An Integrated Approach to Near-duplicate Image Detection

Heesung Yang
School of Computer Science and Engineering
Kyungpook National University
Daegu, KOREA
hs.yang@knu.ac.kr

Hyeyoung Park (Corresponding Author)
School of Computer Science and Engineering
Kyungpook National University
Daegu, KOREA
hypark@knu.ac.kr

2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC) | 978-1-6654-5645-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICAIIC57133.2023.10067005

***Abstract*— Near-duplicate image detection is a task to find clusters of images that are considered to be the same pictures in human view. This is important in image recommendation systems, because when the systems recommend candidate images, redundancies of retrieved candidate images need to be avoided. In addition, in the era of big-data where image data is overflowing, its importance in terms of saving storage resources further increases. In this paper, we propose a robust model for detecting various types of near-duplicate images by integrating four different detection modules, where we use multiple image feature extractors such as Gabor filter and deep networks. The four modules are then integrated to conduct the multivariate log-likelihood ratio test for detecting duplication. Through computational experiments, we confirmed that our method reaches state-of-the-art performance.**

Keywords — deep learning features, feature integration, near-duplicate image detection, image recommendation system

I. INTRODUCTION

Nowadays there have been remarkable advances in camera and storage technology, and we store large size image data on storage using only simple compression techniques. In addition, there are many near-duplicate images taken by camera users who want to get better pictures or include some modulation by internet service providers. This is not desirable because it causes serious waste of computing resources. Furthermore, in the image recommendation system, which has widely been used, a problem has occurred due to near-duplicate images. For example, if the image recommendation system recommends five candidate images, and all five are slightly modulated from a single source image, this is the same as recommending one image. For these reasons, there have been suggested various ways to detect near-duplicate images.

Unlike exactly duplicated image pairs which have same size and identically equal value for each pixel, there are diverse cases of near-duplicate images with some modulation or modification, such as cut-and-paste and splicing. Therefore, near-duplicate image detection requires a clear definition of "near-duplication". In this paper, we use definitions for duplicate and near-duplicate images proposed by Connor et al. [1]. First, a pair of images with a perfectly identical data stream is defined as "duplicate". Otherwise, near-duplicate (ND) images are more strictly defined. Identical near duplicate images (IND) consist of images to which modulation has been applied from a single source image. Differently, images taken of the same object in the same scene are non-identical near-duplicate images (NIND). Fig. 1 shows examples of the IND and NIND pairs in Mir-Flickr Near Duplicate (MFND) Dataset [1], which is used as a benchmark dataset for duplicate image detection. The image pair on the left is an IND because it was made by modifying the brightness after cropping, and the pair on the right is an NIND because it shares the same object but is not modulated from a single source image.

IND Pair NIND Pair

Fig. 1. Example pairs of near-duplicate images

Generally, near-duplicate image detection is performed by extracting features from the image, comparing the distances between features, and determining them using thresholds. As a classical method, Connor et al. [1] extracts image features using perceptual hashing and calculates hamming distance. Among various classical feature extractors, the GIST descriptors has shown the best performance [2, 6]. However, these methods with handcrafted features have the limit of not including semantic information in the image.

In recent computer vision tasks, there has been dramatic increase in performances using convolutional neural networks (CNNs). In [3] and [4], it is shown that features extracted by deep convolutional neural networks are competitive in near-duplicate image detection too. However, the conventional CNN models have been trained for the classification task with ImageNet dataset, that is, they focus on extracting semantic features discriminating given object classes. In [5], a self-supervised model was used to solve this problem, and unlike other methods, image labels were not required during the training process. This method performs well in data with relatively large variations.

In this paper, the classical handcrafted features and the features from neural networks trained for different tasks are integrated to propose a robust detection model for datasets with diverse variations. The proposed method detects ND images first, and distinguishes between IND and NIND as well, which is the first trial in our knowledge. This can prevent the image recommendation system from recommending near-duplicate images when recommending multiple images, and thus it can

This work was supported by the Human Resources Program in Energy Technology of the Korea Institute of Energy Technology Evaluation and Planning(KETEP) granted financial resource from the Ministry of Trade, Industry & Energy, Republic of Korea (No. 20204010600060)

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2020R1A2C1010020).

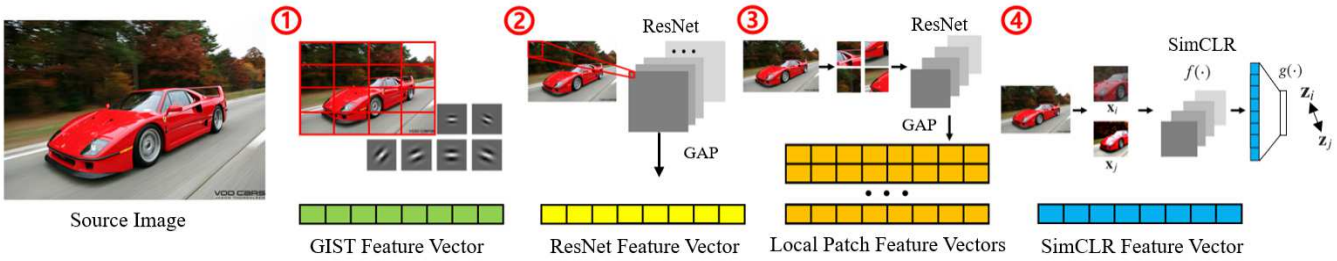

Fig. 2. Features used in the proposed method

provide the users with more appropriate varieties of candidate images.

In Section 2, we describe various feature extraction methods and how to determine near-duplicate images by integrating them. In Section 3, the experimental results are analyzed and compared with previous studies. Finally, conclusions of this paper are made in Section 4.

II. Proposed Method

A. Feature Extraction

Fig. 2 shows all the features used in this paper. First, we use GIST descriptor [6] to extract global shape features. Among various handcrafted features, GIST has shown relatively good performance in the previous work [2]. GIST is a bag-of-features obtained as the responses of convolution using Gabor filters. It extracts global features of the whole area of an image and can give robust representations to simple global variations. In this paper, the number of blocks is set to 8 and all other parameters were set the same as in [3].

Second, we use ResNet [7] to extract global semantic features. ResNet is a CNN model that achieves high performance in object recognition task. In this paper, we use the feature from the last global average pooling (GAP) layer of ResNet50 network pretrained with ImageNet. GAP is an operation that calculates the average values for each plane of convolution layer to transform a three-order tensor to a vector. The obtained feature vector represents global semantic informations, which is robust to more complex variations.

Third, we use randomly selected image patches and ResNet to extract local semantic features. To do this, image is resized to (224, 224), and 64 image patches of size (32, 32) are randomly selected. Then, we obtain 64 feature vectors by applying each local image patch to the pretrained ResNet. Similar to the global ResNet feature, we use the outputs from the last GAP layer.

These global and local ResNet features are extracted from the network trained for object recognition task. Therefore, there is a limitation that it is difficult to extract features capable of representing images of unlearned objects. To solve this problem, we extract features from a network trained for more general tasks by self-supervised contrastive learning strategy [5]. We use SimCLR [8] to obtain global features for general variations, which is trained to project two augmented images from a single source image closely on a vector space. For this reason, this model does not require class labels when learning, and can learn robust features in most visual representations when the learning batch size is large enough. We use ResNet50 [7] as a backbone network, and all the remaining configurations are set to be the same as [5].

B. Near-Duplicate Detection

We first calculate the distances between the two images by using each feature separately, and then integrate them to detect duplication. The distances of two feature vectors are basically calculated by the cosine distance. In case of random local patch features that are composed of 64 feature vectors, a cosine distance matrix is first obtained by calculating distances for all possible pairs of patches in the two images. For parallel computation, the cosine distance matrix is defined as

$$D_{\cos}(X_i, X_j) = \mathbb{1} - X_i X_j^T, \quad (1)$$

where $\mathbb{1}$ is a matrix of which every element has value 1, X_i is a matrix that has 64 normalized local features for i-th image as row vectors. From the matrix $D_{\cos}$, we choose three smallest elements and take their average as the distance of the two images.

Once the distances of all features are obtained, we concatenate them to get a distance vector $\boldsymbol{d}$ for an image pair and perform the log-likelihood ratio test. To proceed this, the mean and covariance matrix of the training data are calculated, and the distributions for the class of ND (Ω_{ND}) pairs and the class of NND (Ω_{NND}) pairs are estimated respectively, under the Gaussian assumption. Using the estimated distributions for the training set, we can calculate the log-likelihood ratios for the test image pairs. The ratio is defined as

$$\mathrm{r} = \log \frac{\mathrm{P}\left(\Omega_{\mathrm{ND}} \middle| \boldsymbol{d}\right)}{\mathrm{P}\left(\Omega_{\mathrm{NND}} \middle| \boldsymbol{d}\right)} = \log \frac{\mathrm{P}\left(\boldsymbol{d} \middle| \Omega_{\mathrm{ND}}\right)\mathrm{P}\left(\Omega_{\mathrm{ND}}\right)}{\mathrm{P}\left(\boldsymbol{d} \middle| \Omega_{\mathrm{NND}}\right)\mathrm{P}\left(\Omega_{\mathrm{NND}}\right)}$$
$$= \log \mathrm{P}\left(\boldsymbol{d} \middle| \Omega_{\mathrm{ND}}\right)\mathrm{P}\left(\Omega_{\mathrm{ND}}\right) - \log \mathrm{P}\left(\boldsymbol{d} \middle| \Omega_{\mathrm{NND}}\right)\mathrm{P}\left(\Omega_{\mathrm{NND}}\right). \quad (2)$$

For each ratio, if the value is greater than a predefined threshold, then the given pair is determined by ND, the opposite is determined by NND.

Furthermore, we also discriminate IND and NIND pair, by estimating the distributions of IND and NIND classes separately. In the training phase, we compute the mean and covariance matrix of the set of IND pairs and the set of NIND pairs, respectively, for estimating $\mathrm{P}(\boldsymbol{d}|\Omega_{\mathrm{IND}})$ and $\mathrm{P}(\boldsymbol{d}|\Omega_{\mathrm{NIND}})$ with Gaussian assumption. In the test phase, it is assigned to a class with the greatest probability for each $\boldsymbol{d}$.

III. Experimental Results

In order to verify efficiency of the proposed method, we conduct experiments on California-ND [10] and MFND dataset [1], and compare with the conventional methods. California-ND dataset has 701 photo images taken by individuals, and is a representative NIND dataset. For each image pair, the degree of duplication was measured manually between 0 and 1. In this

work, we labelled each data pairs as NIND or NND using a threshold 0.5. The test set for detection task consists of 2607 NIND pairs and the same number of NND pairs, which are randomly selected from the total 242743 NND pairs. The MFND dataset contains duplicate, IND, NIND, and NND pairs found in the Mir-Flickr dataset. According to [5], we compose two different test set: MFND-IND and MFND-ALL. MFND-IND consists of 5036 duplicate and IND pairs and the same number of NND pairs; MFND-ALL includes 23178 duplicate, IND, and NIND pairs, and the same number of NND pairs.

TABLE III. DETECTION PERFORMANCE (AUROC) OF SINGLE FEATURE

Features	California-ND	MFND-IND	MFND-ALL
	AUROC	AUROC	AUROC
GIST8	0.880	0.957	0.881
ResNet50	0.945	**0.999**	**0.999**
Local Patch	0.957	0.988	0.971
SimCLR	**0.995**	0.996	0.994

TABLE IV. DETECTION PERFORMANCE (F1 SCORE) OF SINGLE FEATURE

Features	California-ND	MFND-IND	MFND-ALL
	F1 Score	F1 Score	F1 Score
GIST8	0.811	0.909	0.797
ResNet50	0.896	**0.991**	**0.994**
Local Patch	0.918	0.950	0.920
SimCLR	**0.969**	0.976	0.963

First, we evaluated the detection performance using only single feature as a baseline. To obtain the area under ROC (AUROC) values and the best F1-scores, we calculated the detection accuracy on each of 1000 different thresholds within [0, 2]. Tables I and II show the experimental results. In the case of California-ND, a dataset with relatively large variance, SimCLR features have relatively high performance, which are not related to the object recognition task. On the other hand, on MFND-IND and MFND-ALL datasets with relatively small variance and many IND pairs, features obtained from ResNet trained for object recognition show relatively high performances.

Table III shows the results of experiments using integrated features. In this experiment, we randomly select a set of training samples for estimating probability density functions, $P\left(\boldsymbol{d}\middle|\Omega_{\mathrm{ND}}\right)$ and $P\left(\boldsymbol{d}\middle|\Omega_{\mathrm{NND}}\right)$. The performance is evaluated as the average of the values obtained through 5-fold cross validation. The results show that the integration of the global ResNet feature and SimCLR feature works robustly for all variances, which is better than the single feature methods. Fig. 3 shows the scatter plots and kernel density estimation (KDE) plots for the distances calculated with the ResNet and SimCLR features. In California-ND data, while the ResNet feature overlaps relatively more than SimCLR feature resulting in significant performance improvements by combining two features. Additionally, the KDE plots of MFND-ALL show that the distances for IND pairs are closer to zero than the NIND pairs.

TABLE IV. DETECTION PERFORMANCE ON UNTRAINED DATA

Integration Methods	AUROC	F1 Score
GIST, ResNet	0.955	0.921
GIST, Local Patch	0.974	0.938
GIST, SimCLR	**0.995**	**0.972**
ResNet, Local Patch	0.957	0.922
ResNet, SimCLR	0.978	0.928
Local Patch, SimCLR	0.986	0.966
GIST, ResNet, Local Patch	0.968	0.931
GIST, ResNet, SimCLR	0.936	0.936
GIST, Local Patch, SimCLR	0.989	0.968
ResNet, Local Patch, SimCLR	0.980	0.936
Using Every Features	0.984	0.941

Additionally, we consider more practical environments, where the distance distribution cannot be estimated from the same dataset. Assuming the situation, we estimate the distance distributions using MFND-ALL dataset and evaluate the performance on California-ND dataset. Table IV shows result, the accuracy on California-ND data. Notably, the integration of GIST and SimCLR feature shows the highest performance. From this, the features obtained through the self-supervised contrastive learning task are more robust to unseen samples than the features obtained through the object recognition task.

TABLE V. MULTI CLASSES (NND, IND, NIND) CLASSIFICATION ACCURACY

Integration Methods	Accuracy
GIST, ResNet	0.858
GIST, Local Patch	0.816
GIST, SimCLR	0.812
ResNet, Local Patch	0.853
ResNet, SimCLR	0.800
Local Patch, SimCLR	0.808
GIST, ResNet, Local Patch	**0.869**
GIST, ResNet, SimCLR	0.848
GIST, Local Patch, SimCLR	0.835
ResNet, Local Patch, SimCLR	0.828
Using Every Features	0.858

Furthermore, we also conduct an additional experiment that subdivides ND class into IND and NIND classes. To our knowledge, this is the first study to distinguish the degree of near-duplication of image pairs. We experiment with MFND-ALL because MFND-ALL is the only dataset that includes IND, NIND, and NND all. In test phase, each pair is assigned to the class with the highest probability of $P\left(\Omega_{\mathrm{IND}}\middle|\boldsymbol{d}\right)$, $P\left(\Omega_{\mathrm{NIND}}\middle|\boldsymbol{d}\right)$, and $P\left(\Omega_{\mathrm{NND}}\middle|\boldsymbol{d}\right)$. Table V shows the results of average accuracies obtained from 5-folds cross validation. From the result, we can see the highest accuracy by the integration of GIST, ResNet, and local patch features.

TABLE III. 5-FOLDS MULTI FEATURE RESULTS OF AUROC AND F1-SCORE

Integration Methods		California-ND		MFND-IND		MFND-ALL	
		AUROC	F1 Score	AUROC	F1 Score	AUROC	F1 Score
2 Features	GIST, ResNet	0.986	0.939	**0.999**	0.994	**1.000**	**0.995**
	GIST, Local Patch	0.986	0.961	0.994	0.981	0.980	0.946
	GIST, SimCLR	0.994	0.971	0.996	0.976	0.994	0.965
	ResNet, Local Patch	0.974	0.925	**0.999**	0.993	0.999	0.994
	ResNet, SimCLR	**0.996**	0.975	0.998	0.991	0.999	0.993
	Local Patch, SimCLR	0.993	0.973	0.997	0.987	0.995	0.975
3 Features	GIST, ResNet, Local Patch	0.990	0.963	**0.999**	**0.996**	**1.000**	**0.995**
	GIST, ResNet, SimCLR	0.995	0.976	0.998	0.992	0.999	0.994
	GIST, Local Patch, SimCLR	0.994	0.975	0.997	0.988	0.996	0.977
	ResNet, Local Patch, SimCLR	0.995	0.977	0.998	0.992	0.999	0.994
Using Every Features		0.995	**0.978**	0.998	0.994	0.999	0.994

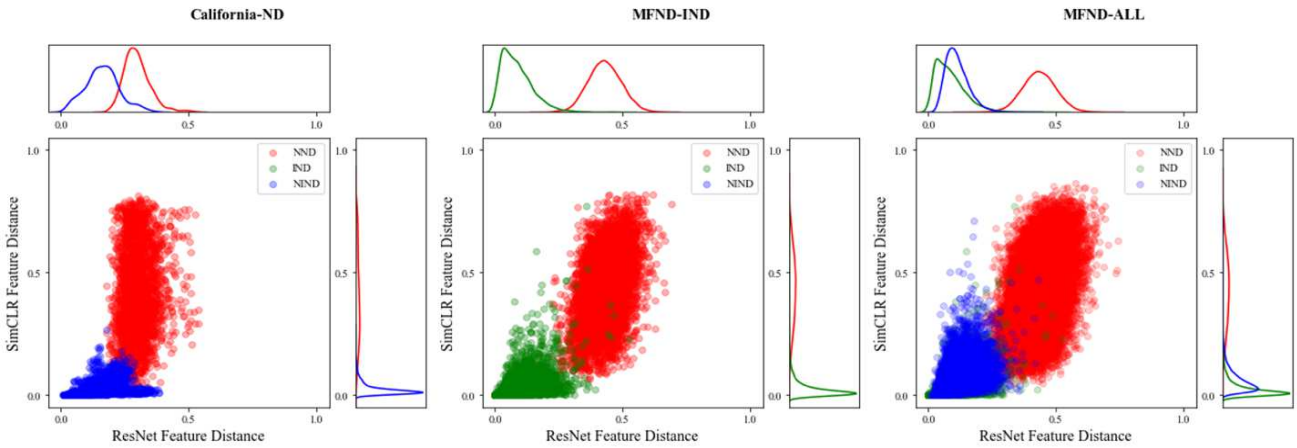

Fig. 3. Scatter plot and kernel density estimate plots of the distances of ResNet features and SimCLR features.

IV. Conclusion

In this paper, we propose a method to detect near-duplicate images by integrating various features rather than using one visual representation. We confirm that the integration of the features from deep networks trained in various ways as well as the handcrafted features achieves state-of-the-art performance. In particular, we show that combination of SimCLR feature trained by self-supervised learning and ResNet feature trained with object recognition task achieves the best for in-sample setting, while the combination of GIST and SimCLR is the best in the practical situations. In addition, we also confirm that the proposed method can be applied to discriminating the types of duplication (IND and NIND), which can provide users with various ways to control duplications.

Research on near-duplicate images has very limited dataset for the performance evaluation. Therefore, it is not clear whether the methods proposed in previous studies work well in practical situations. For this reason, we conducted out of sample experiments, but this is also not enough to verify the performance. In the future, it is necessary to build a new dataset for near-duplicate image detection task and to develop a more elaborate model for detecting IND and NIND.

References

[1] Connor, Richard, et al. "Identification of MIR-Flickr near-duplicate images: A benchmark collection for near-duplicate detection." 10th International Conference on Computer Vision Theory and Applications (VISAPP 2015). 2015.

[2] Connor, Richard, and Franco Alberto Cardillo. "Quantifying the specificity of near-duplicate image classification functions." 11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. 2016.

[3] Morra, Lia, and Fabrizio Lamberti. "Benchmarking unsupervised near-duplicate image detection." Expert Systems with Applications 135 (2019): 313-326.

[4] Yang, Heesung, et al. "Deduplication of Retrieved Image Data Using Deep Network Features." Proceedings of the Korean Information Science Society Conference (2021): 1518-1520.

[5] Yang, Heesung, and Park Hyeyoung "Near-Duplicate Image Detection Using a Self-Supervised Learning Model" The Journal of Korean Institute of Next Generation Computing 18.3 pp.75-84 (2022) : 75.

[6] Oliva, Aude, and Antonio Torralba. "Modeling the shape of the scene: A holistic representation of the spatial envelope." International journal of computer vision 42.3 (2001): 145-175.

[7] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[8] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.

[9] Coates, Adam, Andrew Ng, and Honglak Lee. "An analysis of single-layer networks in unsupervised feature learning." Proceedings of the fourteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2011.

[10] Jinda-Apiraksa, Amornched, Vassilios Vonikakis, and Stefan Winkler. "California-ND: An annotated dataset for near-duplicate detection in personal photo collections." 2013 Fifth International Workshop on Quality of Multimedia Experience (QoMEX). IEEE, 2013.