

Fine-grained Localisation

1st Chen-An Fan

*School of Computing and Information Systems
The University of Melbourne
Melbourne, Australia
chenanf@student.unimelb.edu.au*

1st Hailey Kim

*School of Computing and Information Systems
The University of Melbourne
Melbourne, Australia
heewonk@student.unimelb.edu.au*

Abstract—Location recognition is a hot topic in computer vision. In this work, we introduce several approaches to identify fine-grained locations from 2D images. The methods firstly shortlist the candidate images based on the similarity of the extracted features from pre-trained ViT [1] to speed up the process. After then, they select the best match image sharing the most number of confident matching points from pre-trained LoFTR [3]. The geographic information of the best image represents the input image location. Different approaches involve different techniques such as the image transformation or the result adjustment for better accuracy. We evaluate the performance of the different methods through the experiments and show they can plausibly recognise the positions with unseen images.

Index Terms—location recognition, feature matching, image transformation

I. INTRODUCTION

Nowadays, we are able to search our accurate location with GPS device and even know where a photo was taken from its metadata. However, what if we do not have the sensor data with the image. Can we recognise the geolocation? The image matching technique can solve the problem from other images taken nearby with the known position. There are recently proposed algorithms to find interest points; SuperGlue [4], LoFTR [3] and etc. With these algorithms, applications are various in object recognition, panorama stitching, and 3D reconstruction. In particular, it can also identify locations without GPS-like geographical information. In this project, we propose multiple methods using feature extraction, feature matching, and image transformation theory based on some state-of-the-art models to address the localisation problem. The challenges are to predict the fine-grained position of a photograph by only relying on the image features and to deal with the computational cost with the thousands of images.

II. DATASET

The total number of 7,500 training images and 1,200 test images are provided. Each of which are taken at an art gallery in 680 (W) x 490 (H) pixels. Training images have geographic information, x and y, values from a mapping algorithm. For simplicity, we assume there is no radial distortion and ignore other artefacts and distortion on the images. In order to select the best model, we compare the performance on initially split validation dataset that contains 750 images from the original training images.

III. EVALUATION METRIC

To evaluate the performance of the model, we compute the Mean Absolute Error (MAE) between the predicted and true coordinates as follows (hat indicates predicted coordinates).

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

IV. APPROACH

A. Similarity by ViT

Matching every pair of images from the given training and test dataset is extremely costly in terms of computation time and resource. It can be a major bottleneck in feature matching tasks due to the large number of the images, and hence we compute the similarity scores between train and test image pairs for selective inspection. The scores are calculated by comparing the image features. In this project, we exploit a newly introduced image classifier, Vision Transformer (ViT) [1], to extract the features.

Since the next steps use the more developed feature-focusing matching algorithm, we aim to shortlist the candidate images from the entire scene understanding in this early stage. Research [2] proves that the vision transformer aggregates more global information than convolution neural network (CNN) in the early layer due to the self-attention mechanism, and thus the feature extracted from the ViT has better representation on the holistic scene. It justifies the use of ViT for feature extraction so that we find the relevant scenes while discard irrelevant images. We use the encoded 1D features extracted from ViT to measure the cosine similarity between two images.

After then, we rank the matching images according to the similarity to shortlist the meaningful pairs for further examines. It results in a dramatic decrease in the processing cost from 7,500 x 1,200 to N x 1,200 if we match a test image with the top N similar train images. In the following sections, we use a term "ViT-similar images" to represent these top N similar train images by ViT.

B. Point Matching by LoFTR

LoFTR [3] is a transformer-based detector-free model for local feature matching. It exploits the global receptive field of the transformer by using self- and cross-attention layers for

producing dense matches in low-texture areas. This detector-free approach is proved to outperform the traditional detector-based methods on indoor images that have large illumination variation or low texture [3]. Detector-based methods usually cannot match enough feature points on those indoor images. Moreover, compared with CNN-based feature matching methods which have limited receptive field, the transformer-based method can have a global receptive field (see Fig 1) and therefore LoFTR has the ability to distinguish indistinct local regions by referencing the global context [3].

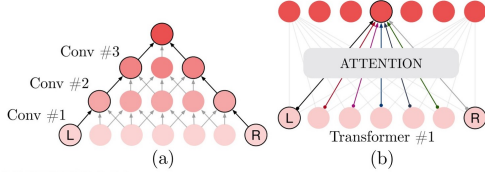


Fig. 1. Receptive field comparison: (a) CNN, (b) Transformer [3]

The two characteristics are crucial for the task because most of the challenging test images are taken in indoor environment and have both low textures and very limited field of view (see Original Image in Table I). We observed that SIFT, one of the feature matching method, has a difficulty to obtain sufficient number of the corresponding points from those images. It is because not only have the images low-texture, but they also are close-shot images. In particular, images of a wall. Therefore, we use LoFTR to reference the global information and successfully find the matching images in the train dataset (see Matched Training Image in Table I).

C. Affine Transformation & Linear Regression

We started the first method with a naive approach. We assume any two images can be transformed each other by affine transformation matrix if the two images are taken at the close location and have 3 corresponding points on the

image planes. The affine transformation is composed of scale, translation, rotation and shear. This geometric transformation matrix consists of 6 parameters. Unlike projective transformation, it requires only 3 points. As we pointed, some given images cannot find 4 matching points that is required for projective matrices due to lack of the texture. We believe this naive approach can resolve the problem and generate rough coordinates with more images although the results may not accurate as projective transformation. Another assumption is that the difference of the positions between the original and the transformed coordinates has to do with the elements of the affine matrix. We used the elements as features to train a linear regression model that outputs the difference of the coordinates so that we predict the adjusted locations. Steps are as follows:

- 1) Shortlist top 10 similar images from ViT-similar images
- 2) Count the number of good points (over 0.7 confidence) from LoFTR, select one image with the most counts
- 3) Randomly select three points from the confident points and generate affine matrix by `cv2.getAffineTransform()`. There is no output when it fails to find the matrix elements
- 4) Train a linear regression (default setting) with affine elements (features) and difference between the true location and the selected image location

D. Camera Transformation - Essential Matrix

Since the scene of our images is often not a plane, we use an essential matrix to represent the transformation between images instead of a homography matrix which is a mapping between two planes. The camera geometry can lead to the further improvement in the accuracy. In this method, we make a hypothesis that if we can find the essential matrix of an image pair, we can transform from one image to the other. It means that the two images are close to each other in geolocation. The used `cv2.findEssentialMat()` itself has RANSAC that iteratively finds the best essential matrix.

TABLE I
CHALLENGING TEST IMAGES TO MATCH POINTS

Test Image		Matched Training Image	
Original Image	Matched Points ¹	Original Image	Matched Points ¹
IMG5344_5		IMG3580_5	
IMG4317_2		IMG4222_4	

¹ Grayscale image with matched points found by LoFTR

Following the steps below, we obtain the best match image with the most inlier points from train images. It means that the image has the highest probability to transfer from the test image and indicates that these two scenes are taken at near to each other. Therefore, we can consider the true location of the match image as the position of the test image.

- 1) Shortlist top 10 similar images from ViT-similar images
- 2) Find the essential matrices between every 10 pairs by `cv2.findEssentialMat()` with LoFTR detected matching points (input)
- 3) Select one image with the most inlier points from its corresponding essential matrix

Notice that, it sometime fails to find the essential matrix from a test image and the top 10 ViT-similar train images unless both of the images have sufficient texture and more than 5 corresponding points detected by LoFTR. Under these circumstances, we gradually release the searching range to top 20, and then to top 40 ViT-similar train images. All test images can find at least one train image that can compute the essential matrix within the top 40 ViT-similar train images.

E. Camera Transformation - Camera Pose

In this approach, we conduct decomposition on the essential matrix for the better geometric constraint by Singular Value Decomposition (SVD) and cheirality check. The cheirality check is to make sure if the decomposed matrices are correct by checking that all the triangulated 3D points have positive depth. As a result of the decomposition, we have rotation and translation transformations. We only count the points that can be described by the two transformations, and then consider the location of the match image having the most points as the finding location. The steps are as follows.

- 1) Shortlist top 10 similar images from ViT-similar images
- 2) Find the essential matrices between every 10 pairs by `cv2.findEssentialMat()` with LoFTR detected matching points (input)
- 3) Decompose the essential matrices into the rotation and translation matrices by `cv2.recoverPose()`

- 4) Select one image with the most points from LoFTR that can be described by the two transformations (rotation and translation)

V. EXPERIMENTATION

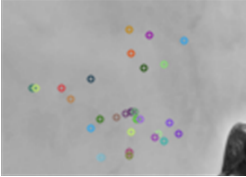
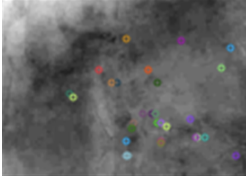
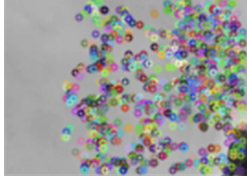
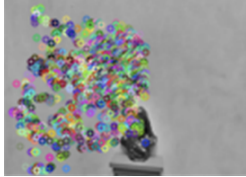
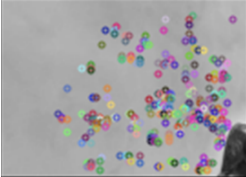
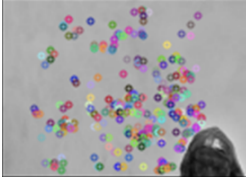
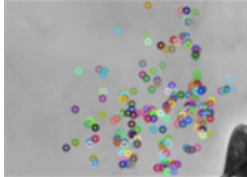

There are 5 different approaches have been attempted as follows.

- 1) *Top 1 ViT-similar* - directly uses the coordinates of the most similar image in train set (top 1 ViT-similar) as the prediction.
- 2) *Top 10 ViT-similar & most LoFTR matching point* - extracts the top 10 ViT-similar images in the train set, and then finds the matching points by LoFTR between the test image and each of the 10 images. Prediction is the same x,y coordinates of the matched train image having the most matching points out of the 10 images - a term 'LoFTR-most train image' is used to represent the image the following section.
- 3) *Affine Matrix& Linear regression* - goes through the same procedure as *Top 10 ViT-similar & most LoFTR matching point* to obtain the LoFTR-most train image. A linear regression model is trained with the affine elements from the train image and the difference in x,y coordinates. The coordinates of the real location of the LoFTR-most train image are adjusted by the predicted differences from the linear regression model. The adjusted position is the prediction.
- 4) *Camera Transformation - Essential Matrix* - extracts the top 10 ViT-similar images in the train set, and then finds the matching points by LoFTR between the test image and each of the 10 image. Based on the detected matching points, the essential matrix is computed per image. After filtering the valid inlier points by its matrix, one image which has the most number of valid points are selected as the best match. Prediction is the same x,y coordinates of the matched train image.
- 5) *Camera Transformation - Camera Pose* - goes through the same procedure as *Camera Transformation - Essential Matrix* to obtain the essential matrices. To find the



Fig. 2. Similar image pairs found by ViT

TABLE II
MATCH RESULT COMPARISON BETWEEN THE DIFFERENT APPROACHES FOR TEST IMAGE IMG4301_3 (LEFT)

Top 1 ViT-similar		Top 10 ViT-similar & most LoFTR matching point ¹	
			
Best Match with IMG3941_1		Best Match with IMG4178_2 (822 points)	
Best match found by essential matrix		Best match found by camera pose	
			
Best Match with IMG3281_4 (255 points)		Best Match with IMG3134_5	

¹ same result with Affine Matrix & Linear Regression which uses only three points

rotation and translation matrices from the essential matrix, we conduct Singular Value Decomposition (SVD) and cheirality check (i.e. make sure triangulated 3D points have positive depth). From the valid points (inliers that can be described by the rotation and translation matrices), we predict the position likewise the *Camera Transformation - Essential Matrix*.

VI. RESULT EVALUATION

To analyse the performance between the different experiments, we firstly sampled some matching results from the *Top 1 ViT-similar*. Although the approach performed the worst amongst the others, 7.01 with unseen images (see in Table III), it is clearly shown that the images were reasonably matched with the another similar image in Fig 2 regardless of the environments (indoor or outdoor) as long as there are enough texture. However, the difference in the performance was notable with low texture close-shot images which are the challenges throughout the project (see in Table II). Similar to the quantitative evaluation in Table III, the matching by the highest ViT-similar was the worst match in qualitative evaluation in Table II. It tells that the only matching by the image features is not satisfactory when the image is low-texture.

Table II illustrates that the best match image is much reasonable by adding *most LoFTR feature points*, yet the result is not the most geolocation-close image compared to the others. As we expected, the naive approach with affine matrix had similar performance as the *most LoFTR feature points*. Most interestingly, the most accurate approach we attempted was *Essential Matrix*. Although it had fewer feature points (255) detected in comparison to *most LoFTR feature points* (822) in Table II, it has the more valid inlier points. It means that it was easy to transform from the test image to the train image, therefore, those two images must be very close to each other. Lastly, we found that there was not a significant improvement,

but a negligible decrease on the accuracy by recovering camera pose. We consider that the imprecise decomposition of essential matrix is the reason why the *Camera Pose* performed a bit worse than *Essential Matrix*. We believe that the essential matrix already represented the transformation between images. In other words, the additional decomposition was unnecessary because *Essential Matrix* approach can reference the camera transformations, rotation and translation, to find the closest train image.

TABLE III
EVALUATION RESULTS

MAE	Approaches ¹				
	1	2	3	4	5
Train(Val)	-	-	11.63	9.13	9.55
Test ²	7.01	6.14	5.86	4.35	4.37

¹The approaches are numbered as in section V.EXPERIMENTATION

²Result on Kaggle

Overall, the different methods fairly perform with test images. The MAE is much lower than the performance of 'test_simple_1.csv' (MAE 9.40). However, if we provide colour information about the images by inputting R, G, B channels into the LoFTR, the accuracy of the prediction still can improve. In particular, the poorly matched images that only have similar texture, but different colour, can have more precise results.

VII. CONCLUSION

In conclusion, we have developed several approaches for image-based location recognition and shown the methods can acceptably predict the fine-grained location. Despite their plausible performance, we believe there still exists a room to improve. As proposed, we can expect the better output with colour maps.

REFERENCES

- [1] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2020.
- [2] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do Vision Transformers See Like Convolutional Neural Networks?," arXiv preprint arXiv:2108.08810, 2021.
- [3] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "LoFTR: Detector-free local feature matching with transformers," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8922-8931.
- [4] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 4938-4947.