Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Exploring the Impact of Similarity Model to Identify the Most Similar Image from a Large Image Database

To cite this article: Ye Chen 2020 *J. Phys.: Conf. Ser.* **1693** 012139

View the article online for updates and enhancements.

Exploring the Impact of Similarity Model to Identify the Most Similar Image from a Large Image Database

Ye Chen

Australian National University

Email: u6563670@anu.edu.au

Abstract—Identifying the most similar image from a given large image database is becoming more and more popular, and many companies have brought out related products. However, there are some incomplete functions and shortcoming of previous work, the database was disorganized and does not have any regulation, and the similarity method used in the image searching system can be improved. In this study, we have implemented an improved similar image searching system that can filter the dataset based on the requirement of users. Besides, an experiment about testing different similarity models has been done. The result is an improved similarity model which can be subsequently applied to the searching system.

1.**Introduction**

The study is focus on utilizing a neural network for multiple purposes. In particular, we are focusing on finding the most similar image in the dataset. It is an application that uses machine learning, pattern recognition, computer vision to analyze, detect, and identify the most similar image from a large image database.

Imagine a user on the internet has an image. Looking for similarities on the internet will return information that are better targeted. In fact, there are more search engines that has integrated such a system[9] such as Google Image[13], Yandex Image[14], and other sales platforms like Amazon.

We want to design a flexible system that is able generate a cleaner dataset based on different searching cases. The system may be updated to include what the user mainly wants to find to minimize the searching scope and increase the searching accuracy and speed. Besides, cosine similarity [4] is the main ideal of the whole project and this application becomes widely used. Google Image is likely to use this method to power its reverse image search functionality. To solve the problem which was described in the abstract, the study consists of three contributions:

First, we redefined the dataset and set the coefficients of each category.

Second, we evaluated different similarity models and presented the test results.

Third, we provide a reasonable explanation for each similarity model and decide to choose which one to be used in an image searching system.

The rest of this paper is organized as follows. In section II, we build a new dataset and show how it is generated. In section III, the theory about how to find a similar images via a neural network is displayed. Section IV is showing an experiment on testing different similarity models and the explanation of the experiment results is written in section V. Finally, the related work is shown in section VI and the final section is the conclusion of the whole project.

2.Dataset Benchmark

Like the last task, this task focus on searching a similar image of the input image. Based on this, adding the searching coefficient to make the system more flexible.

In this project, we will be using Flickr8K [11] dataset. We introduce a new benchmark collection for sentence-based image description and search, consisting of 8,000 images that are each paired with five different captions that provide clear descriptions of the salient entities and events. The images were chosen from six different Flickr groups, and tend not to contain any well-known people or locations but were manually selected to depict a variety of scenes and situations.

To make the entire system return good results in different situations and reduce the inaccuracy caused by the two types of images that are too similar, the entire data set is classified. Among them, there are four major categories: people, animals, aircraft, and landscapes. This is simulating the real-world image retrieval. It is more flexible so the users can design their dataset for training and testing. In more detail, when the user enters the input image, He can tell the system what category the picture belongs to. In this case, the accurate result will be generated by less influence from irrelevant but similar images in the dataset.

In general, there might be 4 main situations when searching for similar images, and it becomes the reason why he dataset is classified into 4 categories. Looking at figure1, it describes the composition of the dataset Firstly, people. In daily life, almost everyone struggles to identify someone in an image. As a result, the first big category is People. Similarly, Animals are another big category. Furthermore, As one of the most common things in life, transportation can also be one of our categories. It not only has many types but also has its own characteristics, making it a good training model. This becomes the reason why the third category is goods. Finally, the landscape. Thinking about this situation: you see a beautiful place but don't know where it is, the final category is designed based on this situation. This is an assumption that there will be a searching system that can help find where the user looking for.

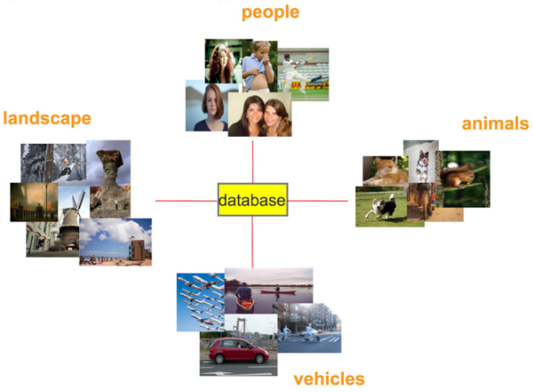

Fig.1 Database categories

3.Identifying a similar image

As we already know a 300-dimensional feature extracted vector can be obtained from the Inception-V3 model, using an image representation generator to take any input image and generate its encoding (300-dimensional feature extracted vector will go as an input in the first step of the caption decoder).

There are two steps processing the function: One is applying the Inception-V3 [1] model to the whole dataset and store their feature extracted vector. Another, is when a user wants to search for a similar image, calculate the cosine similarity between the decoding result of the target image and every item in step 1. Then return the images in the sequence of similarity value.

3.1.Image feature extraction

To extract visual features, GoogleNet [2] or Inception-V3 [1] are applied as base models. Keras provides a set of deep learning models along with pre-trained weights on ImageNet[16]. ImageNet[16] is an image dataset contains more than 14 million images with more than 21 thousand groups or classes. These pre-trained models[3] can be used for image classification, feature extraction, and transfer learning.

Feature extraction[12] and representation is a crucial step for image searching. How to extract ideal features that can reflect the intrinsic content of the images is still a challenging problem in computer vision.

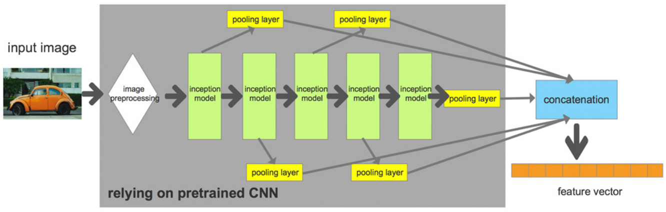

Fig. 2 Structure of Inception-V3

Figure2 introduces the framework of the proposed method of the Inception model[1]. A given input image to be evaluated is run through a pre-trained CNN body(Inception-V3) [10]. Specifically, the global average pooling (GAP) layers are attached to the output of each Inception module. GAP layers are applied in CNNs to reduce the spatial dimensions of convolutional layers. By adding GAP layers to each Inception module, the global average pooling is attached to each Inception module to extract resolution independent deep features at different abstraction levels.

3.2.Test method

In this project, using k-fold cross-validation, To be specific, there are 6000 images in the dataset, randomly separate the dataset into 5 folds, trains the network by using the first 4 folds, and test the model. Besides, the definition of similarity of images is decided by a human, so the evaluation will be done by human eyes.

4. Similarity models

4.1.Similarity

Cosine similarity[4] measures the similarity between two vectors by measuring the cosine of the angle between them. Formula (1) shows that the cosine similarity only measures the included angle, the length of the vectors becomes unconsidered. If two input vectors have the same direction, no matter how long the distance between them is, they are considered similar.

$$sim(X,Y) = \cos(\theta) = \frac{X \cdot Y}{|X||Y|} \tag{1}$$

As cosine similarity is the best model that has been chosen, some typical test results among 4 categories are shown in figure 3, the left side in figure 3 is the input image and the right side are output images.

4.2.Euclidean distance

Euclidean distance[5] refers to the true distance between two points in the m-dimensional space, or the natural length of the vector (that is, the distance from the point to the origin). The Euclidean distance in two-dimensional and three-dimensional space is the actual distance between two points.

$$\rho = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2}$$

4.3.Correlation distance

Correlation distance[6] is designed on the covariance "cov" and standard derivation "σ" between two give vectors.

$$\rho = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \tag{3}$$

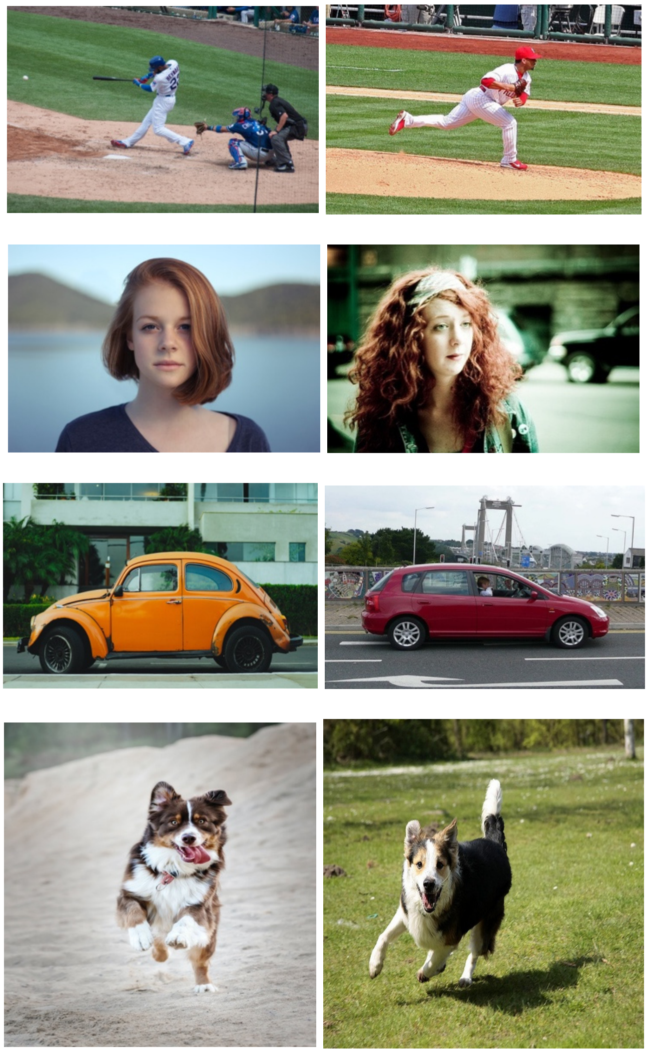

Fig. 3. Typical testing result of 4 categories

4.4.Manhattan distance

Manhattan Distance[7] between two vectors is equal to the one-norm of the distance between the vectors. It calculates the sum of absolute differences.

$$\rho = | x_1 - x_2 | + | y_1 - y_2 | \tag{4}$$

4.5.Chebyshev distance

Chebyshev can be also called Chebyshev distance. It is defined on a vector space where the distance between two vectors is the greatest of their differences along any coordinate dimension.

$$\rho = \max(|x_{1} - y_1|, |x_{2} - y_2|, \dots |x_{n} - y_n|) \tag{5}$$

4.6.Pearson correlation coefficient

Based on cosine similarity, each vector minus the vector composed of the vector mean, which is the Pearson correlation[8] coefficient

$$\rho = \max(|x_{1} - y_1|, |x_{2} - y_2|, \dots |x_{n} - y_n|) \tag{6}$$

Figure 4 shows the accuracy of finding similar image in each data category among different similarity models. As shown in the above stacked bar chart, Cosine similarity is the best model that can predicate 94 percentage similar image. Pearson correlation similarity model has almost the same performance as cosine similarity but is still a little inferior. Correlation distance can only predict 83 percentage similar images. The remaining distance models do not have satisfactory performance and the reason will be explained in the next section.

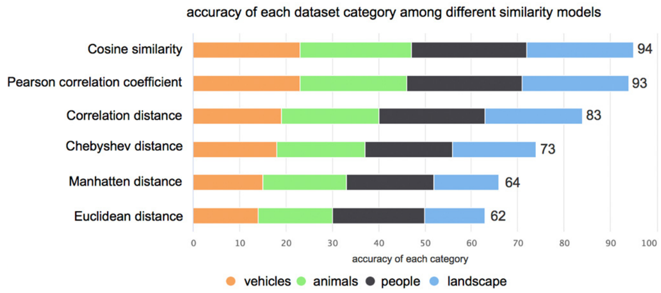

Fig 4. Testing result of different models

5.Explanation of similarity models

After the experiment, a crucial discovery is that images similarity can mainly dependent on the direction of features vectors. The similarity models which only consider the direction of feature vectors can perform much better than the other models. The following explanation can better prove this hypothesis.

What the network does is extracting the image feature. Then use the similarity model to compare the target image features with all embedded images features. We consider the image features as the vectors. The assumption of Cosine similarity is: because we are looking for a similar image, not the same image, the features vectors which are in the same direction can be recognized similarly. For example, if the target image is a man, the features vectors can represent the face, hair, body. We are looking for an image that has those features. It does not matter that the face's size, hair's length, and body are different That means the vectors in the same direction can be recognized as a similar image. The specific distance is not really important. From figure 5, it can be clearly discovered how cosine similarity works. For clear observation, the feature vectors are drawn in 2D. That is the greater the angle, the lower the similarity.

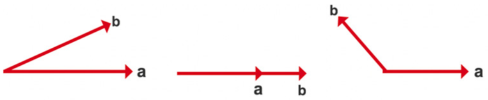

Fig. 5 Cosine similarity ideal

In the similar models declared in the above section, there are many similarity models that are calculating the real distance between the input vectors which are image features in this image searching system. For example, figure 6 shows the Euclidean distance between (x_1, y_1) and (x_2, y_2). In this case, vector direction is not as relevant, what it calculates is the distance between the endpoints. At the same time, the other distance similarity model has a similar ideal of calculating the similarity of vectors. These models are all based on the assumption that: the features vectors of similar images will have both similar direction and length. From the experiment, we know that this assumption is not the best because the direction of the feature vector is far more important than the length of the vectors. As a result, it verifies the experiment results that the cosine similarity model performs best in finding similar images.

Remarkably, the performances of Pearson similarity and Cosine similarity [8] are highly coincident-only a few results are different. Pearson correlation and cosine similarity are invariant to scaling, i.e. multiplying all elements by a nonzero constant. Pearson correlation[8] is also invariant to adding any constant to all elements. For example, for two vectors x_1 and x_2, and the Pearson correlation function is called Pearson(), Pearson(x_1, x_2) =Pearson(x_1, 2 x_2 + 3). This is an important property because you often don't care that two vectors are similar in absolute terms, only that they vary in the same way. Finally, we choose the Cosine similarity model to be used in the similar image searching system.

6.Related work

This section reviews some of the most relevant techniques and ideals. This section will be introduced around three parts: the dataset, test method, and similarity models.

In this project, the Flickr8K [11] dataset has been chosen. We introduce a new benchmark collection for sentence-based image description and search, consisting of 8,000 images that are each paired with five different captions. Because the images are random, the database is messy and does not have any regulation. A more flexible system that the more suitable dataset can be generated from based on different searching cases. The system may know what the user mainly wants to find to minimize the searching scope and increase the searching accuracy and speed. To make the entire system give good results in different situations, the entire data set is classified. 4 big categories has been chosen which are believed to be the most valuable and estimable categories. It helps the system becomes more flexible and human friendly.

In the past project, there is a test folder contains 5 images, Paras Chopra[15] used only 5 images to test the model which is not accurate and academic. In order to make the experiment more accurate, more testing should be done. To avoid find the image itself when looking for similar images in the searching system, k-fold Cross Validation has been introduced. It not only can help test the similarity model but also help improve the network. Moreover, evaluation totally depends on the human eye. But different from Paras Chopra's method[15], there are systematic rules to evaluate the system. As a result, the whole system becomes more reasonable and accurate.

In the past, cosine similarity is chosen directly without any experiment. After testing a lot of similarity models, it can be found that cosine similarity is indeed the best model, but there are many other similarity models performance good too. For example, Pearson similarity is as good as cosine similarity[8]. Different inputs may need different similarity model. All similarity methods should be considered when creating a similar image searching system.

7.conclusion

On the base of Paras Chopra's project[15], an experiment on the testing similarity model is completed. There is a crucial discovery that images similarity can mainly depend on the direction of features vectors of images. So the cosine similarity model performs better than any other distance similar model. A significant change made in the previous database was classifying the dataset into 4 most common categories. This change improved the similar image searching system that users can decide the categories of the tanning dataset. That means the system can give a more accurate results based on the different searching situations. From the experiment result, using the best similarity model we choose, the accuracy has been increased significantly and performs well in the existing categories.

References

[1] Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826

[2] Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; pp. 1–9.

[3] Domonkos Varga. Multi-Pooled Inception Features for No-Reference Image Quality Assessment; Appl. Sci. 2020, 10(6), 2186; 23 March 2020.

[4] Xia, P., Zhang, L. & Li, F. 2015, "Learning similarity with cosine similarity ensemble", Information sciences, vol. 307, pp. 39-52.

[5] Fabbri, R., Costa, L.D., Torelli, J. & Bruno, O. 2008, "2D Euclidean distance transform algorithms: A comparative survey", ACM Computing Surveys (CSUR), vol. 40, no. 1, pp. 1-44.

[6] Zucker, S. 2018, "Detection of periodicity based on independence tests – III. Phase distance correlation periodogram", Monthly notices of the Royal Astronomical Society. Letters, vol. 474, no. 1, pp. L86-L90.

[7] Faisal, M., Zamzami, E.M. & Sutarman 2020, "Comparative Analysis of Inter-Centroid K-Means Performance using Euclidean Distance, Canberra Distance and Manhattan Distance", Journal of physics. Conference series, vol. 1566, pp. 12112.

[8] Dharaneeshwaran, Nithya, S., Srinivasan, A. & Senthilkumar, M. 2017, "Calculating the user-item similarity using Pearson's and cosine correlation", IEEE, pp. 1000.

[9] Deselaers, T., Keysers, D. and Ney, H., 2003. Clustering visually similar images to improve image search engines. hist, 1(1), p.1.

[10] Alom, M.Z., Hasan, M., Yakopcic, C. and Taha, T.M., 2017. Inception recurrent convolutional neural network for object recognition. arXiv preprint arXiv:1704.07709.

[11] Kumari, K.A., Mouneeshwari, C., Udhaya, R.B. and Jasmitha, R., 2019, January. Automated Image Captioning for Flickr8K Dataset. In International Conference on Artificial Intelligence, Smart Grid and Smart City Applications (pp. 679-687). Springer, Cham.

[12] ping Tian, D., 2013. A review on image feature extraction and representation techniques. International Journal of Multimedia and Ubiquitous Engineering, 8(4), pp.385-396.

[13] Cui, J., Wen, F. and Tang, X., 2008, October. Real time google and live image search re-ranking. In Proceedings of the 16th ACM international conference on Multimedia (pp. 729-732).

[14] Adrakatti, A.F., Wodeyar, R.S. and Mulla, K.R., 2016. Search by image: a novel approach to content based image retrieval system. International Journal of Library Science, 14(3), pp.41-47.

[15] Paras Chopra, One neural network, many uses,1,March,2019, Online at https://towardsdatascience.com/one-neural-network-many-uses-image-captioning-image-search-similar-image-and-words-in-one-model-1e22080ce73d accessed 10,August 2020.

[16] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Fei-Fei, L., 2009, June. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee