Indoor Topological Localization Based on a Novel Deep Learning Technique

Qiang Liu, Ruihao Li, Huosheng Hu and Dongbing Gu

***Abstract*—Introduction: Millions of people in the world suffer from vision impairment or vision loss. Traditionally, they rely on guide sticks or dogs to move around and avoid potential obstacles. However, both guide sticks and dogs are passive. They are unable to provide conceptual knowledge or semantic contents of an environment.**

Methods: To address this issue, this paper presents a vision-based cognitive system to support the independence of visually impaired people. More specifically, a 3D indoor semantic map is constructed first with a hand-held RGB-D sensor. The constructed map is then deployed for indoor topological localization. Convolutional Neural Networks are used for both semantic information extraction and location inference. We additionally use semantic information to further verify localization results and eliminate errors. The topological localization performance can thus be improved despite significant appearance changes within an environment.

Results: Experiments have been conducted to verify that the proposed method can increase both precision and recall rates.

Conclusions: The system can be potentially deployed by visually impaired people to help them move around independently.

***Index Terms*—Localization, semantic map, Convolutional Neural Networks, visually impaired people.**

I. Introduction

Nowadays, 285 million people are estimated to be visually impaired worldwide, among which 39 million suffer from total blindness [1]. Guide sticks and dogs can be deployed to help visually impaired people move around independently. However, guide sticks are not effective enough to use and guide dogs are expensive to train. Furthermore, both of them are unable to interact with human users or provide conceptual knowledge or semantic contents of an environment. Thus, it remains a major challenge for visually impaired people to move around independently, especially in an unfamiliar environment. This paper proposes a potential solution by using wearable electronic devices which are capable of localizing objects, planning paths and providing audio prompts.

The proposed vision-based assistive system is shown in Fig. 1. The system consists of a wearable device and a server, both of which are connected to the Internet. The server carries out data processing tasks, e.g., map building, model training and location inference. The wearable device consists of an Odroid XU3 board, a USB camera and a pair of earphones. The wearable device collects images, sends them to the server, receives localization results and provides audio prompts. The system can infer topological locations when a user visits an unfamiliar environment, e.g., shopping malls, museums or office buildings.

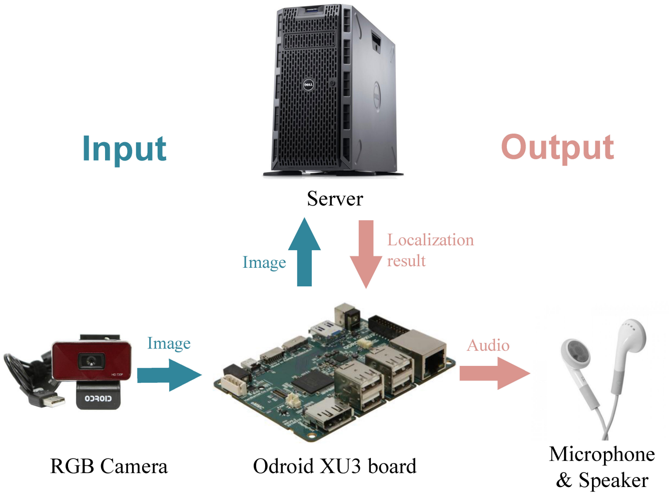


Fig. 1: An overview of the proposed vision-based assistive system.

In order to carry out the tasks mentioned above, a map of the environment needs to be constructed beforehand. Maps built by traditional robotic systems are either geometric or topological maps, which are navigation oriented. Both of them are designed for obstacle avoidance and path planning [2]. However, they are passive and cannot communicate with users or provide semantic assistive guidance. A map containing human-compatible information is required. Semantic information such as location and object names should be interpreted from scenes during map building [3]. In other words, a semantic map containing linguistic words that represent places, landmarks and daily objects is necessary since it serves as an effective human-machine interface.

In this paper, we first build a 3D indoor geometric map with an RGB-D sensor and an off-the-shelf algorithm [4]. To extract semantic information, we adopt deep Convolutional Neural Networks (ConvNets) for object detection, rather than the bag-of-visual-words model (BoW) which is commonly deployed by the SLAM community in recent years. In the case of ConvNets, models trained for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [5] can classify 1.3 million images into as many as 1,000 classes with high accuracy. Therefore, much more objects in an environment can be

Q. Liu, R. Li, H. Hu and D. Gu are with the School of Computer Science and Electric Engineering, University of Essex, Colchester, CO4 3SQ, UK. E-mail: oceanlq0830@gmail.com, liruihao2008@gmail.com , hhu@essex.ac.uk, dgu@essex.ac.uk. Conflict of Interest: The authors declare that they have no conflict of interest.

detected. Only object names are stored in our database rather than raw images. Once an object is detected, the relationship between the object and locations is then represented by the anchoring method. A semantic map is thus obtained. We then use the pre-built semantic map for indoor topological localization.

Generally speaking, visual localization failure is always caused by significant environmental appearance changes [6]. For instance, lighting conditions vary between day and night. Objects (chair, laptop, mug, curtain or even human) may be at random locations. To address these problems, we propose a novel localization method based on a two-stream ConvNet. Previous literature [7], [8], [9] has proven that object recognition can help place recognition. Thus, we use distinctive objects detected and labeled during semantic mapping to further verify the localization results.

Experiments with long-term operations have been carried out. Compared to other state-of-the-art algorithms, our method generates higher precision and recall rates (recall is an essential factor for other tasks e.g. online map updating). The rest of the paper is organized as follows. Section II reviews related works on both semantic mapping and place recognition. The proposed semantic information extraction and topological localization methods are detailed in Section III. Training and experimental results are subsequently presented and discussed in Section IV. Finally, a brief conclusion and future works are given in the last section.

II. Related Works

Semantic mapping has become a popular research topic in the past decade and drawn enormous attention in the robotics domain. Generally speaking, semantic mapping can be divided into two steps, namely environment construction (also called SLAM) and scene understanding.

In the case of environment construction, traditional methods are feature-based [4], [10]. Endres *et al.* [4] proposed a real-time indoor mapping system which deployed a low-cost, light-weight Kinect camera. SURF feature [11] was adopted by this system. Several deep learning based systems [12], [13], [14] have also been proposed recently. These systems employ an end-to-end training manner. Compared to traditional feature-based methods, they can be easily applied to other scenes without labor-intensive structure redesigning [15]. Since SLAM is not our focus in this paper, we adopt an off-the-shelf feature-based method [4] for environment construction.

Semantic information can be extracted from both range and visual sensors. In indoor environments, semantic information extraction can be essentially considered as object, place or sign recognition. Grimmett *et al.* [16] proposed a vision-only automated parking system which can identify driving lanes and parking spaces. Hart *et al.* [17] presented a door sign localization method based on a corner feature. A custom ConvNet was proposed by Maturana *et al.* [18] to classify ground types such as trail and grass. In our paper, types of rooms and objects are considered as semantic information.

Topological localization tackles the problem of recognizing places when we revisit a scene. Arroyo [19] and Li *et al.* [20] realized outdoor topological localization with a visual sensor and a GPS respectively. Several binary feature descriptors were tested in [19]. In terms of visual features, the BoW model has been widely used in recent SLAM systems [21]. However, Sharif Razavian *et al.* [22] have shown that ConvNets outperform BoW in terms of most recognition tasks, especially when significant appearance changes exist [23], [24].

ConvNets have been widely used as robust visual feature extractors in computer vision and machine learning domains. Although some ConvNets are trained for a specific task (e.g., object recognition), researchers have managed to transfer these models for other related but different tasks such as image super-resolution, image segmentation, place recognition and object detection [25], [26], [22], [27]. This is because the generic features learned by a ConvNet are always versatile and transferable [28], [6]. In this paper, we adopt the Inception-v3 model [29] due to its high performance in the ImageNet competition. Inception-v3 is an updated version of GoogLeNet [30]. Rather than simply stacking convolutional layers deeper and deeper, it is heavily engineered and carefully fine-tuned. The inception building blocks convolve the input tensor with multiple filters and then concatenate their results. Batch normalization is applied to activation inputs and is used extensively throughout the model. The culmination of ideas developed by multiple researchers leads to the first runner up for image classification in 2015 with a top 5 error rate of 3.58%.

Researches have already shown that place recognition can benefit from object recognition [7], [8], [9], especially in indoor environments where the type of a room can be easily revealed by the objects detected in it. However, if a recognition method relies only on objects, it fails in the case where no distinctive objects can be spotted within the camera's field of view. Moreover, some objects (curtain, mug, computer, etc.) are not distinctive enough to infer locations on their own. Zeng *et al.* [31] proposed a pedestrian reidentification system by using a two-stream multi-rate recurrent neural network to extract both spatial static feature and motion optical flow feature. Chen *et al.* [27] also incorporated an addition ConvNet for depth images to boost the object class detection performance. In our paper, we propose a two-stream ConvNet for topological localization which combines object detection with holistic image recognition.

III. Preliminaries

This section explains our semantic mapping and topological localization methods. Readers who are familiar with 3D environment construction or visual SLAM may wish to skip to Section III-B directly.

A. Environment Construction

Our environment construction method is modified from Felix's approach [4]. The method is presented with the blue boxes in Fig. 2. Both RGB and depth images are deployed as system inputs. All RGB images are saved to train our ConvNet later. SURF feature is first extracted from an RGB image. We then calculate a pose transformation (rotation and translation)

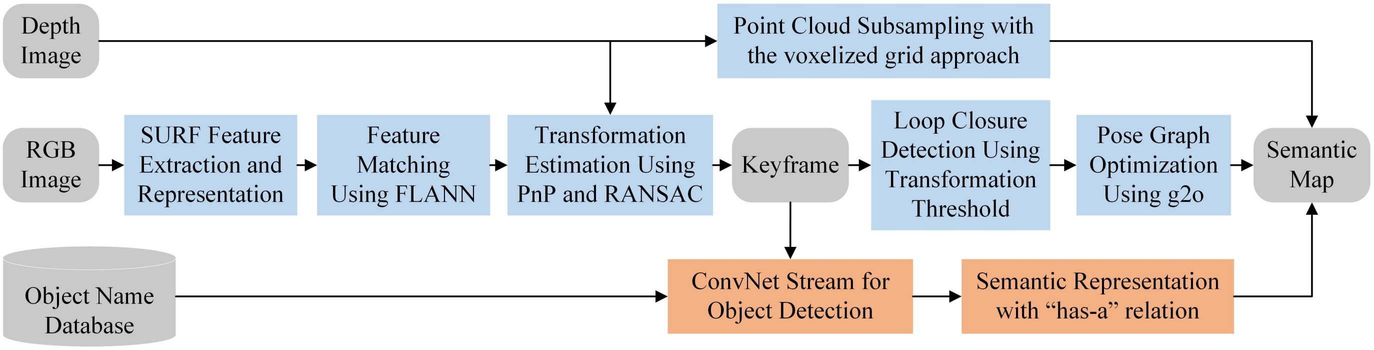

Depth Image
Point Cloud Subsampling with the voxelized grid approach
RGB Image
SURF Feature Extraction and Representation
Feature Matching Using FLANN
Transformation Estimation Using PnP and RANSAC
Keyframe
Loop Closure Detection Using Transformation Threshold
Pose Graph Optimization Using g2o
Semantic Map
Object Name Database
ConvNet Stream for Object Detection
Semantic Representation with "has-a" relation

Fig. 2: Semantic mapping. Blue boxes: 3D scene construction. Orange boxes: semantic information extraction.

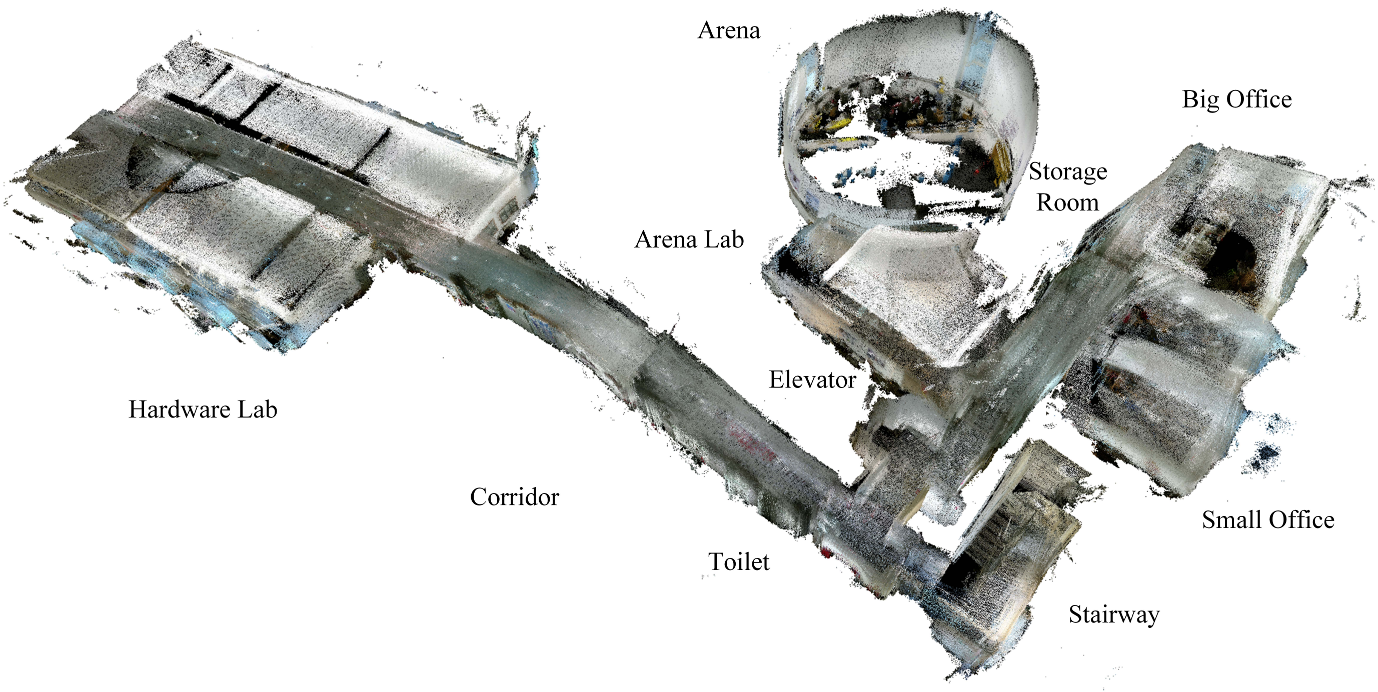


Fig. 3: The 3D map of a floor in an office building.

matrix from the RGB image and the depth image of its previous keyframe. A new keyframe is labeled if the movement of the camera is substantial enough. Both local and global loop closure detection is employed for post-processing. Finally, the global pose graph is optimized by the g2o framework [32]. Fig. 3 shows the 3D map of a floor at the University of Essex and Fig. 4 shows the 3D map of a student flat.

Perspective-n-Point (PnP), which originates from camera calibration is used to calculate transformation matrices. RANSAC [33] is applied to eliminate outliers during this process. A transformation matrix consisting of a rotation matrix (roll, pitch and yaw) and a translation vector is calculated by pairs of continuous keyframes. Based on the pinhole camera model, a scene view can be formed by projecting 3D points in the world coordinate system into an image 2D plane using the perspective transformation formula

$$s\boldsymbol{p} = \boldsymbol{C}\boldsymbol{P}, \tag{1}$$

where s is the scale factor, $\boldsymbol{C}$ is the camera intrinsic matrix, $\boldsymbol{p}$ is a pixel in the image 2D plane and $\boldsymbol{P}$ is its associated 3D point in the world coordinate system.

The transformation matrix can thus be calculated by

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2}$$

where f_x, f_y, c_x, c_y are the parameters of the camera intrinsic matrix, (u, v) are the 2D coordinates of a point $\boldsymbol{p}_t$ in the current frame, (x, y, z) are the coordinates of its associated 3D point $\boldsymbol{P}_{t-1}$ in the previous frame, $\boldsymbol{R}$, $\boldsymbol{T}$ are the estimated rotation matrix and translation vector.

Equation 2 can also be expressed in a more concise way:

$$s\boldsymbol{p}_t = \boldsymbol{C}\left[\ \boldsymbol{R} \mid \boldsymbol{T}\ \right]\boldsymbol{P}_{t-1}. \tag{3}$$

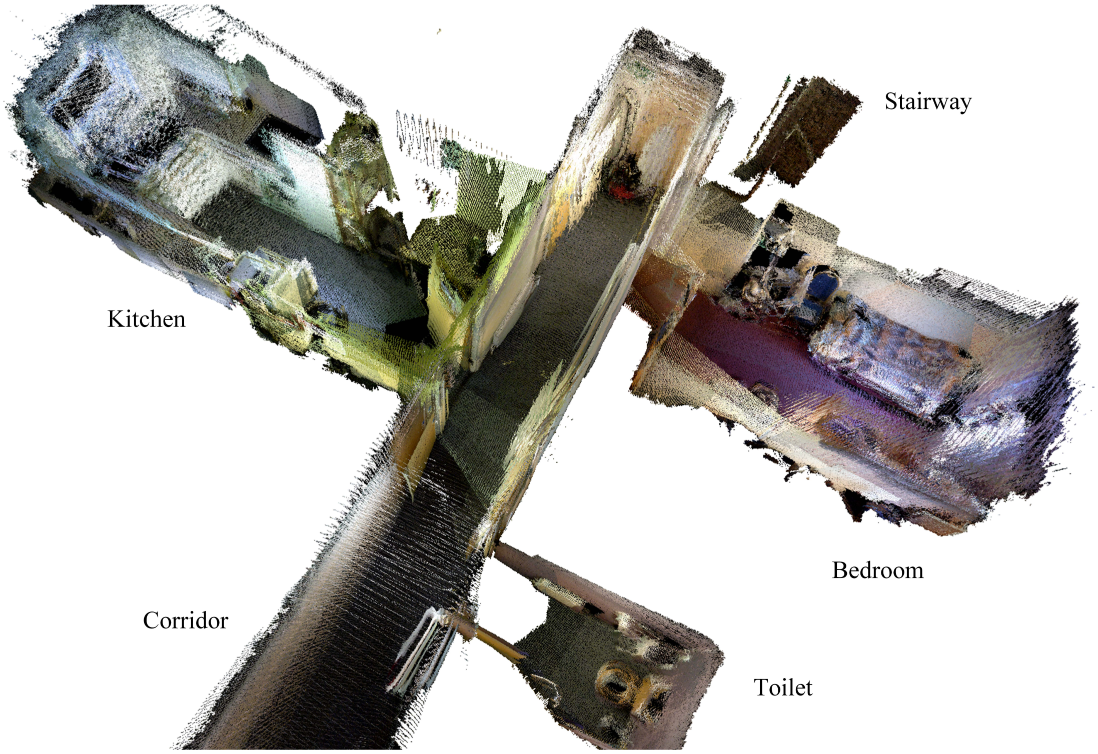


Fig. 4: The 3D map of a student flat.

We then apply loop closure detection with an efficient strategy [4]. Specifically, we first try to detect local loop closure in the neighboring frames of a keyframe. Then we randomly select several much earlier keyframes for global loop closure detection. Once found, the best match among its neighboring keyframes is marked as a successful loop closure.

Finally, the g2o framework is used for global pose graph optimization in order to minimize accumulated errors by

$$\boldsymbol{F}(\boldsymbol{x}) = \sum_{(i,j)\in C} e(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{z}_{ij})^T \boldsymbol{\Omega}_{ij} e(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{z}_{ij}), \qquad (4)$$

where $\boldsymbol{x} = (\boldsymbol{x}_1^T, \cdots, \boldsymbol{x}_n^T)^T$ is the vector of the estimated camera pose, $\boldsymbol{z}_{ij}$ and $\boldsymbol{\Omega}_{ij}$ are the mean and the information matrix of a constraint of pose $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, $e(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{z}_{ij})$ is the error between pose $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Ideally, the error $e(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{z}_{ij})$ is 0 if the estimated poses are absolutely accurate, i.e., equal to the ground truth.

B. Semantic Information Extraction and Representation

The semantic information used in this paper consists of objects and room types. Room types are hand-coded into the database, whereas objects are detected from the aforementioned keyframes. An indoor environmental map constructed by hundreds of keyframes normally contains various objects. If all objects in each keyframe are identified and labeled, our database would be redundant and intractable.

In fact, we are more interested in distinctive objects which can directly infer types of rooms. Moreover, errors inevitably exist during object detection. Thus, the following rules are used for object detection.

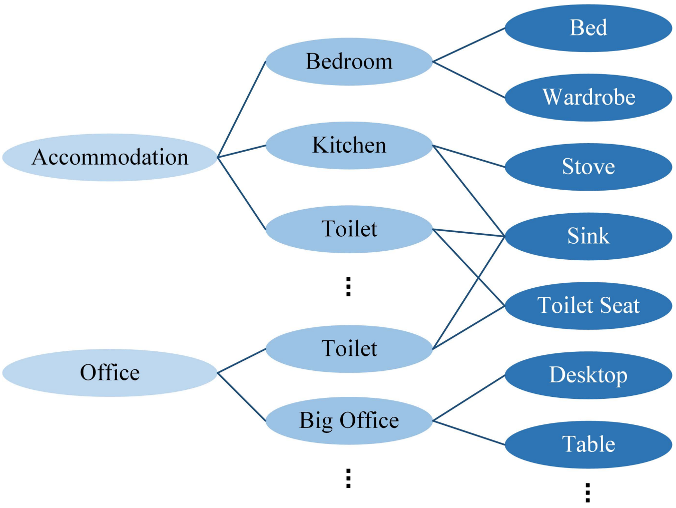


Fig. 5: Semantic information representation.

- Only one object can be detected from each keyframe.
- The score of the detected object needs to exceed a threshold.
- An object can be labeled only if it has been detected in 15 continuous keyframes.

Therefore, semantic information extraction becomes an object recognition problem in our paper since it only identifies whether an image contains a specific object, rather than the location of the object within the image. A pre-trained Inception-v3 model [29] is deployed. The model is trained for the ImageNet competition and can classify objects into 1,000 categories, which is powerful enough for our task.

Finally, the conceptual knowledge is represented with the "has-a" relations [34], as shown in Fig. 5. The straight lines from left to right indicate this relationship. On the other hand, the objects on the right can reveal the associated locations on the left. Note that some objects in our database can infer multiple locations.

C. Topological Localization

In this section, we explain the proposed topological localization method in detail. An overview of our method is shown in Fig. 6. In some cases, distinctive objects in an indoor environment can directly infer types of rooms. However, it is still necessary to deploy a holistic place recognition approach since not all observations contain distinctive objects. In addition, some objects can be discovered at multiple locations. Therefore, relying entirely on object detection for localization is impractical. Thus, a two-stream Convolutional Neural Network is proposed in this paper.

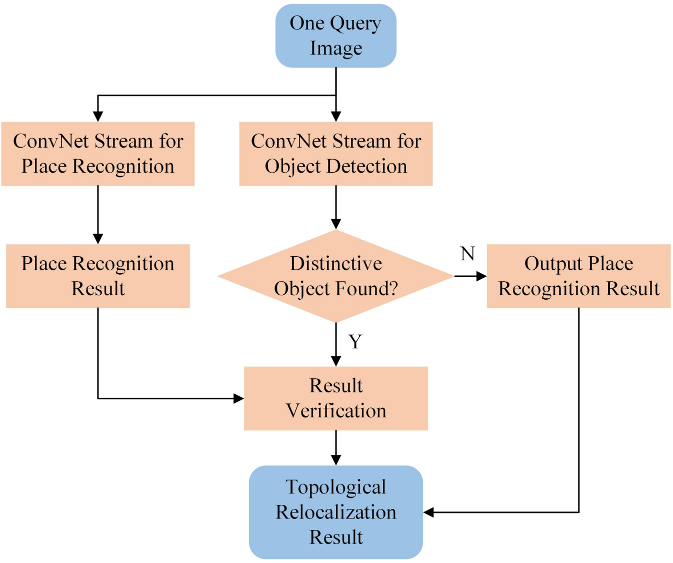

One Query Image
ConvNet Stream for Place Recognition
ConvNet Stream for Object Detection
Place Recognition Result
Distinctive Object Found?
N
Output Place Recognition Result
Y
Result Verification
Topological Relocalization Result

Fig. 6: An overview of the proposed topological localization method.

Once a query RGB image is captured, we directly feed the image into the two ConvNet steams for place recognition and object detection, respectively. A place recognition result consisting of normalized prediction scores for all locations can be obtained. If the object detection score is over a threshold, i.e., a distinctive object in the database is found, we use this additional semantic information to rectify the place recognition prediction scores, which can be viewed as a post-processing step. Otherwise, the place recognition result is taken as the localization final result. The threshold is discussed in the experiment section.

Researches have shown that generic features learned from different ConvNets are transferable. One ConvNet can be retrained and utilized for other recognition tasks. Therefore, we adopt the same Inception-v3 model [29] for both place recognition and object detection. The training methods are detailed in the next section.

If a distinctive object in the database is found, we then use a Bayesian approach to rectify the place recognition scores. Let $\boldsymbol{L}$ be the location vector

$$\boldsymbol{L} = \{l_1, l_2, \ldots, l_n\}, \tag{5}$$

where n is the total number of locations in the database, l_i represents location i. Given a query image x with a distinctive object detected within it, the basic Bayesian inference is applied to estimate the rectified score $P(l_i|x)$

$$P(l_i|x) = \frac{P(l_i)P(x|l_i)}{P(x)}, \tag{6}$$

where $P(l_i)$ is the place recognition score generated from the place recognition ConvNet stream, $P(x)$ is the object detection accuracy, $P(x|l_i)$ is the empirical knowledge. Since the denominator $P(x)$ is identical to all locations, we have

$$P(\boldsymbol{L}|x) \propto P(\boldsymbol{L})P(x|\boldsymbol{L}), \tag{7}$$

in which $P(x|\boldsymbol{L})$ is the empirical probability distribution. The normalized distribution $P(\boldsymbol{L}|x)$ is then considered as the final topological localization result.

Each object in the database has its own empirical probability distribution in terms of all locations. Based on the semantic representation detailed in semantic mapping, assume $\boldsymbol{L}_w = \{l_1, l_2, \ldots, l_p\}$ is a set of locations with a specific labeled object y in them, whereas $\boldsymbol{L}_{wo} = \{l_1, l_2, \ldots, l_q\}$ is a set of locations without this object in them. A ratio is used to obtain the distribution by

$$\xi = \frac{P(y|l_r)}{P(y|l_s)}, \tag{8}$$

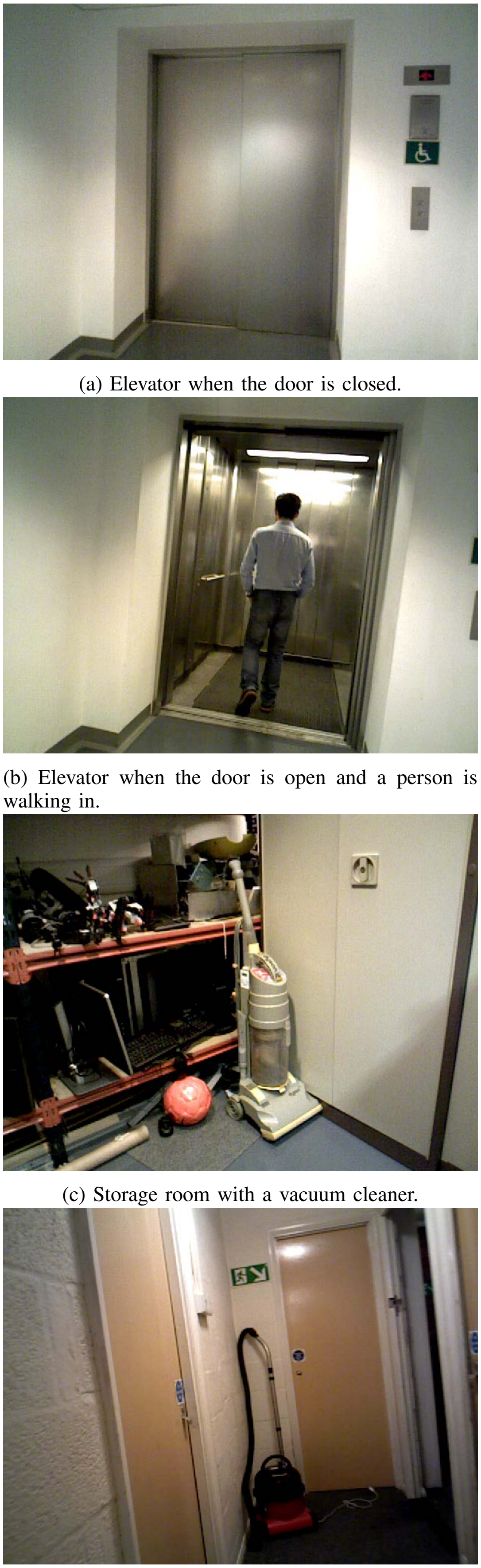
in which ξ is a given factor, $l_r \in \boldsymbol{L}_w$, $l_s \in \boldsymbol{L}_{wo}$.

The factor ξ plays an important role in our system. It controls the weights of the two ConvNet stream outputs. On one hand, we prefer a large value so that the system still performs well even though a location suffers from significant appearance change or human intervention (Fig. 7a and Fig. 7b, sliding door detected). However, the precision drops if the value is too high since object detection errors inevitably exist. Furthermore, objects randomly appear at other locations where they should not belong to also lead to localization errors. For example, although a vacuum cleaner is found in Fig. 7d, the location should still be recognized as "accommodation corridor" rather than "storage room". The factor ξ is further discussed in Section IV.

IV. Experiments

A. System Configuration

This section details how the vision-based cognitive system is configured in our experiments. The system consists of a server, an RGB camera, an Odroid XU3 board and a pair of earphones, as shown in Fig. 1. The server is used for data processing. The user carries the camera, Odroid (with a portable power supplier) and earphones. The camera and earphones are plugged into the Odroid. The Odroid communicates with the server via a wireless connection. Specifically, the Samba file server [35] is installed on the Odroid. The Odroid has a wireless access point (a Wi-Fi dongle in "master" mode),



(a) Elevator when the door is closed.

(b) Elevator when the door is open and a person is walking in.

(c) Storage room with a vacuum cleaner.

(d) Accommodation corridor with a randomly appeared vacuum cleaner.

Fig. 7: Test images showing the importance of the factor.

which allows the Odroid to share files with the server. A person with normal vision needs to build the semantic map of an environment with the mapping approach detailed in Section III-A and III-B. Now we assume the semantic map is already obtained.

The camera captures RGB images and saves them to the Odroid. The server reads the images in the shared folder at a specific frequency, runs the topological localization algorithm and writes the result into a text file on the Odroid. The Odroid then reads the result and plays the pre-recorded audios. This process is shown by the arrows in Fig. 1. The Odroid plays the "you are in ***" audio every 10 seconds if the user remains in the same room. The Odroid plays the "you are entering ***" audio after the localization result has changed over 5 continuous frames.

B. Training

The two-stream ConvNet is trained on a desktop with an Intel Core i7-3370 @3.4GHz CPU and a GeForce GTX 980 GPU. The code is written on the TensorFlow platform [36]. TensorFlow is an open-source software library originating from Google's Machine Intelligence research organization for numerical computation using data flow graphs.

The 2D images used for environment construction are directly deployed to train the place recognition ConvNet. The training dataset contains 20,298 images from 17 locations. We have tried three ways to train the Inception-v3 network. Our first attempt is to train the entire network from scratch with random initialized variables, which is a computationally intensive task. However, we fail to obtain a decent result after training for 3 days since the number of images is not sufficient for Inception-v3.

Our second attempt is to use transfer learning strategy to fine-tune a pre-trained model. The pre-trained model is trained on the ImageNet dataset. We divide our dataset into training, validation and test subsets based on the ratio of 8:2:1. We build a similar model to Inception-v3 with the number of labels in the final classification layer altered to 17. All weights from the pre-trained model are restored except the final classification layer is randomly initialized.

During this process, all previous weights from all layers can be modified. The smoothed curve in Fig. 8 evaluates the model precision against training steps. The training time at step 20,000 is about 8 hours. The precision increases significantly until 14,200 steps and reaches 96.2%, and then starts to drop slightly afterward. At the same time, we find the loss remains steady after 14,200 steps. Thus, the precision drop is caused by over-fitting since the model is too complex for our dataset. Some particular features in the training images that can not be applied generally are memorized by the model.

In order to further reduce the training time, we only retrain the final classification layer from scratch, while leaving all the previous layers untouched. In other words, the previous layers are treated as a fixed feature extractor for our own dataset. This is due to the fact that lower-level portion of a ConvNet generates more generic features that can be deployed for other tasks, whereas top layers contain more specific features of the

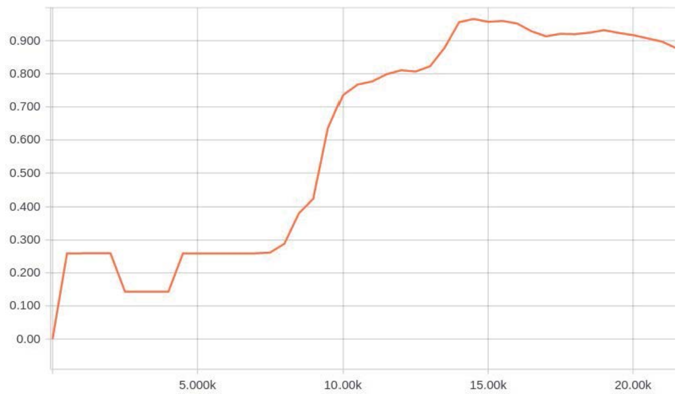


Fig. 8: Precision evaluation when fine-tuning among all layers. X-axis: training steps. Y-axis: precision.

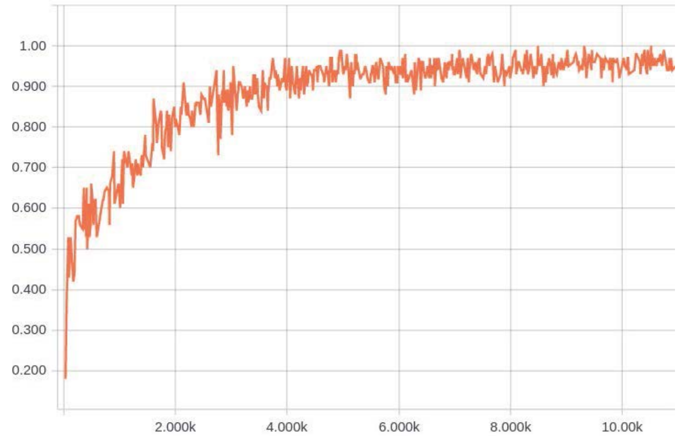


Fig. 9: Precision evaluation when only fine-tuning the final layer. X-axis: training steps. Y-axis: precision.

training dataset. The ratio of the image numbers in training, validation and test subsets is 8:1:1. The initial learning rate is set to a low value so that we can obtain higher overall precision.

We find that the system has the best performance when the learning rate is 0.001. The entire validation subset is used for calculation to reduce the fluctuation among iterations. However, its drawback is longer training time. The unsmoothed curve in Fig. 9 shows the precision of the model against training steps. The training time at step 8,000 is 24 minutes, with average 97.7% precision. If the GPU is not used and the model is trained only on an Intel Core i7-3370 @3.4GHz CPU, the time is 103 minutes. In this case, training is much quicker than fine-tuning among all layers. Moreover, the precision of the trained model is slightly higher.

In terms of the ConvNet stream for object detection, we adopt the pre-trained Inception-v3 model. Since the model classifies objects into 1,000 categories, another linear classifier is added to minimize the number of labels. For example, “bobtail”, “chow chow”, “tabby cat” are merged into “animals”. “Police van”, “shark”, “military plane” are merged into “others”. We have also modified some labels to make them suitable for our task.

TABLE I: Distinctive objects found at each location.

Location	Objects
arena	desk, monitor, tripod, Baxter robot, projector window shade
arena lab	desk, desktop computer, monitor, printer
bedroom	umbrella, running shoe, folding chair, quilt radiator, desk, table lamp, monitor, paper towel backpack, wardrobe, suit
big office	desk, desktop computer, monitor, file cabinet printer
hardware lab	desk, desktop computer, monitor, printer lab chair, oscilloscope
home corridor	corridor
home stairway	banister, handrail
home toilet	washbasin, toilet seat
kitchen	refrigerator, microwave, washbasin, toaster dining table
lecture room	board, desk, folding chair, theater seating
elevator	sliding door
office corridor	sliding door, corridor
office stairway	banister, handrail
office toilet	washbasin, toilet seat
shower room	bathtub, shower curtain, washbasin
small office	desk, desktop computer, monitor, radiator file cabinet, bookcase
storage room	file cabinet, space heater, crutch, mop, desk oscilloscope, croquet ball, project vacuum cleaner, lab chair

C. *Evaluation on Various Appearance Change Conditions*

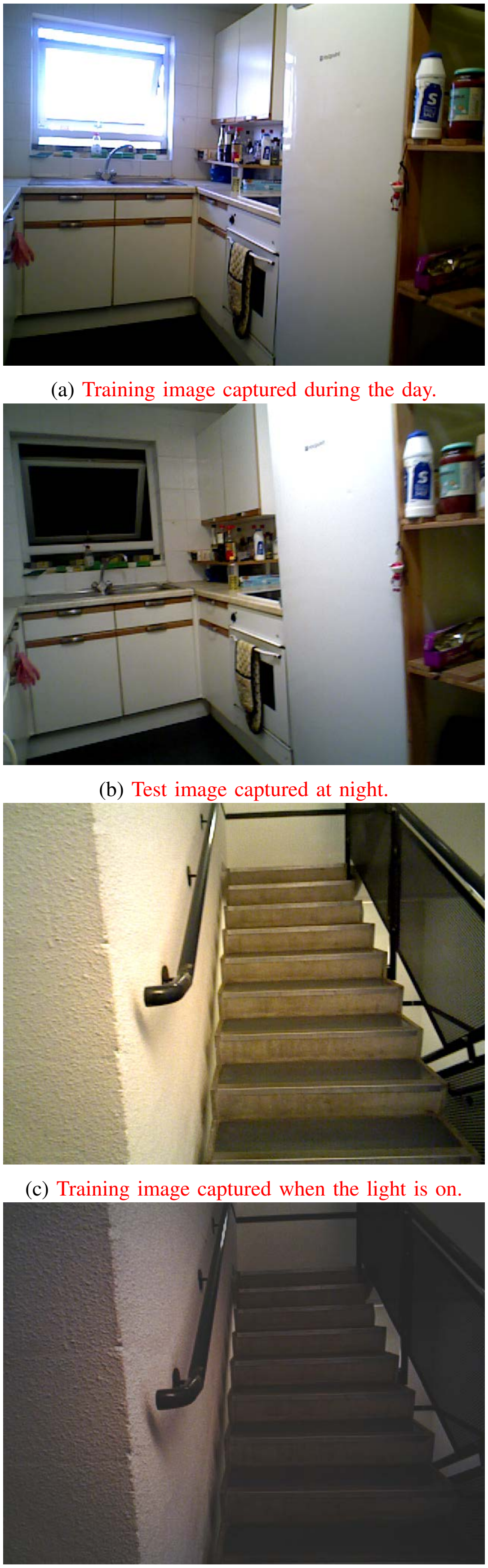
In this section, we evaluate the performance of our model based on various condition changes. We compare the performance of the proposed localization system with other state-of-the-art methods. The test environment (accommodation and office) contains 17 locations in total, among which some of them have similar appearances, such as toilets, corridors, labs and offices.

The objects detected in the process of mapping are listed in Table I. Some objects are unique objects that can be found at only one location, while others can be found at multiple locations.

We captured new images for testing rather than modifying the images in the training dataset. Since the training images are directly obtained from the mapping process and the images are captured from different viewpoints, the training result indicates the localization performance when camera viewpoint changes. Topological localization is similar to place recognition. Thus, precision-recall curves are used for performance evaluation.

1) Change in Lighting Conditions: We first evaluate the influence of lighting condition on the localization performance. The number of test images is 6,875. All objects in the environments remain untouched. The training images are captured during the day, while the test images are captured at night. Regarding the locations where there are no windows or use window shades all the time, we switch some of the lights off to simulate the change in lighting conditions. Examples are shown in Fig. 10.

Since the experiment is carried out in indoor environments, the change in lighting conditions has a minor impact on both methods. From the precision-recall curves in Fig. 11, we can



(a) Training image captured during the day.

(b) Test image captured at night.

(c) Training image captured when the light is on.

(d) Test image captured when the light is off.

Fig. 10: Images captured when lighting conditions change.

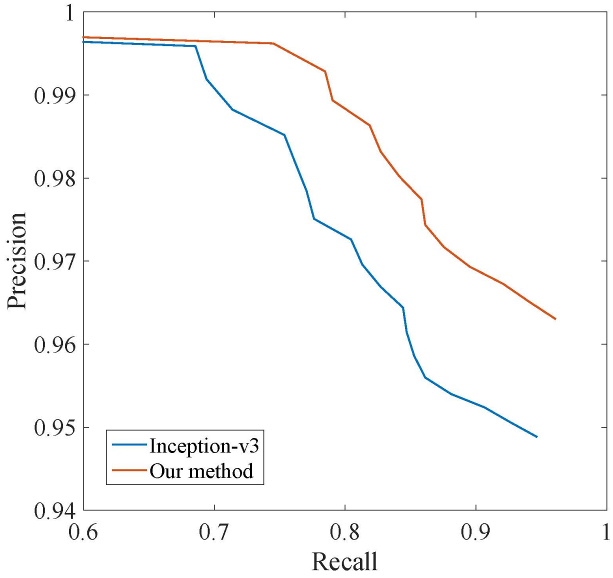
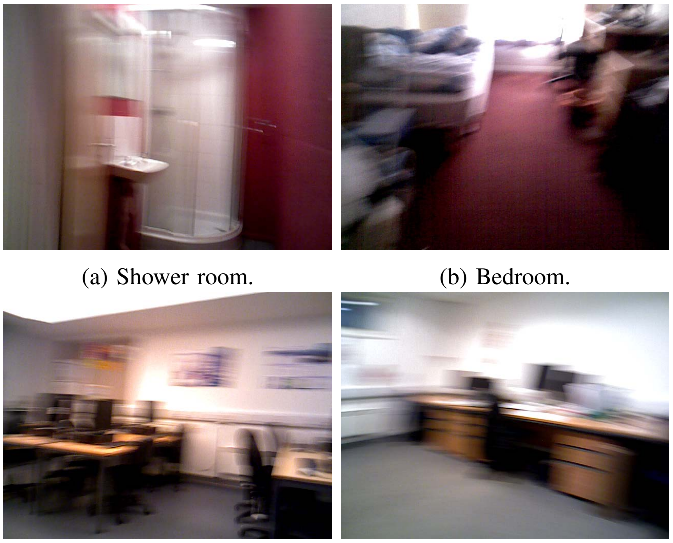


Fig. 11: Model evaluation when lighting conditions change ($\xi = 1.3$).

see our method performs slightly better than the Inception-v3 model. Based on the entire test dataset, our method results in a 96.3% localization accuracy with the maximum recall rate of 96.0%. Some errors are caused by different shapes of shadows captured.

2) Blur Images: When a camera is placed on a robot or a wearable device, we can not guarantee all captured images to be sharp at all times. If the sensor is moving or rotating at a high speed, blur images are inevitably generated. In this experiment, we test the robustness of our method to these images. There are 2316 blur images captured during the day for testing. Some of them are shown in Fig. 12. Fig. 13 shows both methods perform poorly in this experiment. The two



(a) Shower room.

(b) Bedroom.

(c) Arena lab.

(d) Small office.

Fig. 12: Blur images captured for testing.

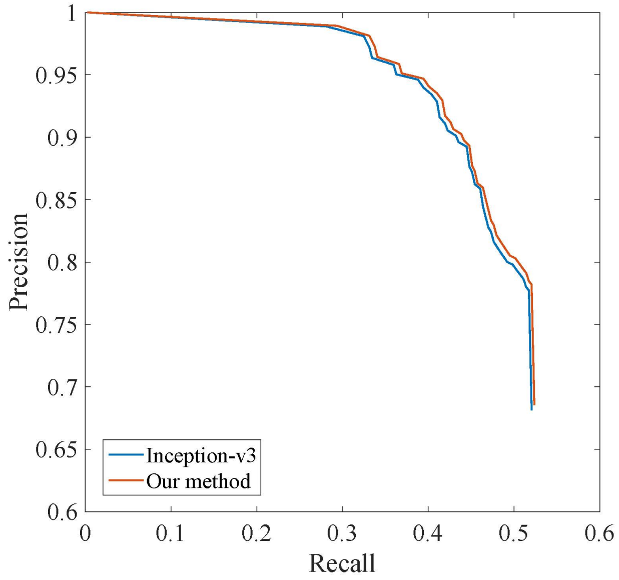


Fig. 13: Model evaluation when blur images exist ($\xi = 1.3$).

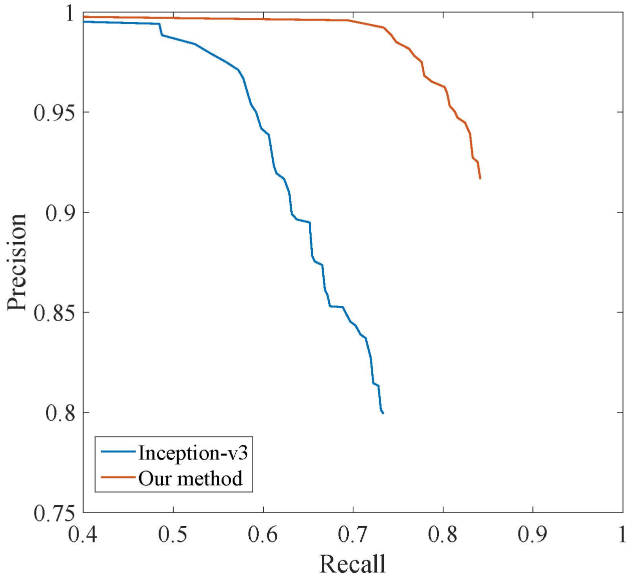
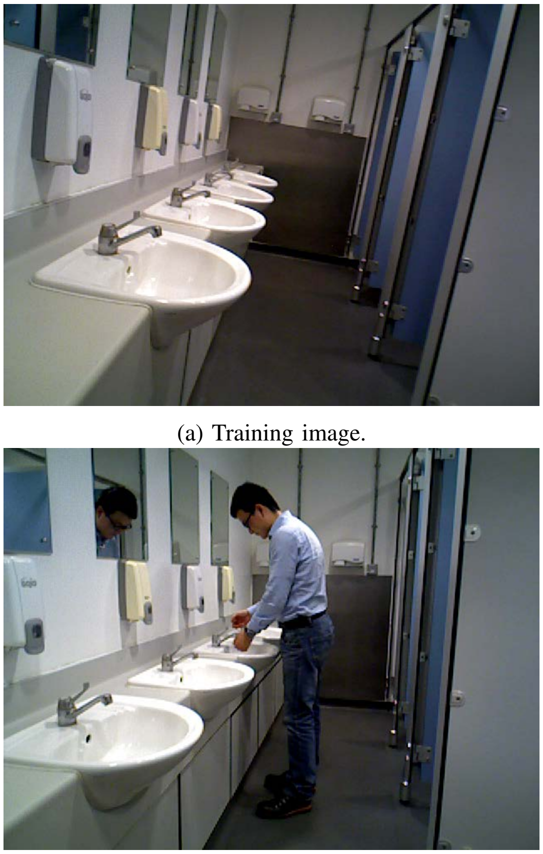


Fig. 14: Model evaluation when object location changes ($\xi = 1.3$).

curves are almost coincident. The reason is that the object detection scores are not high enough to exceed the threshold.

3) Change of Object Locations: The change of object locations usually causes significant appearance change in indoor environments. In this experiment, 5,208 images are used for testing and the following conditions are considered.

- The locations of objects (chair, kitchen utensil, vacuum cleaner, clothes, elevator door, etc.) are changed.
- The deformation of some objects, such as curtain, window shade and quilt.
- New facilities or appliances are installed, such as the stove and oven in the newly refurbished kitchen.
- Randomly appeared humans.



(a) Training image.

(b) Test image.

Fig. 15: Example of how localization results can be rectified by a distinctive object.

The images in Fig. 15 illustrate how a topological localization result can be rectified by semantic information in spite of human intervention. The test image is incorrectly identified as "home corridor" by the place recognition stream. However, a distinctive object "washbasin" is detected. Thus, the localization result is rectified as "office toilet". Fig. 14 shows the performance of these two methods. The precision of Inception-v3 starts to drop significantly from the recall rate of 48%, whereas our method drops from 70%. The evaluation on the entire test dataset shows that the precision of Inception-v3 is 79.9% with the maximum recall rate of 73.4%, while our method results in 91.7% precision with the maximum recall rate of 84.1%.

D. Factor ξ

The factor ξ plays an important role. Generally speaking, it controls how much the object detection stream is involved in our system. In this section, we evaluate ξ based on the dataset used for object location change evaluation. Results are shown in Fig. 16. When $\xi = 1.0$, the output from the place recognition stream actually remain unchanged. Thus, the curve is the same as the one generated by Inception-v3. We start to raise the value of ξ from 1.1. Both precision and recall rates increase with the value of ξ, which means the object detection stream starts to work and the distinctive objects detected start

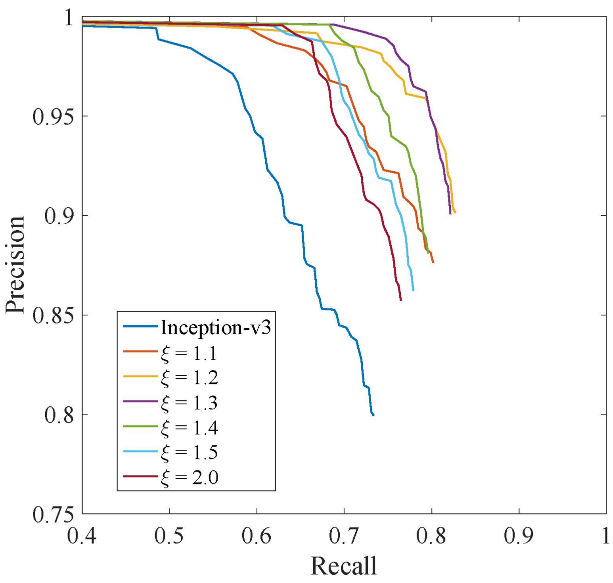


Fig. 16: Model evaluation when factor ξ changes.

to rectify the localization results. The precision and recall reach their peak values when $\xi = 1.3$.

However, if we continue raising the value of ξ, both the precision and recall rates begin to drop. This is due to the object detection errors. In addition, distinctive objects in certain rooms which somehow randomly appear at other locations also contribute to localization errors. We have also carried out some tests when $\xi > 2$ and found that the curves are all similar to the curve produced by $\xi = 2$. But all of them performs better than Inception-v3. Therefore, $\xi = 1.3$ is used in all the aforementioned experiments.

E. Comparison with Other Algorithms

In this section, we compare the proposed algorithm to other state-of-the-art algorithms, namely VGG-16 [37] and Places365-ResNet [38]. The test dataset used for object location change evaluation is again employed here. Transfer learning is applied to all models. Inception-v3, VGG-16 and our model are pre-trained on ImageNet and fine-tuned, whereas Places365-ResNet is pre-trained on Places365 [38], which is a 10 million image dataset for scene recognition. The precision-recall curves are shown in Fig. 17.

Places365-ResNet generally performs better than Inception-v3 and VGG-16, both of which have similar performance in terms of object location changes. This is due to the fact that Places365-ResNet is pre-trained on a dataset of scene photographs. Thus, it contains more generic visual features for place recognition. However, the proposed model still outperforms Places365-ResNet even though ours is pre-trained on a dataset of object photographs.

F. Time Complexity

In this section, we evaluate the time complexity on a desktop computer with an Intel Core i7-3370 @3.4GHz CPU, a GeForce GTX 980 GPU and 16GB RAM. The captured

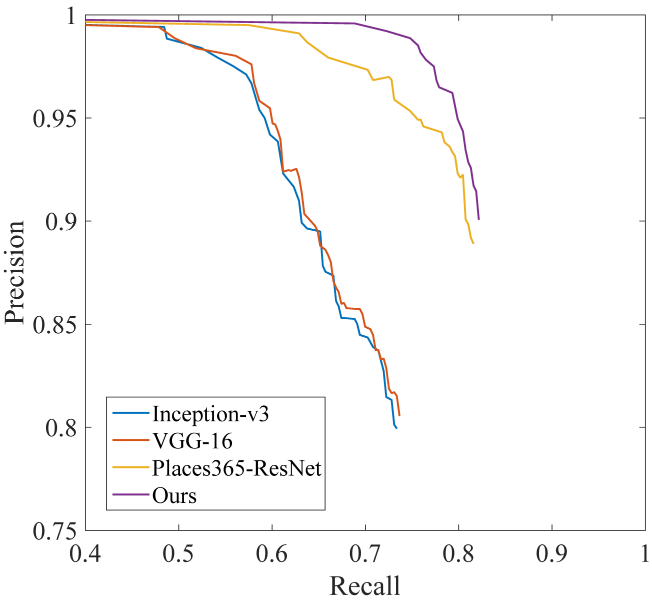


Fig. 17: Model performance compared to Inception-v3, VGG-16 and Places365-ResNet.

image size is $640 \times 480 \times 3$ and then resized to $299 \times 299 \times 3$. Each batch has 32 images. We have also tested the processing time without using the GPU. Results are presented in Table II. Compared to Inception-v3, our method costs more than twice the processing time.

TABLE II: The average processing time of one image. Unit: second.

	Processing Time
Our method with GPU	0.079
Our method without GPU	3.432
Inception-v3 with GPU	0.037
Inception-v3 without GPU	1.492
VGG with GPU	0.081
Places365-ResNet with GPU	0.046

V. Conclusion

We have proposed a vision-based assistive system to help visually impaired people move around independently. The system can provide audio prompts when the user visits an unfamiliar environment. A two-stream ConvNet is proposed for topological localization. Semantic information is used to further rectify the localization result by detecting distinctive objects within the environment. The performance of our system is evaluated in terms of various appearance changes in two indoor environments. Experimental results show that both the precision and recall rates are improved over Inception-v3. But our model is less computationally efficient. The proposed localization approach can also be applied to a mobile robot for its indoor navigation.

In the future, we will continue improving the reliability and real-time performance of the proposed system. Recurrent Neural Networks will be incorporated for higher accuracy.

Acknowledgments

Our thanks go to Robin Dowling and Ian Dukes for their technical support.

Compliance with Ethical Standards

Funding

No funding applicable.

Conflict of Interest

The authors declare that they have no conflict of interest.

Ethical Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Informed Consent

Informed consent was obtained from all individual participants included in the study.

References

[1] The World Health Organization - Visual impairment and blindness. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs282/en/

[2] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler, "Semantic visual localization," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6896–6906.

[3] Q. Liu, R. Li, H. Hu, and D. Gu, "Extracting semantic information from visual data: A survey," *Robotics*, vol. 5, no. 1, p. 8, 2016.

[4] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D mapping with an RGB-D camera," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 177–187, 2014.

[5] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[6] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.

[7] R. Biswas, B. Limketkai, S. Sanner, and S. Thrun, "Towards object mapping in non-stationary environments with mobile robots," in *2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1. IEEE, pp. 1014–1019.

[8] M. Brucker, M. Durner, R. Ambruş, Z. C. Márton, A. Wendt, P. Jensfelt, K. O. Arras, and R. Triebel, "Semantic labeling of indoor environments from 3d rgb maps," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1871–1878.

[9] S. Garg, N. Suenderhauf, and M. Milford, "Don't look back: Robustifying place categorization for viewpoint-and condition-invariant place recognition," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3645–3652.

[10] T.-j. Lee, C.-h. Kim, and D.-i. D. Cho, "A monocular vision sensor-based efficient slam method for indoor service robots," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 1, pp. 318–328, 2019.

[11] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[12] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison, "CodeSLAM—learning a compact, optimisable representation for dense visual SLAM," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2560–2568.

[13] R. Li, S. Wang, Z. Long, and D. Gu, "Undeepvo: Monocular visual odometry through unsupervised deep learning," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7286–7291.

[14] Q. Liu, R. Li, H. Hu, and D. Gu, "Using unsupervised deep learning technique for monocular visual odometry," *IEEE Access*, vol. 7, pp. 18 076–18 088, 2019.

[15] R. Li, S. Wang, and D. Gu, "Ongoing evolution of visual slam from geometry to deep learning: Challenges and opportunities," *Cognitive Computation*, vol. 10, no. 6, pp. 875–889, 2018.

[16] H. Grimmett, M. Buerki, L. Paz, P. Pinies, P. Furgale, I. Posner, and P. Newman, "Integrating metric and semantic maps for vision-only automated parking," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2015, pp. 2159–2166.

[17] J. W. Hart, R. Shah, S. Kirmani, N. Walker, K. Baldauf, N. John, and P. Stone, "PRISM: Pose Registration for Integrated Semantic Mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 896–902.

[18] D. Maturana, P.-W. Chou, M. Uenoyama, and S. Scherer, "Real-time semantic mapping for autonomous off-road navigation," in *Field and Service Robotics*. Springer, 2018, pp. 335–350.

[19] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, and E. Romera, "Are you able to perform a life-long visual topological localization?" *Autonomous Robots*, vol. 42, no. 3, pp. 665–685, 2018.

[20] Y. Li, Z. Hu, Y. Hu, and D. Chu, "Integration of vision and topological self-localization for intelligent vehicles," *Mechatronics*, vol. 51, pp. 46–58, 2018.

[21] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an RGB-D camera," in *Robotics Research*. Springer, 2017, pp. 235–252.

[22] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 806–813.

[23] G. Wen, Z. Hou, H. Li, D. Li, L. Jiang, and E. Xun, "Ensemble of deep neural networks with probability-based fusion for facial expression recognition," *Cognitive Computation*, pp. 1–14, 2017.

[24] Z. Yue, F. Gao, Q. Xiong, J. Wang, T. Huang, E. Yang, and H. Zhou, "A novel semi-supervised convolutional neural network method for synthetic aperture radar image recognition," *Cognitive Computation*, pp. 1–12, 2019.

[25] P. Ren, W. Sun, C. Luo, and A. Hussain, "Clustering-oriented multiple convolutional neural networks for single image super-resolution," *Cognitive Computation*, pp. 1–14, 2017.

[26] R. Li, D. Gu, Q. Liu, Z. Long, and H. Hu, "Semantic scene mapping with spatio-temporal deep neural network for robotic applications," *Cognitive Computation*, pp. 1–12, 2017.

[27] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun, "3D object proposals using stereo imagery for accurate object class detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1259–1272, 2018.

[28] N. Sunderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the performance of ConvNet features for place recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 4297–4304.

[29] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," *arXiv preprint arXiv:1512.00567*, 2015.

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[31] Z. Zeng, Z. Li, D. Cheng, H. Zhang, K. Zhan, and Y. Yang, "Two-stream multirate recurrent neural network for video-based pedestrian reidentification," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3179–3186, 2018.

[32] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2011, pp. 3607–3613.

[33] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[34] H. Zender, O. M. Mozos, P. Jensfelt, G.-J. Kruijff, and W. Burgard, "Conceptual spatial representations for indoor mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 493–502, 2008.

[35] Samba - opening windows to a wider world. [Online]. Available: https://www.samba.org/

[36] TensorFlow. [Online]. Available: https://www.tensorflow.org/

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv: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.