COMP90086-project report

Jiawei Luo
University of Melbourne
student id:1114028

Yifan Deng
University of Melbourne
student id:1150027

***Abstract*—This paper presents two methods for finding the location of a photograph for a Kaggle competition. The Kaggle competition provided 7,500 images and information on the positioning of their pictures and 1,200 test photographs. Predictions of the positioning of the test photos need from the training information. One method is called perceptual hashing and is a standard method of matching images. Another technique uses a pre-trained model to extract image features and calculate similarity. The second method obtained better results in this competition.**

I. Introduction

Image search is a technique that many search engine companies highly value. It aims to search for similar images in the search engine's database by using an image provided by the user. In the Geo-location problem, to locate the geographic location of a particular image, it is necessary to find a specific reference in the image and compare it with the reference on the image in its own database to locate the possible geographic location of the searched image. The common approach is to compare the two images and retrieve a score that reflects the similarity of the two images.

The Kaggle competition is to find geolocation information from image information, and the problem can be understood as a search problem for image features. Firstly, in the training set provided, every five images share one coordinate, which means that every five images come from the same geographical location. An assumption can be made that the images in the test set can find images from the same true geographic location in the training set. If the true location of the test image does not exist in the training set, the image with the most similar features is searched for as a reference. The assumption is made that images with similar features are also close in geographic location.

In the field of image matching, SIFT [1] is a commonly used technique that has proven to be very effective in target recognition applications. Still, it also has the significant drawback that the computational complexity required for SIFT is too great.

The training database contained 7,500 images for this Geo-location problem, while 1,200 images were included in the test data. The huge complexity of SIFT is fragile for such image search problems that require many matches. An alternative solution is perceptual hashing(PHash) [2]. PHash is a hashing algorithm commonly used for image similarity detection. It works by generating a fingerprint string for each image and comparing the fingerprint information of different images to determine the similarity of the images. The closer the result is to the image, the more similar it is. Its advantage is that the algorithm's complexity is very low, and it can accurately match similar images and calculate the score.

On the other hand, convolutional neural networks(CNNs) are superior in feature extraction. If CNN-based methods can be used to extract features from images, the robustness of CNNs can be used to overcome errors in rotation, lightness, colour and viewpoint in image search.

In this paper, the methods used for similar image search are described in the part II and their results are compared in the part III.

II. Methods

A. PHash

A perceptual hash differs from a random hash in that it generates a hash ”fingerprint” with information about the original characteristics of the image. Random hashes are primarily used in encryption algorithms, where the hashes generated from different data are directly different and cannot be compared. The fingerprint of an image can be compared. Therefore a perceptual hash can be used as a criterion to evaluate whether two images are similar.

The steps typically used by PHash include [2]:

- Resize: scales the image down to a size of 8*8, a total of 64 pixels. This process removes the details of the image, retaining only the basic information such as structure, light and dark, and discarding the differences in the image due to different sizes or scales.
- Grayscale conversion: takes the reduced image and converts it to grayscale.
- Calculating the average: calculating the average of the greyscales of all 64 pixels.
- Comparing the greyscale of pixels: comparing the greyscale of each pixel, with the average value, with a value greater than or equal to the average value being recorded as 1 and less than the average value being recorded as 0.
- Calculating the hash value: the results of the comparison from the previous step are combined to form a 64-bit integer, which is the fingerprint of this image.

After obtaining the fingerprints, the similarity of the two images is calculated using the Hamming distance.

B. SIFT method

SIFT (scale-invariant feature transform) is a description used in the field of image processing. This description scales invariant can detect critical points in the image and is a local feature descriptor. In this article, we focus on finding the most similar images by matching them with SIFT.

The processing process is divided into the following main steps [3]:

- Find the feature points on each image, usually these feature points may be corners, edges.
- Precise positioning of key points
- Orientation determination of key points
- Generation of feature vectors
- Feature matching by KD tree.
- Use the RANSAC to filter the error points

But as mentioned before, the SIFT matching takes too long because after finding out the feature points, it will match them one by one. On top of that, with a total of 1200 test images and 7500 training images, it takes even longer. Therefore, later in this paper, we will use a combination of PHash and SIFT for the coordinate prediction.

C. ResNet-Based feature extraction

Besides using PHash, another method of detecting similar images is to use a pre-trained CNN-based model for feature extraction. A set of feature vectors is obtained at the last layer of the pre-trained model, and a score represents the two vectors' similarity.

As shown in *Fig.*1, the image is input into the same convolutional neural network, and a one-dimensional feature vector of length n is output at the last fully connected layer. The image can be ungrayed and the colour information retained while extracting features depends entirely on the pre-trained neural network. A well-trained neural network can be sufficiently robust to characterise different spatial, lightness, colour, and viewpoints of the same geographical location.

On the other hand, to compare two feature vectors, cosine similarity [4] is a common metric. Cosine similarity describes the space of inner products between two non-zero vectors, as shown in (1), the cosine of the angle between two vectors in a vector space is used to measure the similarity. As the two vectors become more similar, the closer the cosine is to 1, and the angle tends to 0. Conversely, the closer the cosine is to 0, the less similar the two vectors are.

$$similarity = cos(\theta) = \frac{\mathbf{A} * \mathbf{B}}{|A||B|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{1}$$

D. Coordinates selection Method

The image similarity calculation enables a ranking of the scores on the training images to be obtained. The coordinates are filtered to select the searched images better and output the coordinates (x, y). A parameter k is set to indicate the selection of the top k images in the ranking as candidates, i.e. the images closest to the test image.

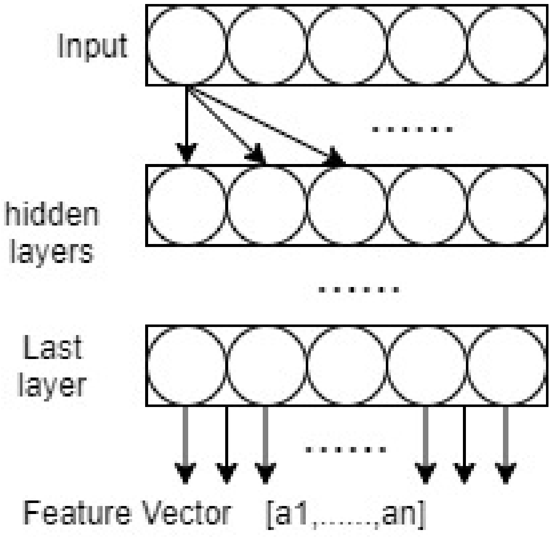

Fig. 1. Extraction of feature vectors

Two methods are defined here:

- **Vote method**: The first method is to directly use the (x, y) coordinates of one of the candidates. This method assumes that the images in the test set can be found similar in the training set and taken in the same location. Since every five images in the training set can be found sharing one coordinate (x, y), it can be assumed that all five images are from the same location. If the number of similar training images from the same coordinate is greater, the more likely the test image came from that coordinate.
- **Average method**: The second method is to take the average of the coordinates of the first k candidates. This method assumes that images from similar geographical locations may have similar image characteristics, architectural style, brightness, colour, etc. By averaging the coordinates of the first k candidates, the predicted value of the test image is used.

III. Result And evaluation

A. Result of CNN-based method

The prediction results of ResNet/VGG combined with cosine similarity are shown in Table I.

The evaluation metric used half of the data from test's 1200 to calculate the mean absolute error(2) between the predicted test coordinates (x, y) and the actual coordinates.

$$MAE = \frac{1}{N}(\sum_{i=1}^{n}(|x_i - \hat{x}_i| - |y_i - \hat{y}_i|)) \tag{2}$$

1) Parameter k: In the results, the value of the parameter k plays a vital role in the outcome. This method can be seen as a k-nearest neighbour algorithm with many labels but only 5 instances of each label. A suitable value of k needs to be

TABLE I
RESULT OF CNN + COSINE SIMILARITY

Method	**ResNet152+CosSim+Vote**			VGG16	VGG19	NasNet
top K	1	3	5	1	3	1
MAE	7.858	7.801	7.811	8.422	9.01	14.705

chosen to encompass a sufficient number of candidates so that there are more candidates from the target geographical location than from other geographical locations. When the value of k is equal to 1, the algorithm selects the coordinate with the highest score as the output, so that if the value of k is too small or too large, the weights of images from the target geographic location will be diluted.

It was previously thought that a large k value, although diluting the weight of the target localisation, could have a greater chance of making two or more images from the target localisation candidates while providing a better Vote result. Due to many localisations, there may be fewer images from the same non-targeted localisation, so choosing a larger k value may improve the localisation results. After experimentation, it was choosing a k value greater than 3 resulted in progressively lower results, which means that this assumption is not accurate. This may be because many of the training images from different localisations have similar features, such as some images from different museum interiors are very close to each other.

2) Selection of Convolutional Neural Networks: The choice of a convolutional neural network determines the lower limit of the image matching result. Intuitively, a strong convolutional neural network can provide good extraction of key features while using pre-trained weights. ResNet152 has more layers and better feature extraction than VGG16 so that a better matching result can be obtained.

However, we have tried different pre-trained networks and have found that a deeper network does not always have better feature extraction results. As the pre-training weights are derived from a training set of tens of thousands of categories, choosing a model with good robustness or focusing on buildings, indoor and outdoor environments would be more suitable for this task.

3) Coordinates selection Method: The experimental results show that using Vote outperforms using average as the output condition, ceteris paribus, due to the random nature of the geographical location distribution. This implies the assumption that images in neighbouring regions have similar features is weaker relative to the assumption of the Vote method.

4) Pros: Compared to SIFT, CNN is much less computationally complex because it uses pre-trained models that do not need to be retrained using existing data. In practice, training from random weights will not be efficient due to the limited data. After the training features have been extracted using a CNN, the training features can be retained and used for subsequent computations.

5) Cons: This algorithm is easy to accomplish, but it has little control over the results of feature extraction. The feature extraction result of this method depends on the pre-trained CNN effectiveness. Thus it is challenging to have excessive control over the parameters to improve the outcome.

B. Result of PHash method

At the very beginning, the low frequency part of the image is selected, which is the 8*8 array in the upper left corner of the 32*32 two-dimensional array. And simply selected the coordinates of the graph with the highest similarity to the images in the test set for output. However, the performance is very poor and the score in kaggle is only **54.190**. The reason for this situation can be divided into two parts. One is because pHash ignores most of the picture details while only taking the low frequency part. The second part is due to the fact that we only considered the most similar images, and since pHash is not very accurate, the most similar images are often not the most reasonable.

In order to optimize from these two aspects, this paper can improve the level of detail of the input image by selecting the middle part of a 32*32 two-dimensional array for calculation. Secondly, this paper selects the images from training set with similarity greater than 75%, and then by counting the groups of images that appear more than 3 times. Finally, in these groups of images, the number of features is calculated by SIFT and summed up to get the total number of features for each group of images. If the number of eligible images is greater than 1, the average of the coordinates of the top two images with the highest feature count will be calculated as the final result. The results obtained by the optimized algorithm improved by 19.70% and scored **43.527**, which is still not ideally, and it even took almost eleven hours to output the results.

1) Pros: The advantage of pure PHash is its extremely fast computation, as it scales down the image to a minimal size and performs hashing and it takes just over a minute to process the entire test set. It retains the spatial information of the image using only a string of coded numbers.

2) Cons: PHash only takes the low-frequency part of the image in order to tolerate the deformation of the image, which results in not extracting the detailed part of the image during feature extraction, making it not ideal to do similar image finding for solid color or near solid color images. For example, in our test set there are quite a few images that resemble blank walls.

IV. CONCLUSION AND FUTURE DIRECTION

In this article, two methods are described for identifying the geographical location of a photograph taken. The CNN-based approach achieves better results due to its higher robustness in the extraction of image information. Even though PHash has not achieved good results in this problem, it still has an irreplaceable advantage in matching similar images. It is more often applied to matching images with the same structure, for which it is difficult to find nearly identical images due to the rotations involved and the change in viewpoint.

One possible way to improve performance is to calculate the Essential matrix and the fundamental matrix to calibrate the view of the image and determine the Epipolar geometry of common feature points in the image. This localisation problem is still essentially a feature matching problem. Using the Essential matrix to find the feature points at the appropriate locations may be more accurate and combine this with feature recombination test data from multiple target localisation images.

However, the method of computing the Essential matrix requires additional computational complexity. It requires calculating the potential matching points for each image by affine-invariant SIFT, which is not acceptable in terms of the time necessary to perform 1200*7500 matches. It is worth looking forward to finding out the geographical location of a photo more precisely if the calculation time can be reduced based on this method.

On the other hand, a popular deep neural network approach may have additional effects in matching feature similarity. Siamese neural networks are able to measure the similarity of two inputs [5]. By feeding two inputs into two neural networks, these two neural networks each map the inputs to a new space, forming a representation of the inputs in the new space. The similarity of the two inputs can be well evaluated by the calculation of Loss. Nowadays, Siamese neural networks are also widely used for image, text similarity matching, and it can be expected that it will work well for this problem.

References

[1] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[2] “pHash.org: Home of pHash, the open source perceptual hash library.” http://www.phash.org/ (accessed Oct. 15, 2021).

[3] D. G. Lowe, ”Object recognition from local scale-invariant features,” Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999, pp. 1150-1157 vol.2, doi: 10.1109/ICCV.1999.790410.

[4] “Cosine similarity,” Wikipedia. Oct. 11, 2021. Accessed: Oct. 15, 2021. [Online].

[5] D. Chicco, “Siamese Neural Networks: An Overview,” in Artificial Neural Networks, H. Cartwright, Ed. New York, NY: Springer US, 2021, pp. 73–94. doi: 10.1007/978-1-0716-0826-5_3.