Localized Vision-Based Navigation Aid for the East Bay campus

1) Goal

When people normally navigate without the use of a Google Maps type app, they usually do so with visually recognizable landmarks. When a user is either unfamiliar with the local landmarks or is visually impaired, navigation becomes significantly more challenging. Our goal is to help people who are either visually impaired or new to campus navigate the map through a vision-based aid.

Our goal is to train a model to accurately recognize landmarks around the CSU East Bay Campus, load it into an app, then from a device camera's live preview frame use the landmarks in vision to attempt to locate the user on the campus and provide their relative location and orientation (lat/long, compass direction) on campus. When indoors or without obvious landmarks, our model will attempt a best guess.

1.1) Input

The input to our CNN model training will be a significant number of labelled and geotagged images of the landmarks around campus. We'll go around campus, take ~25 pictures of each building from a variety of angles (while sticking to the 'normal' pathways around campus), then label each picture with the landmark and its name. This will train both the CNN landmark detector as well as the feedforward NN for geolocation.

The input to the phone app will be the device's camera live preview frame as they walk around the campus. We will likely stick to locating people on the pathways around campus and the interiors of the more popular buildings, instead of anywhere campus visitors wouldn't ordinarily be.

1.2)Output

The CNN model's output will be the landmarks detected in the camera buffer, while the feedforward NN will output a proposed geolocation. The app will present both drawn boxes over the camera frame boxing the landmarks in vision as well as a miniature map overlay of the CSUEB campus, with lat/long values and compass orientation overlaid in text.

2)Development Platform and Target AVD and Device

We will target API 32, Android 11.0, using OpenCV version 4.6.0 in Android Studio and will test on Samsung Galaxy S8, Google Pixel 3a, and Samsung Galaxy A20.

3) Algorithms

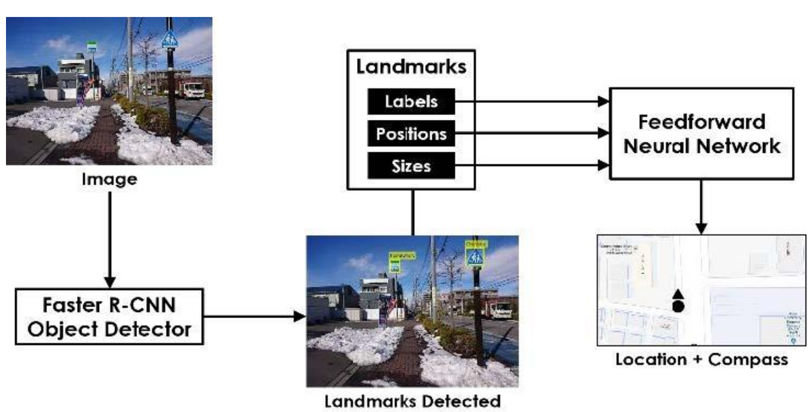
In this proposed method, Faster RCNN deals with the detection of landmarks in the image from the camera, while FFNN utilizes the detected landmarks from the Faster R-CNN and gives the current location and direction of the device. The training architecture proposed by Nilwong, etc from Hosei University looks like a promising initial direction.

We will train both the Faster RCNN and FFNN models and load them in a pipeline. From the device camera buffer, we will attempt to infer landmarks from the RCNN. If a landmark is found, there is a good chance we will get a close match to one of the reference images and specific angles in the training set from the FFNN's geolocation. If no landmarks are found, we will have to best-guess based off of generic features in the camera buffer, matching against the no-landmark images in the training set. Given the experience of both the mail.ru and the CERTH teams, this is unlikely to be promising at the hyperlocal level (the parking lots probably look pretty similar).

3.1) Datasets

Dataset used in the training of the Faster R-CNN and FFNN will be the set of around 500-600 images with geotags and labeled bounding boxes, composing representative images of the buildings around campus (as well as some pictures taken without any landmarks in view), taken from the ordinary pathways around campus. The images will be taken by a smartphone camera in the California State University, East Bay Campus, and resized appropriately for training. Images will be tagged with geotags at the time taken by the camera. Implemented geotags include latitude angle, latitude reference, longitude angle, longitude reference, and magnetic-referenced image direction (compass orientation). The compass orientation has the value ranges from 0 to 359.99, where 0 is the absolute north direction, and the orientation changes clockwise as the value increases. All images in the dataset will be hand-labeled with bounding boxes of classes of landmarks.

3.2) System Design



3.3) Experimentation

The experiments done by Hosei University were split into two parts: detection experiments and localization experiments. Detection experiments focused on landmark detection using Faster R-CNN, and localization experiments were about using the landmark detection results on FFNN to get localization results.

We will be doing something similar to test both our combined model's landmark recognition F1 as well as its relative localization accuracy.

(a) (b) (c)

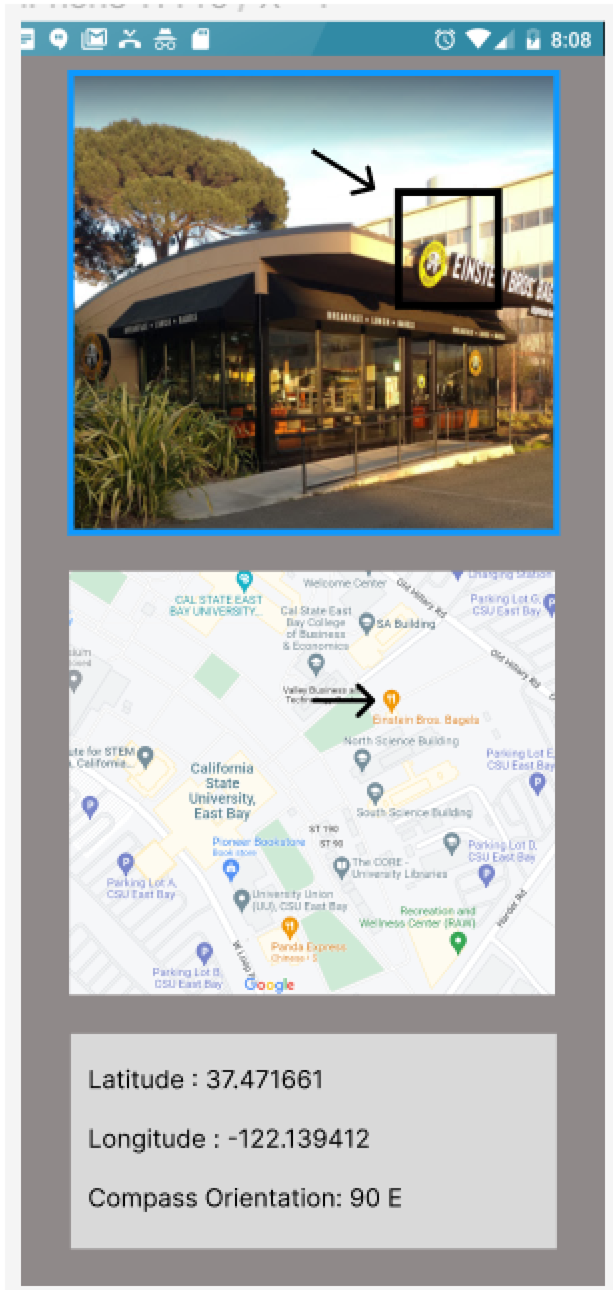
Figure 1

(a) (b) (c)

Figure 2

4) GUI Interface

We intend to make a simple, one-screen Android app. The initial screen should be a live camera view on the top half of the phone, with boxes drawn on the screen to indicate recognized landmarks. The bottom of the screen will show a miniature map of the CSUEB campus with a directional marker placed where we think the user is, as well as three text fields for latitude, longitude, and compass orientation in degrees.



5) References

Andrei Boiarov and Eduard Tyantov. 2019. Large Scale Landmark Recognition via Deep Metric Learning. In The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357956

Giorgos Kordopatis-Zilos, Panagiotis Galopoulos, Symeon Papadopoulos, Ioannis Kompatsiaris. 2021. Leveraging EfficientNet and Contrastive Learning for Accurate Global-scale Location Estimation. In Proceedings of the 2021 International Conference on Multimedia Retrieval (ICMR '21), August 21–24, 2021, Taipei, Taiwan. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3460426.3463644

Sivapong Nilwong, Delowar Hossain, Shin-ichiro Kaneko, and Genci Capi. 2018. Outdoor Landmark Detection for Real-World Localization using Faster R-CNN. In Proceedings of the 6th International Conference on Control, Mechatronics and Automation (ICCMA 2018). Association for Computing Machinery, New York, NY, USA, 165–169.
https://doi-org.proxylib.csueastbay.edu/10.1145/3284516.3284532