The University of Melbourne
School of Computing and Information Systems
COMP90086 Computer Vision, 2021 Semester 2

Final Project: Fine-grained localisation

| | |
|---|---|
| **Project type:** | Group (teams of 2) |
| **Due:** | 7pm, 22 Oct 2021 |
| **Submission:** | Source code and written report (as .pdf) |
| **Marks:** | The assignment will be marked out of 30 points, and will contribute 30% of your total mark. |

Geolocation is the problem of localising a person or device in the world using sensor data. Depending on the device, the environment, and the level of accuracy required, geolocation may rely on GPS coordinates, network routing addresses, or image data. Geolocation is an important problem in many AI and computing applications, from autonomous vehicle navigation to search engine queries based on the user's current location (e.g., "restaurants near me").

在此 image information 是指?

In this project, you will investigate the problem of fine-grained geolocation in a small indoor/outdoor environment (an art museum). Image information is particularly important for this type of problem, because other sources of information, like GPS, may not be accurate enough to provide fine-grained position data and may not be able to distinguish between different floors in indoor environments.

Your task is to develop a method to recognize the location from which an image was taken. You will be provided a dataset of images with position data to train your method. How you approach the problem is up to you. The following are some possible approaches:

不太理想

- Match each image in the test set to the most similar image in the training set, using any visual features you wish to measure "similarity," and assume the test image has the same position data as its closest match. (Note that there is no guarantee that the test images will come from exactly the same locations as the images in training set, but since they come from the same museum environment they are likely to be from nearby locations.)
- Identify key features, objects, or text in the test images and use these to locate training images which show the same features, objects, or text.
- 比1好 ⇒ Match each image in the test set to multiple near neighbours in the training set, and develop a method to compute the test image's most likely location based on multiple nearby views.
- feature matching ⇒ Use matching features and geometric constraints to compute the likely change in pose between training and test views.
- Or any combination of the above.

Note that these are only suggestions to help you get started; you are free to use your own ideas.

Whatever methods you choose, you are expected to evaluate these methods using the provided data, to critically analyse the results, and to justify your design choices in your final report. Your evaluation should include error analysis, where you attempt to understand where your method works well and where it fails.

You are encouraged to use existing computer vision libraries your implementation. You may also use existing models or pretrained features as part of your implementation. However, your method should be your own; you may not simply submit an existing model for this problem.

## Dataset

Figure 1: Example training images

The dataset is a collection of images taken in and around an art museum (the Getty Center in Los Angeles, U.S.A.). Example images are shown in Figure 1. The dataset is split into 7500 training images and 1200 test images. Each image in the training set is annotated with positional data, which is an (x,y) value derived from a mapping algorithm. You can assume that the (x,y) values accurately reflect position in the real world, although the units of these values are unknown. The training dataset includes multiple views from each of several locations around the museum. Different views from the same location are denoted with a suffix (e.g., “_1”, “_2”, etc.).

FoV is fixed ⇒ then focal length should also be fixed

The images are rendered from Google Streetview images, simulating a camera with a 73.7 deg horizontal x 53.1 deg vertical field of view. The optical centre of the camera is in the centre of the image and the lens has no radial distortion. However, because the images are simulated from Google Streetview imagery, they may contain artefacts or distortion from the Streetview panorama stitching process. Faces in the images have been blurred for privacy. Please note that because the images were collected in a real-world public environment, it is possible that they may contain inappropriate or offensive content.

## Scoring Predictions

You should submit your predictions for the test images on Kaggle. Your submissions for Kaggle should follow the same format as the `train.csv` annotation file, with three columns: `id,x,y`. `id` should be a string corresponding to a test image name, and `x` and `y` should be the predicted position of that image.

The evaluation metric for this competition is the mean absolute error in x and y computed on the test set. This can also be thought of as the Manhattan distance between the true and predicted (x,y) coordinates, averaged over all N test images:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} abs(x_i - \hat{x}_i) + abs(y_i - \hat{y}_i)$$

(Although Euclidean distance would probably make more sense for this task, Kaggle does not have an evaluation metric which computes Euclidean distance.)

### Kaggle

To join the competition on Kaggle and submit your results, you will need to register at `https://www.kaggle.com/`.

Please use the "Register with Google" option and use your @student.unimelb.edu.au email address to make an account. Please use *only* your group member student IDs as your team name (e.g., "1234&5678"). Submissions from teams which do not correspond to valid student IDs will be treated as fake submissions and ignored.

Once you have registered for Kaggle, you will be able to join the COMP90086 Final Project competition using the link under **Final Project: Code** in the Assignments tab on the Canvas LMS. After following that link, you will need to click the "Join Competition" button and agree to the competition rules.

### Group Formation

Done

You should complete this project in a group of 2. You are required to register your group membership on Canvas by completing the "Project Group Registration" survey under "Quizzes." You may modify your group membership at any time up until the survey due date, but after the survey closes we will consider the group membership final.

### Submission

Submission will be made via the Canvas LMS. Please submit your code and written report separately under the **Final Project: Code** and the **Final Project: Report** links on Canvas.

Your **code** submission should include your model code, your test predictions (in Kaggle format), a readme file that explains how to run your code, and any additional files we would need to recreate your results. You should not include the provided train/test images in your code submission, but your readme file should explain where your code expects to find these images.

Your written **report** should be a .pdf that includes the description, analysis, and comparative assessment of the method(s) you developed to solve this problem. The report should follow the style of a short conference paper with **no more than four A4 pages** of content (excluding references, which can extend to a 5th page). The report should follow the style and format of an IEEE conference short paper. The **IEEE Conference Template** for Word, LaTeX, and Overleaf is available here: https://www.ieee.org/conferences/publishing/templates.html.

Your report should explain the design choices in your method and justify these based on your understanding of computer vision theory. You should explain the experimentation steps you followed to develop and improve on your basic method, and report your final evaluation result. Your method, experiments, and evaluation results should be explained in sufficient detail for readers to understand them without having to look at your code. You should include an error analysis which assesses where your method performs well and where it fails, provide an explanation of the errors based on your understanding of the method, and give suggestions for future improvements. Your report should include tables, graphs, figures, and/or images as appropriate to explain and illustrate your results.

## Evaluation

Your submission will be marked on the follow grounds:

| Component | Marks | Criteria |
|---|---|---|
| Report writing | 5 | Clarity of writing and report organisation; use of tables, figures, and/or images to illustrate and support results |
| Report method and justification | 10 | Correctness of method; motivation and justification of design choices based on computer vision theory |
| Report experimentation and evaluation | 10 | Quality of experimentation, evaluation, and error analysis; interpretation of results and experimental conclusions |
| Kaggle submission | 3 | Kaggle performance |
| Team contribution | 2 | Group self-assessment |

The report is marked out of 25 marks, distributed between the writing, method and justification, and experimentation and evaluation as shown above.

In addition to the report marks, up to 3 marks will be given for performance on the Kaggle leaderboard. To obtain the full 3 marks, a team must make a Kaggle submission that performs reasonably above a simple baseline. 1-2 marks will be given for Kaggle submissions which perform at or only marginally above the baseline, and 0 marks will be given for submissions which perform at chance. Teams which do not submit results to Kaggle will receive 0 performance marks.

Up to 2 marks will be given for team contribution. Each group member will be asked to provide a self-assessment of their own and their teammate's contribution to the group project, and to mark themselves and their teammate out of 2 (2 = contributed strongly to the project, 1 = made a small contribution to the project, 0 = minimal or no contribution to the project). Your final team contribution mark will be based on the mark assigned to you by your teammate (and their team contribution mark will be based on the mark you assign to them).

### Late submission

The submission mechanism will stay open for one week after the submission deadline. Late submissions will be penalised at 10% of the total possible mark per 24-hour period after the original deadline. Submissions will be closed 7 days (168 hours) after the published assignment deadline, and no further submissions will be accepted after this point.

### Updates to the assignment specifications

If any changes or clarifications are made to the project specification, these will be posted on the LMS.

### Academic misconduct

You are welcome — indeed encouraged — to collaborate with your peers in terms of the conceptualisation and framing of the problem. For example, we encourage you to discuss what the assignment specification is asking you to do, or what you would need to implement to be able to respond to a question.

However, sharing materials — for example, showing other students your code or colluding in writing responses to questions — or plagiarising existing code or material will be considered cheating. Your submission must be your own original, individual work. We will invoke University's Academic Misconduct policy (`http://academichonesty.unimelb.edu.au/policy.html`) where inappropriate levels of plagiarism or collusion are deemed to have taken place.