Image Matching Module for Screen Captured Image using Deep Learning

Hye-won Chun[1], Seong-Soo Han[2], Chang-Sung Jeong[1],

[1]Department of Electrical Engineering, Korea University, Seoul, Republic of Korea
[2] Department of Division of Liberal Studies, Kangwon National University, Samcheok, Republic of Korea
{uclacarol}@korea.ac.kr, {sshan1}@kangwon.ac.kr, {csjeong}@korea.ac.kr

Abstract. In this paper, the difference in image size in the smart device environment deals with the problem of finding the region at the matching point, even if the space of the captured area, the color difference, etc. are transformed. We provide an AI-based Image matching Module so that such variants can also find the appropriate region. Use D2-Net to find matching points between Screen image and query image and use MAGSAC, RGB value comparison, and distance comparison between points for Region of Interest(RoI). Even if various deformations occur in the matching area via this image matching module, the correct position of the query image can be found in the screen image and high performance can be obtained.

Keywords: Feature detection and extraction; Feature matching; Deep Learning; Screen captured image;

1 Introduction

As the rate of technological development increased day by day, the development and release of new products became very short, and the types and numbers of products that had to be tested on the products increased significantly. As a result, there is a need for technology that can support various products without changing or additionally implementing Test Platform. Conventional methods, there are two problems. The first is slow speed and low accuracy. Second, when applying the test script of the existing tester to various smartphone environments, it is not possible to use the test script for transformation for each image (screen image, query image).

Use D2-Net with Deep Convolutional Neural Network-based Feature Extraction & Detection for feature matching of screen image and query image where deformation exists. In order to find a suitable matching point for the Screen captured image, refine the Megadepth dataset of the existing D2-Net Training dataset, create a dataset in the Screen captured image, and train. The image matching Module for screen captured image is to find the region that matches the query image in the screen image with the feature point obtained by D2-Net. Execute Mutual Nearest Neighbors Matching and Fitting Algorithm, Comparison of RGB values of Matching Points, and Comparison of distance between matching points in order at feature points.

Matching points and matching regions can be found in the transformed screen and query image via this module, and results can be obtained with better performance than the existing D2-Net.

2 Related Work

2.1 Image Matching

Image matching is a technique for comparing two images to find out if an object with the same structure exists. There are various methods for that.

SIFT (Scale-Invariant Feature Transform) [1] solves the problem that is sensitive to scale changes in existing feature point extraction based on DoG (Difference of Gaussian). Extracting scale-invariant feature points produces an image pyramid that is a scaled-down image by gradually reducing the image. At this time, the edge is detected in each scale, but the same point is detected in the edge scale. With this structure, a scale invariant feature point can be obtained.

Template Matching is a method to find the area that best matches the query image on the screen image. As the query image moves on the screen image, it calculates the pixel value and searches for the best matching area.

There is more development in the deep-learning generation. Studies prior to D2-Net [2] distinguish between feature detectors and feature descriptors and are called the detect-then-describe approach. However, this approach has its limitations. Local descriptors potentially encode higher levels of structure, taking into account larger patches, and key-point detectors look only for small image areas. Therefore, this method significantly reduces performance due to extreme appearance changes.

2.2 COLMAP

COLMAP [3] is a general-purpose SfM (Structure-from-Motion) and MVS (Multi-View Stereo) pipeline with graphics and command-line interfaces. After reversely tracking the position and direction of the captured camera using the motion information of the image captured in two dimensions, the relationship between the image and the camera is structured. MegaDepth [4], a large dataset, is generated via SfM and MVS.

3 Image Matching Module

The Image Matching Module uses D2-Net to perform Feature Extraction and progress the Region of Interest (RoI) for the query area.

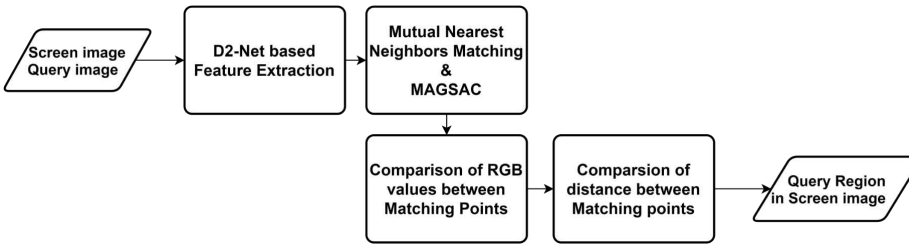

Fig. 1. Receive screen captured image and query image at input, and output is the Query image area on Screen Captured Image.

3.1 D2-Net based Feature Extraction

Studies prior to D2-Net distinguish between feature detectors and feature descriptors and proceed in the detect-then-describe approach. The key-point detector of this approach has a search limit only in a small image area. Therefore, this method is very vulnerable to changes in external conditions, looking only for small image areas.

D2-Net proposes to generate feature detectors and feature descriptors at the same time. That is, it generates feature detectors and feature descriptors at the same time with low-level information without performing feature detection initially. Using this method, we show that even difficult conditions such as day and night changes and indoor scenes are beyond the cutting edge. D2-Net has high accuracy of key-point detection results and shows high performance even in disease types such as day and night changes and indoor scenes, so it is suitable for feature extraction to multiple variations of smartphone screens.

Using the D2-Net feature extraction method, the screen image and query image are received as input, and the feature key-point and descriptors are set as output values.

3.2 Mutual Nearest Neighbors Matching and Fitting Algorithm

In the existing D2-Net, we get the points to be matched via mutual nearest neighbors matching with key-points and descriptors which are the outputs of feature extraction. Mutual nearest neighbors matching is Brute-force matching of descriptors. Matching points output an appropriate model using the RANSAC (RANdom SAmple Consensus) algorithm. However, in the case of RANSAC, when the model is complicated and the number of accurate points is small, RANSAC takes time to obtain a satisfactory result, and many matching points cannot be obtained. To make up for this shortcoming, the MAGSAC (marginalizing sample consensus) series is used for outlier removal instead of RANSAC. In the case of MAGSAC [5] and MAGSAC ++ [6], more points can be obtained with less repetition than RANSAC. Mutual nearest neighbors matching and fitting algorithm receives key-points and descriptors, which are the outputs of feature extraction, as inputs, and outputs become MAGSAC inliers.

3.3 Comparison of RGB values of Matching Points

By giving a threshold in the range of about 60% of the comparative color value, it is possible to obtain the same performance even when the external conditions change. There are three reasons to compare RGB value. The first is to improve the accuracy of the matching points that exist in the screen image and query image. The second is to solve the problem that the query image and color existing in the screen image give the wrong result as the query image region for the other icon region. The third is that the same icon as the query image that exists in the screen image exists, and one or more query image region results are obtained. If the color has the wrong query image region in the screen image for other icons, this threshold is also used to resolve it.

Finally, if the screen image has multiple icons that are the same as the query image, the image matching algorithm is repeatedly executed by covering only the icon part obtained by executing the first image matching algorithm with a mask. In this way, we can get multiple regions that are identified by the same query image.

The input of comparison of RGB values of matching point is inliers, and the output is a list of inliers whose inlier RGB values of screen image and query image are constant threshold or more.

3.4 Comparison of distance between matching points

Finally, we need to find the exact region in the screen image that corresponds to the query image. The idea for this is to find the area where the points are densely located in the area where the query image is likely to exist, so the points in the screen image are united. Measure the distance between points using the Euclidean Distance to find a dense area. Find the distance from point to another point, compare the total distance per point with the total distance of other points, and define the point with the smallest total as the point in the dense area.

The input of Comparison of distance between matching points is a list of inliers, and the output is a point of a dense area.

4 Experimental Evaluation

Dataset. Purify the Megadepth dataset for training, change the Screen captured image data to Megadepth format, and train. Screen captured images of all smart devices can be used for datasets.

In order to have Screen captured image data and train it, it must match the Megadepth data structure. It is necessary to define the scene with Screen captured image based on the definition of COLMAP. First, configure the scene on a screen with the same resolution. Second, the screens that make up the scene must be screens that perform the same action.

As a result, create a base Screen captured image with Screen captured image and proceed with augmentation. Then, considering the execution time of colmap, a total of 325 scenes and 40 or more image pairs per scene generate a total of 13000 datasets.

Evaluation. Evaluate two datasets. The screen image of the first dataset is a capture of the screen of a smartphone, and the screen image has three scales (1440x2880, 1440x2560, 1080x1920). The screen image of the second dataset is the navigation screen, and after shooting with the camera, the transform was advanced and the scale is 1024x768.

The first dataset evaluates whether various scales and the second dataset can perform well against changes in illuminance, angle, and image quality.

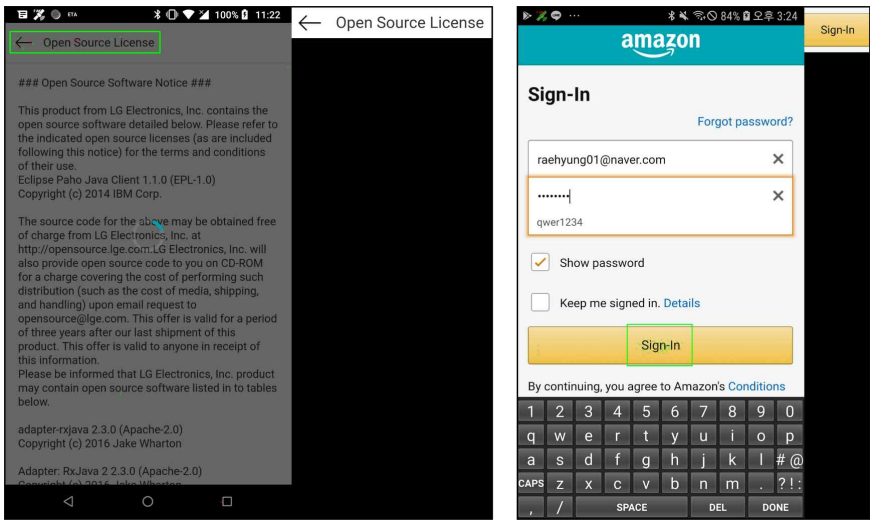

Fig. 2. Results of Screen Captured Image using Matching Module. If the screen image has the same query image or the screen image and query image are deformed, the query area is returned.

Table 1 and Table 2 have the acquired accuracy and speed obtained from the two datasets. Base D2-Net uses the existing D2-Net. Image Matching_v1 used MAGSAC ++ and Image Matching_v2 used MAGSAC. We can use the Image Matching Module to see more than 95% accuracy and less than 1 second of speed. There is a 60-90% performance improvement when compared to the existing Template matching results, and at least a 3-20% performance improvement when compared to the base D2-Net results.

Table . Evaluation on the capture of a smartphone screen dataset.

Mothed	Accuracy(%)	Speed(sec.)
Template Matching	28.95%	9.95
Base D2-Net	89.30%	1.23
Image Matching_v1	92.29%	0.75
Image Matching_v2	96.47%	0.99

Table 2. Evaluation on the navigation screen dataset.

Mothed	Accuracy(%)	Speed(sec.)
Template Matching	0.01%	2.243
Base D2-Net	73.59%	0.468
Image Matching_v1	94.16%	0.465
Image Matching_v2	95.25%	0.554

5 Conclusion

We proposed an image matching module for Screen Captured images using feature matching based on Deep Convolutional Neural Network and Region of Interest (RoI).

Feature extraction was done using CNN-based D2-Net. In order to find the correct query region in the screen captured image in the result feature, we proceeded with a total of three processes: removing outliers after Mutual nearest neighbors matching, comparing matching point RGB values, and comparing the distances between points.

The image matching module for Screen captured image has two contributions. First when testing an application, regions can be found in images of various smart devices, even if there are various image size, query image gaps, icon color, and size variations. Second, we were able to achieve the performance of finding a region even if changes such as illuminance, angle, and image quality differences occur in the Screen image and query image.

References

1. Lindeberg, Tony.: Scale invariant feature transform (2012)
2. Dusmanu, Mihai and Rocco, Ignacio and Pajdla, Tomas and Pollefeys, Marc and Sivic, Josef and Torii, Akihiko and Sattler, Torsten.: D2-net: A trainable cnn for joint detection and description of local features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8092--8101 (2019)
3. Schonberger, Johannes L and Frahm, Jan-Michael.: Structure-from-motion revisited. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4104--4113. IEEE (2016)
4. Li, Zhengqi and Snavely, Noah.: MegaDepth: Learning Single-View Depth Prediction from Internet Photos. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2014--2050 (2018)
5. Barath, Daniel and Matas, Jiri and Noskova, Jana.: Magsac: marginalizing sample consensus. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 10197--10205 (2019)
6. Barath, Daniel and Noskova, Jana and Ivashechkin, Maksym and Matas, Jiri.: MAGSAC++, a fast, reliable and accurate robust estimator. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1304--1312 (2020)