing spots. These tourists own mobile devices, $c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8$, and c_9, respectively. The tourists download a base model trained on a limited number of sightseeing objects beforehand (from signage or the Internet). They train this base model on any of their annotated data, if available. Fig. 1 shows the path of the tourist c_1 through the different sightseeing spots. As the tourist c_1 travels between locations (Nara Park to Todaiji Temple and finally to Kasuga Taisha Shrine), s/he comes into contact with other tourists and signage. At each of these events, both the tourist and the other tourist/signage iteratively update their own models through parameter exchanges.

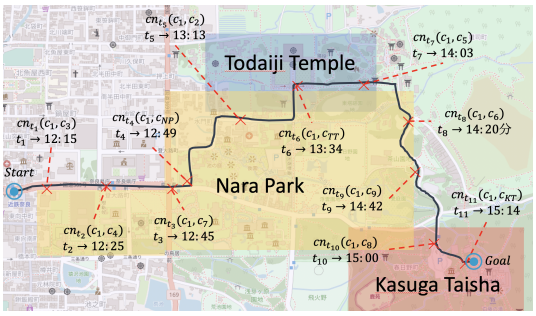

Fig. 1: Contact between c_1 and each device

B. Challenges

In the prior scenario, if conventional Federated Learning [4], [17] is used for model updates, an aggregation server that can be accessed by all devices is required. However, as the number of tourists increases, the load on the server and the network bandwidth required to access it becomes saturated. When this occurs, the aggregation server limits the number of models that can be trained and reduces the update frequency.

A CNN model for image classification is used as the tourism object recognition model. Updating models such as these consume a large amount of bandwidth. Lee et al. [5] found that the communication process consumes a lot of power because direct communication is performed multiple times during training between devices. Since tourists do not always have access to methods to charge their devices, it is necessary to reduce the power consumed by these model updates.

For these requirements to be met, the following challenges must be solved:

1) Update models without the use of wide-area wireless communication and aggregations servers.
2) Maximize the model accuracy while minimizing the number of parameter exchanges.

C. Problem Formulation

We define the assumed environment and summarize the notations used in Tab. I.

TABLE I: Elements in the Assumed Environment

Element	Description
A	Set of tourist area
A_c	Target area for enhanced cognitive ability of c
O_a	Set of recognized objects in $a \in A$
O_c	Set of recognition enhancement objects in c
$C_{stationary}$	Set of fixed devices
C_{mobile}	Set of mobile devices
C	Set of all devices
R	Communication range
D_c	Set of data in c
D	All data
M_c	Model in c.
W_c	Model parameter of M_c
T	Set of time t in the assumed environment
$pos(c,t)$	Position of c at time c.
$cn(c,c',t)$	Contact of c, c' at time t.
CN	Contact of all devices

We consider A to be a set of sightseeing areas. In each tourist area $a \in A$, there is a set O_a of tourist objects to be recognized. O_a can be dynamic objects (animals, crowds, stalls, etc.) or fixed objects (buildings, gates, trees, etc.). There are multiple tourists moving within and travelling between different tourist areas in A. Each tourist is assumed to have a single mobile device such as a smartphone, $c \in C_{mobile}$, where C_{mobile} is the set of mobile devices. In addition, fixed devices such as signage are placed in the sightseeing area, and the set of these devices is denoted by $C_{stationary}$. The set of devices C in the assumed environment A is denoted by equation (1).

$$C = C_{stationary} \cup C_{mobile} \tag{1}$$

The device $c \in C_{mobile}$ changes its position in the environment at every timeslot $t \in T$. We denote the position of a device c at t as $pos(c,t)$. If $pos(c,t)$ and $pos(c',t)$ are within communication range R, they are considered to be in contact. Contact between c and c' is denoted by $cn(c,c',t)$. The set of all contacts CN then is denoted by equation (2).

$$CN = \bigcup_{c,c' \in C, t \in T} cn(c,c',t) \tag{2}$$

Each mobile device $c \in C_{mobile}$ has training data, while fixed devices $c \in C_{stationary}$ do not. We denote this local data in each $c \in C$ as D_c, where $D_c \neq \emptyset$ for $c \in C_{mobile}$ and $D_c = \emptyset$ for $c \in C_{stationary}$. All the data in this environment are defined as D by equation (3).

$$D = \bigcup_{c \in C_{mobile}} D_c \tag{3}$$

Each $c \in C_{mobile}$ has a tourist object recognition model M_c trained on its own data D_c. These weights W_c of M_c will be

averaged with $W_{c'}$ of other tourists. The model M_c in each c has a tourist area $A_c \subseteq A$ to be recognized, and the set of tourist objects O_c is defined as the expression (4).

$$O_c = \bigcup_{a \in A_c} O_a \tag{4}$$

When two tourists come into contact at time t a parameter exchange may occur. When a tourist is in range of two or more other tourists, they choose at most one of them to communicate and exchange parameters with. We assume that parameter exchange of the models between devices are completed within timeslot t (we assume enough time width such as 30 seconds for each time slot t). This is defined by Equation (5) using the binary variable $x_{cn(c,c',t)}$. $x_{cn(c,c',t)}$ is a variable that indicates the presence or absence of communication between devices, $x_{cn(c,c',t)} = 1$ when communication (model parameter transmission) occurs from device c to another device c', and $x_{cn(c,c',t)} = 0$ otherwise.

We assume that the model parameter transmission and reception by each device can occur at most once per time slot as follows.

$$\sum_{cn(c,c',t) \in CN, c \neq c'} x_{cn(c,c',t)} \leq 1, \forall c \in C, \forall t \in T \tag{5}$$

We also limit the number of times the model parameter of each device can be exchanged, defined as L. This is done in consideration of power consumption suppression. Equation (6) expresses this constraint.

$$\forall c \in C, \sum_{cn(c,c',t) \in T} x_{cn(c,c',t)} + x_{cn(c',c,t)} \leq L \tag{6}$$

The goal of the proposed approach is to maximize the model improvement with a minimal number of parameter exchanges. This is done by selecting only contacts which will effectively improve the accuracy of the local model. This degree of improvement of the model accuracy after integrating the parameter $M_{c'}$ into M_c is given by $Improve(M_c, M_{c'})$. This objective function is expressed in Equation (7).

$$\begin{aligned} Maximize \sum_{c \in C} \sum_{c' \in C\{c\}} \sum_{t \in T} \sum_{cn(c,c',t) \in CN} \\ x_{cn(c,c',t)} \cdot Improve(M_c, M_{c'}) \\ subject\ to\ (5) - (6) \end{aligned} \tag{7}$$

The problem defined above is NP-hard or in more difficult problem classes since it implies the Knapsack problem as a special case even when the trajectory of each mobile device is known in advance.

IV. FedTour

To solve the problem defined in the previous section, we propose *FedTour*, a novel and efficient model update method based on distributed federated learning. FedTour reduces unnecessary parameter exchanges by predicting the object recognition accuracy of the merged model even before the node and the peer node exchange parameters.

A. Investigation of Impact of Model Integration

The purpose is to predict the accuracy of the merged model with as small amount of information as possible before merging two models because the parameter size of CNN models is huge (e.g., 60 MB) in general.

To investigate the impact of FedAvg-based model integration, we checked the effect of the integration on accuracy improvement. We created a large number of CNN models with various recognition accuracy (trained with different data in the same dataset), integrated any pair of models, and checked the number of pairs whose accuracy was improved. We employ a simple integration method where all model parameters are simply averaged between two models. The model to be integrated uses VGG16, and fine tuning of the convolutional layer is performed. For fine tuning, we used parameters already learned in ImageNet.

In this study, we focused on the 10-class classification problem as a feasibility study to validate the effectiveness of our method. Since there is a limit to the number of niche objects available at each tourist attraction, we believe that a 10-class classification problem is a reasonable first step. Thus, we used CIFAR-10 dataset[†], which includes 60,000 images consisting of 10 types (classes) of objects. We split the data into 50,000 and 10,000 images for training and evaluating the integrated model respectively. To maintain the diversity in accuracy of detecting each class, we created multiple datasets with a different number of images per class. To keep the diversity in accuracy of detecting each class, we created multiple datasets with a random number of images per class. The number varied from small (0-1000), medium (1001-3000) and large (3001-5000).

Among 3^{10} combinations (three categories for each of 10 classes), 66 diverse combinations were selected, and for each combination, we created three datasets, resulting in 198 datasets. Also, for models with significantly lower accuracy, we added 33 more datasets with (0-10) or (11-100) additional data per class. In total, 231 models were created and the accuracy of the integrated model was recorded.

Out of 53,361 (231×231) pairs of models, the integration improved the accuracy of 37,315 pairs. The integration offers a chance of accuracy improvement for the integrated model.

B. Accuracy Prediction

Based on the integration results in the previous section, we constructed a support vector regressor that predicts the change in the accuracy of the user's model after integration based on the accuracies of two initial models. To train the regressor, we used the difference in accuracy between before and after integration of two models as prediction value, and accuracies of the two models as the input value. The pair of prediction and input values can be obtained from the integration results.

[†]https://www.cs.toronto.edu/ kriz/cifar.html

Algorithm 1 FedTour Algorithm

Input: $\{W_c^t, Q_c^t, t\}$

1: $threshold \leftarrow getThreshold()$;
2: $txn \leftarrow numTransmission(Q_c^t, L)$;
3: $rxn \leftarrow L - txn$;
4: $continueTour \leftarrow true$
5: **while** $continueTour$ **do**
6: **if** $cn(c, c', t)$ **then**
7: **if** $txn > 0$ **then**
8: $sendMetadata(c', Q_c^t)$;
9: **if** $TxRequested(c')$ **then**
10: $sendWeights(c', Q_c^t)$;
11: $txn \leftarrow txn - 1$;
12: **end if**
13: **end if**
14: $Q_{c'}^t \leftarrow recvMetadata(c')$;
15: $Q' \leftarrow Q_c^t + accuracyRegressor(Q_c^t, Q_{c'}^t)$
16: **if** $Q' > threshold$ **then**
17: **if** $rxn > 0$ **then**
18: $W_{c'}^t \leftarrow RecvWeights(c')$
19: $W_c^{t+1} \leftarrow fedAvg(W_c^t, W_{c'}^t)$
20: $rxn \leftarrow rxn - 1$
21: **end if**
22: $Q_c^{t+1} \leftarrow evaluate(W_c^{t+1})$
23: **if** $(Q^{t+1} > threshold) \vee (Q_c^{t+1} > Q_c^t)$ **then**
24: $threshold \leftarrow increaseThreshold(Q_c^{t+1})$
25: **else**
26: $threshold \leftarrow decreaseThreshold(Q_c^{t+1})$
27: **end if**
28: **end if**
29: **end if**
30: $t \leftarrow t + 1$;
31: $continueTour \leftarrow decideContinueTour(c, t)$
32: **end while**

Thus, the regressor is trained with 53,361 integration patterns. The trained regressor achieved mean absolute percentage error (MAPE) under 6%.

C. Peer Model Selection

As tourists move through the area, their models will be constantly integrating other users' models. This method of model integration allows models to iteratively update and train with the goal of improving the accuracy. However, this can be ineffective, requiring more integration to reach higher accuracy.

In the proposed method, we set a threshold for the target accuracy of the integrated model. Each user device performs model integration only if the predicted accuracy of the integrated model exceeds this threshold.

The regressor in IV-B is used for this purpose. First, when a user device encounters the other user's device, they exchange only the accuracy information of their models and predict the accuracy improvement that would be gained by integrating each other's model. Then, either (both) of the devices requests the peer device's model parameters only if the threshold constraint is satisfied. Finally the device receives the parameters and integrates them by the integration in IV-A.

D. FedTour Algorithm

Algorithm 1 shows how FedTour is run on each device c. It takes model parameters W_c^t and accuracy Q_c^t of c's model at the present time t as inputs. At line 1, it first sets the appropriate threshold. It will update the model with the peer model parameters only when the merged model's accuracy exceeds this threshold. The threshold is determined by the function $getThreshold()$[‡] as a median value of the distribution of model accuracy of all devices C. If Q_c^t is greater than the median value, the threshold is set to Q_c^t (so that the model accuracy is improved by update). At lines 2-3, the limit of parameter exchange times L is distributed to the limit of transmissions txn and the limit of receptions rxn. We use the function $numTransmissions()$ to determine the value for txn. We empirically employ the formula $txn = L \times (Q_c^t - Q_{min})/(Q_{max} - Q_{min})$ where Q_{max} and Q_{min} are maximum and minimum accuracy of the models in all devices C. The main loop of the algorithm is between lines 5–31. Whenever a contact $cn(c, c', t)$ happens, c sends the metadata (i.e., Q_c^t) and the model weights (Q_c^t) if requested, to the peer device c' at lines 7–13. At lines 14–28, it tries to update the model. At line 14, the metadata (model accuracy) $Q_{c'}^t$ of the peer device is received. This is used as input in $accuracyRegressor(Q_c^t, Q_{c'}^t)$ which predicts the accuracy of the updated model Q'. If the accuracy of Q' is higher than $threshold$, then the peer model's weights $W_{c'}^t$ are received and the updated weights W_c^{t+1} are obtained by averaging W_c^t and $W_{c'}^t$. At lines 22–27, $threshold$ is increased or decreased depending on the result of update. When the updated model's accuracy is higher than the original model, then $threshold$ is increased to continue improving the accuracy of the model in the future. Otherwise, $threshold$ is decreased so that the update probability will be higher in future encounters.

V. Simulation Experiment

To evaluate the proposed method, we conducted a model update simulation based on mobile phone trace data.

A. Overview of Simulation and Evaluation

In the simulation, we evaluate the change in the average accuracy of all models when the number of model parameter exchanges is limited. Fig. 2 shows the overview of the simulation. Each user is provided a CNN model for image classification and model updates are done via simple averaging as described in Sect. IV-A. $threshold$ is initially set to the median of the overall model accuracy for all users or to the accuracy of the local model if it is higher than the median. The $decreaseThreshold$ and $increaseThreshold$ functions modify the $threshold$ to 0.9 and 1.1 times the current value respectively. txn is set to vary in proportion to the accuracy of the model.

[‡]We assume that each device uploads its current model accuracy to the cloud server when the application is activated and the sever has the distribution of model accuracy of all devices in the target area A.

Algorithm 1 is then run on all user devices. We compared the proposed method to a gossip method that randomly performs the model parameter exchange with the encountered user with a probability of 10%. We conducted the simulation with these two methods for each contact in chronological order. Since the model is about 58.9MB of data and 18 transmissions or receptions are about 1GB, we set the limit of model parameter exchange times L to 40 to keep the communication volume within 2.5GB. The proposed method and the gossip method each have different methods for setting the number of reception and transmission limits. In the proposed method, reception limit rxn and transmission limit txn are determined by dividing $L = 40$ based on the formula described in Sect. IV-C. In the gossip method, rxn and txn are set to half of the limit $L = 40$. After the simulations, we calculated the average accuracy of all models and evaluated the changes. We also evaluated the average accuracy when L was set to 20, 30, 50, 70, and 100 times.

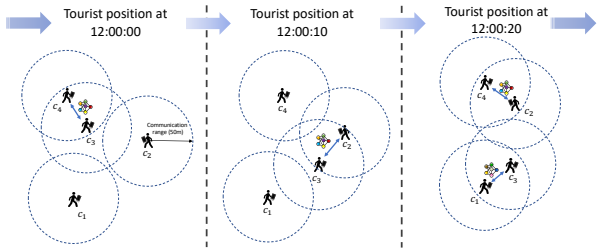

Fig. 2: Overview of simulation experiment. For each time slot, we use the coordinates of each user to find which users are pass by each other. Each user is assumed to only be able to communicate with devices within a radius of 50 meters. If there are users that pass each other in a time slot, they will execute the proposed algorithm.

B. Mobile Phone Trace Data

In the simulation experiment, we used the "point-type flow population data" [18] provided by Agoop Inc. We extracted users, who were considered to be tourists, from the trace data in the area of Fig. 3 during the time period of 6:00 to 18:00 from October 31, 2020 to November 30, 2020, and obtained data for 1,900 users. Since the number of users per day is small and it is not possible to trace users for more than two days, we treated the data for one month as the data for one day (assuming that Agoop data covers about a few percentage of the actual population). In addition, if a user was within a 50-meter radius other users, we considered them as having made contact.

C. Model Assigned to Each User

We built a model for 1,900 people for the simulation. The model used was VGG16. For the dataset we used CIFAR-10 instead of the tourist photos. 50,000 out of the 60,000 images in CIFAR-10 were used for training, and the training data were distributed to each model while avoiding duplication. When these data were assigned to each user, the accuracy

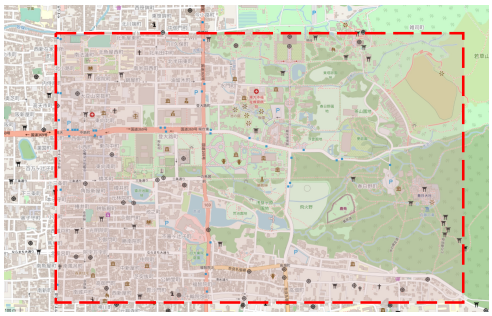

Fig. 3: Simulation area

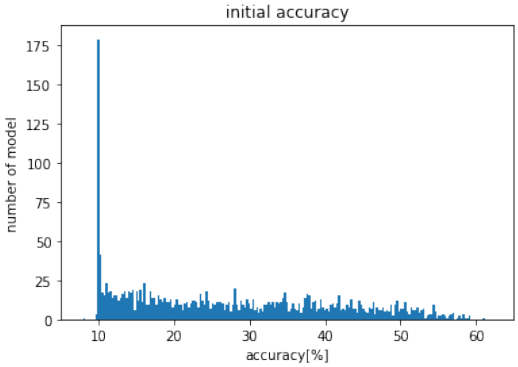

Fig. 4: Initial accuracy distribution of the models

of the entire model became low, so we expanded the data to 500,000 images (by applying rotation, magnification, etc). The remaining 10,000 images were used to evaluate the accuracy of each model in simulation. It is expected that many users will have only a few images in the real environment. The distribution of data to each model was made so that the number of users with less data would be larger, and conversely, the number of users with more data would be smaller. As a result of training and evaluating the models after data distribution, the accuracy of each model was distributed as shown in Fig. 4, where many of the models have an accuracy of 10% to 15%, and these were used as the initial models for the simulations.

D. Simulation Results

The accuracy distribution of each user's model, given a limit of 40 parameter exchanges, is shown in Fig. 5. The average accuracy of all models is 45.97% with the gossip method, which is 1.69 times higher than the initial average accuracy of 27.22% in Fig. 4, while the initial maximum accuracy of 61.14% was decreased to 59.04% after simulation. Using the FedTour algorithm results in an average accuracy of 57.74% as shown in Fig. 5 which is 2.12 times higher than the initial average accuracy. The maximum accuracy of the model is 62.45% which is higher than the initial maximum accuracy.

The average accuracy for different limits of parameter exchange times ($L = 20, 30, 40, 50, 70, 100$) is shown in Fig. 6. In both methods, the average accuracy tends to increase as the limit of parameter exchange times is increased, but overall, the average accuracy obtained by FedTour is higher than that of the gossip method. In particular, FedTour outperformed the gossip method by at least 10%, regardless of the limits.

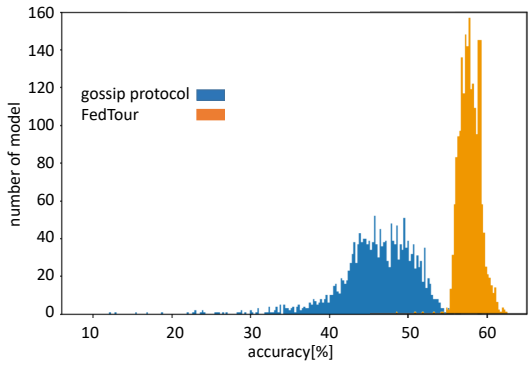

Fig. 5: Accuracy distribution after simulation

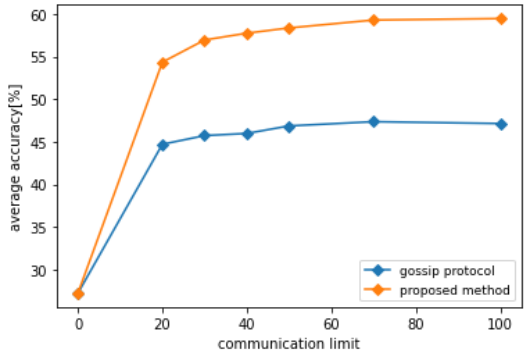

Fig. 6: Average accuracy for each communication limit count

For FedTour, when $L > 40$, the accuracy improvement is saturated. This is due to the difficulty each user had in finding a peer that would improve their model accuracy. However, this saturation occurs even before the actual parameter exchanges reaches the defined limit. For $L = 50$, the actual parameter exchanges were less than 10.

VI. Conclusion

In this paper we proposed FedTour, a method to construct tourism object recognition models using tourists' personal data. Here, a model update method based on federated learning is adopted to learn the data while protecting the privacy information within the tourist's data. Specifically, when two user devices are in range, their model parameters are exchanged, and updated using FedAvg. To reduce the number of model parameter exchanges, we proposed a FedTour algorithm that predicts the change in accuracy after updating. Model parameters are exchanged only when improved model accuracy is expected. We conducted a simulation with the proposed method and the gossip-based method and evaluated it by the average of the final model accuracy. Results show that the average accuracy of the gossip-based method is 45.97% and that of the proposed method is 57.74%. Since the average accuracy of the model before the simulation was 27.22%, there is a significant improvement in accuracy when the proposed method is applied. We believe that predicting the accuracy before the actual parameter integration enable us to select a model that is effective in improving accuracy under a limited number of parameter exchanges.

In the future, we will conduct simulations under various conditions, such as changing the number of simulation days and the number of tourists, to verify the effectiveness of this method. We also plan to evaluate our method under real world situations with more tourism objects included.

Acknowledgment

This work was partly supported by JSPS KAKENHI Grant Number JP21H03431.

References

[1] C.-Y. Tsai, G. Paniagua, Y.-J. Chen, C.-C. Lo, and L. Yao, "Personalized tour recommender through geotagged photo mining and lstm neural networks," in *MATEC Web of Conferences*, vol. 292. EDP Sciences, 2019, p. 01003.

[2] C.-Y. Sun and A. J. T. Lee, "Tour recommendations by mining photo sharing social media," *Decis. Support Syst.*, vol. 101, pp. 28–39, 2017.

[3] U. Gretzel, M. Sigala, Z. Xiang, and C. Koo, "Smart tourism: foundations and developments," *Electronic markets*, vol. 25, no. 3, pp. 179–188, 2015.

[4] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," *Proceedings of the 20 th International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 54, 2017.

[5] S. Lee, X. Zheng, J. Hua, H. Vikalo, and C. Julien, "Opportunistic federated learning: An exploration of egocentric collaboration for pervasive computing applications," in *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2021, pp. 1–8.

[6] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Wireless communications for collaborative federated learning in the internet of things," *IEEE Communications Magazine*, vol. 58, no. 12, pp. 48–54, 2020.

[7] Y. Ye, S. Li, F. Liu, Y. Tang, and W. Hu, "Edgefed: Optimized federated learning based on edge computing," *IEEE Access*, vol. 8, pp. 209 191–209 198, 2020.

[8] Z. Yu, J. Hu, G. Min, H. Xu, and J. Mills, "Proactive content caching for internet-of-vehicles based on peer-to-peer federated learning," *2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 601–608, 2020.

[9] J. Yuan, M. Xu, X. Ma, A. Zhou, X. Liu, and S. Wang, "Hierarchical federated learning through lan-wan orchestration," *ArXiv*, vol. abs/2010.11612, 2020.

[10] R. Wu, A. Scaglione, H.-T. Wai, N. Karakoç, K. Hreinsson, and W.-K. Ma, "Federated block coordinate descent scheme for learning global and personalized models," in *AAAI*, 2021.

[11] A. Lalitha, O. C. Kilinc, T. Javidi, and F. Koushanfar, "Peer-to-peer federated learning on graphs," *CoRR*, vol. abs/1901.11173, 2019. [Online]. Available: http://arxiv.org/abs/1901.11173

[12] A. G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger, "Braintorrent: A peer-to-peer environment for decentralized federated learning," *ArXiv*, vol. abs/1905.06731, 2019.

[13] H. Wang, L. Muñoz-González, D. Eklund, and S. Raza, "Non-iid data re-balancing at iot edge with peer-to-peer federated learning for anomaly detection," *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2021.

[14] A. Koloskova, S. U. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," 2019.

[15] A. Elgabli, J. Park, A. S. Bedi, M. Bennis, and V. Aggarwal, "Gadmm: Fast and communication efficient framework for distributed machine learning," 2020.

[16] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Federated learning for data privacy preservation in vehicular cyber-physical systems," *IEEE Network*, vol. 34, no. 3, pp. 50–56, 2020.

[17] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.

[18] Agoop, "Apply big data to spark happiness around the world. – big data as a value – creating new business values from large-scale mobile data analytics," https://www.agoop.co.jp/en/, accessed: 2021-07-15.