information

MDPI

Article

Image Retrieval via Canonical Correlation Analysis and Binary Hypothesis Testing [†]

Kangdi Shi [1,*], Xiaohong Liu [1], Muhammad Alrabeiah [1], Xintong Guo [1], Jie Lin [2], Huan Liu [1] and Jun Chen [1]

[1] The Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 4S1, Canada; liux173@mcmaster.ca (X.L.); alrabm@mcmaster.ca (M.A.); guox127@mcmaster.ca (X.G.); liuh127@mcmaster.ca (H.L.); chenjun@mcmaster.ca (J.C.)

[2] The Institute for Infocomm Research, A-STAR, Singapore 138632, Singapore; lin-j@i2r.a-star.edu.sg

[*] Correspondence: shik9@mcmaster.ca

[†] This paper is an extended version of our presentation at the 16th Canadian Workshop on Information Theory, Hamilton, ON, Canada, 2–5 June 2019.

Abstract: Canonical Correlation Analysis (CCA) is a classic multivariate statistical technique, which can be used to find a projection pair that maximally captures the correlation between two sets of random variables. The present paper introduces a CCA-based approach for image retrieval. It capitalizes on feature maps induced by two images under comparison through a pre-trained Convolutional Neural Network (CNN) and leverages basis vectors identified through CCA, together with an element-wise selection method based on a Chernoff-information-related criterion, to produce compact transformed image features; a binary hypothesis test regarding the joint distribution of transformed feature pair is then employed to measure the similarity between two images. The proposed approach is benchmarked against two alternative statistical methods, Linear Discriminant Analysis (LDA) and Principal Component Analysis with whitening (PCAw). Our CCA-based approach is shown to achieve highly competitive retrieval performances on standard datasets, which include, among others, Oxford5k and Paris6k.

Keywords: canonical correlation analysis; chernoff information; hypothesis testing; image retrieval; multivariate gaussian distribution

check for updates

Citation: Shi, K.; Liu, X.; Alrabeiah, M.; Guo, X.; Lin, J.; Liu, H.; Chen, J. Image Retrieval via Canonical Correlation Analysis and Binary Hypothesis Testing. *Information* **2022**, *13*, 106. https://doi.org/10.3390/info13030106

Academic Editor: Vincenzo Moscato

Received: 7 January 2022
Accepted: 18 February 2022
Published: 23 February 2022

1. Introduction

The past two decades have witnessed an explosive growth of online image databases. This growth paves the way for the development of visual-data-driven applications, but at the same time poses a major challenge to the Content-Based Image Retrieval (CBIR) technology [1].

Traditional approaches to CBIR mostly rely on the exploitation of handrafted scale- and orientation-invariant image features [2–6], which have achieved varying degrees of success. Recent advances [7,8] in Deep Learning (DL) for image classification and object detection have generated significant interests in bringing Convolutional Neural Networks (CNNs) to bear upon CBIR. Although CNN models are usually trained for purposes different from CBIR, it is known [9] that features extracted from modern deep CNNs, commonly referred to as DL features, have great potential in this respect as well. Retrieval methods utilizing DL features can generally be divided into two categories: without/with fine-tuning the CNN model [10]. The early application of CNN to CBIR almost exclusively resorts to methods in the first category, which use Off-The-Shelf (OTS) CNNs (i.e., popular pre-trained CNNs) for feature extraction (see, e.g., [11–14]). A main advantage of such methods is the low implementation cost [15,16], which is largely attributed to the direct adoption of pre-trained CNNs. Performance-wise, they are comparable to the state-of-the-art traditional methods that rely on handcrafted features. In contrast, many recent methods, such as [17–19], belong to the second category, which take advantage of the fine-tuning gain to enhance the discriminatory power of the extracted DL features. A top representative from this category is the end-to-end learning framework proposed in [20]. It outperforms most existing traditional and OTS-CNN-based methods on standard testing datasets; however, this performance improvement comes at the cost of training a complex triple-branched CNN using a large dataset, which might not always be affordable in practice.

Many preprocessing methods have been developed with the goal of better utilizing DL features for image retrieval, among which Principal Component Analysis with whitening [21] (PCAw) and Linear Discriminant Analysis [22] (LDA) are arguably most well known. Despite being extremely popular, PCA and LDA have their respective weaknesses: the dimensionality reduction in PCA often leads to the elimination of critical principal components with a small contribution rate while the performance of LDA tends to suffer from decreasing differences between mismatched features. As such, there is great need for a preprocessing method with improved robustness against dimensionality reduction and enhanced sensitivity to feature mismatch. In this work, we aim to put forward a potential solution with desired properties by bringing Canonical Correlation Analysis (CCA) [23] into the picture.

CCA is a multivariate technique for elucidating the the associations among two sets of variables. It can be used to identify a projection pair of a given dimension that maximally captures the correlation between the two sets. The applications of CCA are too numerous to list. In cross-modality matching/retrieval alone, extensive investigations have been carried out as evidenced by a growing body of literature, from those based on handcrafted features [24] to the more recent ones that make use of DL features [25–27]. There is also some related development on the theoretical front (see, e.g., [28,29]).

Motivated by the consideration of computational efficiency and affordability as well as the weaknesses inherent in the existing preprocessing methods, we develop and present in this paper a new image retrieval method based on OTS deep CNNs. Our method is built primarily upon CCA, but has several notable differences from the related works. For the purpose of dimensionality reduction (i.e., feature compression), the proposed method employs a basis-vector selection technique, which invokes a Chernoff-information-based criterion to rank how discriminative the basis vectors are. Both the basis vectors and their ranking are learned from a training set, which consists of features extracted from a pre-trained CNN—the neural network itself is not retrained/finetuned in our work. Given a new pair of features, the ranked basis vectors are used to perform transformation and compression. This is followed by a binary hypothesis test on the joint distribution of pairs of transformed features, which yields a matching score that can be leveraged to identify top candidates for retrieval. We show via extensive experimental results that the proposed CCA-based method is able to deliver highly competitive results on standard datasets, which include, among others, Oxford5k and Parise6k.

This paper is organized as follows. The proposed CCA-based preprocessing method along with the associated matching procedure is detailed in Section 2. Section 3 includes the experimental results and the relevant discussions. We close the paper in Section 4 with some concluding remarks.

2. Proposed Method

The proposed image retrieval method utilizes CCA in an essential way. It leverages a training dataset of features extracted from a pre-trained CNN model (see Figure 1) to learn a set of canonical vectors, which serve as the basis vectors of the feature space. These vectors are used to project the features of a pair of images into a new space, in which a Chernoff-information-based selection method is applied to identify the most discriminative elements of the transformed features. Such elements then undergo a binary hypothesis test to measure the similarity between the features and, consequently, the two images. This process is expounded in the following four subsections (see also Figure 2).

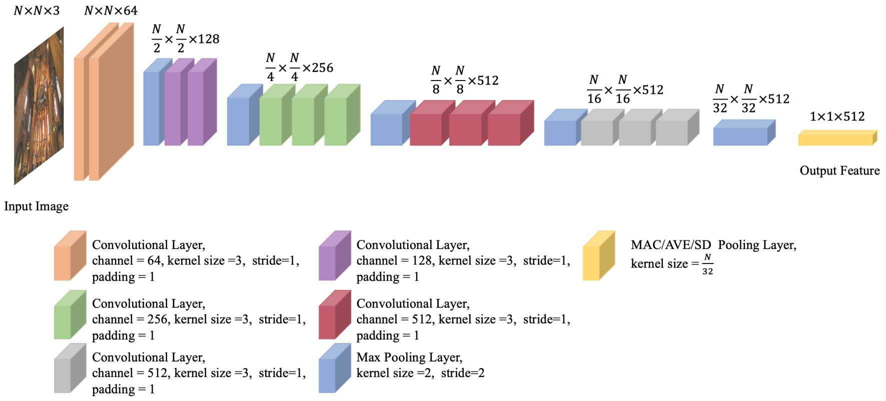

$N \times N \times 3$
$N \times N \times 64$
$\frac{N}{2} \times \frac{N}{2} \times 128$
$\frac{N}{4} \times \frac{N}{4} \times 256$
$\frac{N}{8} \times \frac{N}{8} \times 512$
$\frac{N}{16} \times \frac{N}{16} \times 512$
$\frac{N}{32} \times \frac{N}{32} \times 512$
1×1×512
Output Feature
Input Image
Convolutional Layer, channel = 64, kernel size =3, stride=1, padding = 1
Convolutional Layer, channel = 128, kernel size =3, stride=1, padding = 1
MAC/AVE/SD Pooling Layer, kernel size = $\frac{N}{32}$
Convolutional Layer, channel = 256, kernel size =3, stride=1, padding = 1
Convolutional Layer, channel = 512, kernel size =3, stride=1, padding = 1
Convolutional Layer, channel = 512, kernel size =3, stride=1, padding = 1
Max Pooling Layer, kernel size =2, stride=2

Figure 1. Modified VGG16 for feature extraction.

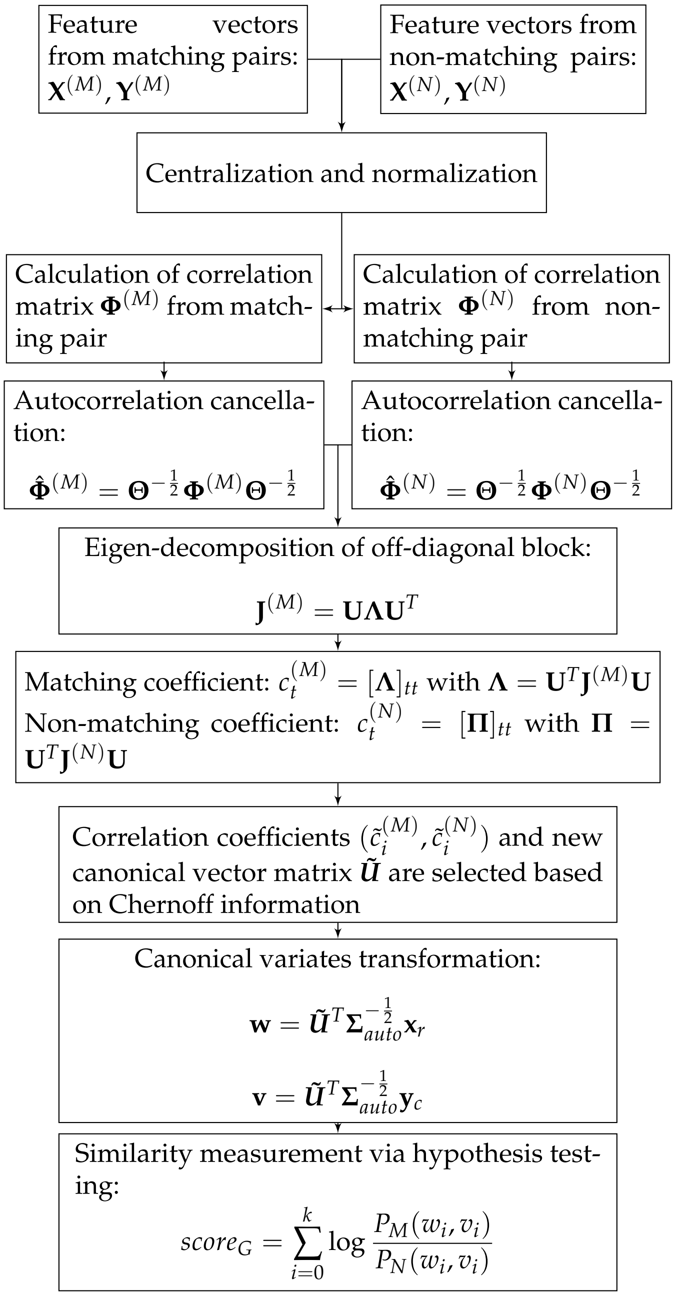

Feature vectors from matching pairs: $\mathbf{X}^{(M)}, \mathbf{Y}^{(M)}$
Feature vectors from non-matching pairs: $\mathbf{X}^{(N)}, \mathbf{Y}^{(N)}$
Centralization and normalization
Calculation of correlation matrix $\mathbf{\Phi}^{(M)}$ from matching pair
Calculation of correlation matrix $\mathbf{\Phi}^{(N)}$ from non-matching pair
Autocorrelation cancellation: $\hat{\mathbf{\Phi}}^{(M)} = \mathbf{\Theta}^{-\frac{1}{2}}\mathbf{\Phi}^{(M)}\mathbf{\Theta}^{-\frac{1}{2}}$
Autocorrelation cancellation: $\hat{\mathbf{\Phi}}^{(N)} = \mathbf{\Theta}^{-\frac{1}{2}}\mathbf{\Phi}^{(N)}\mathbf{\Theta}^{-\frac{1}{2}}$
Eigen-decomposition of off-diagonal block: $\mathbf{J}^{(M)} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$
Matching coefficient: $c_t^{(M)} = [\mathbf{\Lambda}]_{tt}$ with $\mathbf{\Lambda} = \mathbf{U}^T\mathbf{J}^{(M)}\mathbf{U}$
Non-matching coefficient: $c_t^{(N)} = [\mathbf{\Pi}]_{tt}$ with $\mathbf{\Pi} = \mathbf{U}^T\mathbf{J}^{(N)}\mathbf{U}$
Correlation coefficients $(\tilde{c}_i^{(M)}, \tilde{c}_i^{(N)})$ and new canonical vector matrix $\tilde{\boldsymbol{U}}$ are selected based on Chernoff information
Canonical variates transformation: $\mathbf{w} = \tilde{\boldsymbol{U}}^T\mathbf{\Sigma}_{auto}^{-\frac{1}{2}}\mathbf{x}_r$ $\mathbf{v} = \tilde{\boldsymbol{U}}^T\mathbf{\Sigma}_{auto}^{-\frac{1}{2}}\mathbf{y}_c$
Similarity measurement via hypothesis testing: $score_G = \sum_{i=0}^{k} \log \frac{P_M(w_i, v_i)}{P_N(w_i, v_i)}$

Figure 2. Block diagram of the proposed method.

2.1. Image Pre-Processing and Feature Extraction

The CNN model adopted in this work for feature extraction is VGG16 [30]. It takes an input image of maximum size 1024 × 1024 and produces 512 feature maps of maximum size 32 × 32 from its very last pooling layer. A single feature element is extracted from each feature map via pooling. A 512-dimensional vector, which resulted from the concatenation of these elements, is converted, through centralization and normalization (here centralization is performed by subtracting the mean (computed based on the training set) while normalization yields a unit-length vector), to a global feature vector, which serves as a compact representation of the image.

2.2. Correlation Analysis and Canonical Vectors

At the heart of the proposed method lies so-called canonical vectors, which are learned from a large training set of matching and non-matching image features in a manner inspired by CCA. The learning process consists of the following steps.

Step 1 : Construct two raw matching matrices

$$\begin{aligned}\mathbf{X}^{(RM)} &= [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L],\\ \mathbf{Y}^{(RM)} &= [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L],\end{aligned}$$

where L is the number of raw matching pairs, $\mathbf{x}_l$ and $\mathbf{y}_l$ for $l \in \{1, 2, ..., L\}$ are a pair of global feature vectors representing two matching images (here “matching images" means images from the same class while “non-matching images" means images from different classes).

Using the raw matching pairs $\mathbf{X}^{(RM)}$ and $\mathbf{Y}^{(RM)}$, a pair of matching-feature matrices is formed:

$$\begin{aligned}\mathbf{X}^{(M)} &= [\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2, ...\mathbf{y}_L, \mathbf{x}_L],\\ \mathbf{Y}^{(M)} &= [\mathbf{y}_1, \mathbf{x}_1, \mathbf{y}_2, \mathbf{x}_2, ..., \mathbf{x}_L, \mathbf{y}_L].\end{aligned}$$

The total number of training pairs is $2L$ after feature order flipped. This is performed to ensure that in Equation (1) below, the diagonal blocks are identical and symmetric, so are the off-diagonal blocks. The size of both $\mathbf{X}^{(M)}$ and $\mathbf{Y}^{(M)}$ is 512 × $2L$. The training data matrix of matching features $\mathbf{H}^{(M)}$ is constructed by stacking $\mathbf{X}^{(M)}$ on $\mathbf{Y}^{(M)}$:

$$\mathbf{H}^{(M)} = \begin{bmatrix}\mathbf{X}^{(M)}\\ \mathbf{Y}^{(M)}\end{bmatrix}_{(1024\times 2L)}.$$

The estimated covariance matrix of matching features is given by

$$\begin{aligned}\mathbf{\Phi}^{(M)} &= \frac{1}{2L-1}\mathbf{H}^{(M)}(\mathbf{H}^{(M)})^T\\ &= \frac{1}{2L-1}\begin{bmatrix}\mathbf{X}^{(M)}\\ \mathbf{Y}^{(M)}\end{bmatrix}\begin{bmatrix}\mathbf{X}^{(M)}\\ \mathbf{Y}^{(M)}\end{bmatrix}^T\\ &= \begin{bmatrix}\mathbf{\Sigma}_{XX}^{(M)} & \mathbf{\Sigma}_{XY}^{(M)}\\ \mathbf{\Sigma}_{YX}^{(M)} & \mathbf{\Sigma}_{YY}^{(M)}\end{bmatrix},\end{aligned} \tag{1}$$

where

$$\begin{aligned}\mathbf{\Sigma}_{XX}^{(M)} &= \frac{\mathbf{X}^{(M)}(\mathbf{X}^{(M)})^T}{2L-1}, \quad \mathbf{\Sigma}_{YY}^{(M)} = \frac{\mathbf{Y}^{(M)}(\mathbf{Y}^{(M)})^T}{2L-1},\\ \mathbf{\Sigma}_{XY}^{(M)} &= \mathbf{\Sigma}_{YX}^{(M)} = \frac{\mathbf{X}^{(M)}(\mathbf{Y}^{(M)})^T}{2L-1} = \frac{\mathbf{Y}^{(M)}(\mathbf{X}^{(M)})^T}{2L-1}.\end{aligned}$$

Step 2: Randomly permuting the columns of one of the raw feature matrices, say from $\mathbf{Y}^{(RM)}$ to $\mathbf{Y}^{(RN)}$, yields two raw non-matching matrices. More specifically, we construct two raw non-matching matrices by successively associating each column of $\mathbf{X}^{(RM)}$ with a randomly selected (without replacement) non-matching column from $\mathbf{Y}^{(RM)}$. For example,

$$\begin{aligned}\mathbf{X}^{(RN)} &= [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L],\\ \mathbf{Y}^{(RN)} &= [\mathbf{y}_3, \mathbf{y}_7, \ldots, \mathbf{y}_{L-4}].\end{aligned}$$

Based on these two raw non-matching matrices, the feature order flipping is performed to generate $\mathbf{X}^{(N)}$ and $\mathbf{Y}^{(N)}$:

$$\mathbf{X}^{(N)} = [\mathbf{x}_1, \mathbf{y}_3, \mathbf{x}_2, \mathbf{y}_7, ..., \mathbf{x}_L, \mathbf{y}_{L-4}],$$

$$\mathbf{Y}^{(N)} = [\mathbf{y}_3, \mathbf{x}_1, \mathbf{y}_7, \mathbf{x}_2, ..., \mathbf{y}_{L-4}, \mathbf{x}_L].$$

With a procedure similar to that of *step 1*, we can estimate the covariance matrix $\mathbf{\Phi}^{(N)}$ for non-matching features $\mathbf{H}^{(N)}$:

$$\begin{aligned}\mathbf{\Phi}^{(N)} &= \frac{1}{2L-1}\mathbf{H}^{(N)}(\mathbf{H}^{(N)})^T\\ &= \frac{1}{2L-1}\begin{bmatrix}\mathbf{X}^{(N)}\\ \mathbf{Y}^{(N)}\end{bmatrix}\begin{bmatrix}\mathbf{X}^{(N)}\\ \mathbf{Y}^{(N)}\end{bmatrix}^T\\ &= \begin{bmatrix}\mathbf{\Sigma}^{(N)}_{XX} & \mathbf{\Sigma}^{(N)}_{XY}\\ \mathbf{\Sigma}^{(N)}_{YX} & \mathbf{\Sigma}^{(N)}_{YY}\end{bmatrix}.\end{aligned} \tag{2}$$

Note that

$$\mathbf{\Sigma}^{(N)}_{XX} = \mathbf{\Sigma}^{(N)}_{YY} = \mathbf{\Sigma}^{(M)}_{XX} = \mathbf{\Sigma}^{(M)}_{YY} = \mathbf{\Sigma}_{auto},$$

for they are the covariances of sets of random image features. As in Equation (1), the diagonal blocks in Equation (2) are also identical and symmetric, so are the off-diagonal blocks.

Step 3: Since $\mathbf{\Sigma}_{auto}$ is positive definite, it follows that $\mathbf{\Theta}^{-\frac{1}{2}}$ is well defined, where

$$\mathbf{\Theta} = \begin{bmatrix}\mathbf{\Sigma}_{auto} & \mathbf{0}\\ \mathbf{0} & \mathbf{\Sigma}_{auto}\end{bmatrix}.$$

We can multiply both covariance matrices, $\mathbf{\Phi}^{(M)}$ and $\mathbf{\Phi}^{(N)}$, on the left and right by $\mathbf{\Theta}^{-\frac{1}{2}}$ to de-correlate their diagonal blocks:

$$\begin{aligned}\hat{\mathbf{\Phi}}^{(M)} &= \mathbf{\Theta}^{-\frac{1}{2}}\mathbf{\Phi}^{(M)}\mathbf{\Theta}^{-\frac{1}{2}} = \begin{bmatrix}\mathbf{I} & \mathbf{J}^{(M)}\\ \mathbf{J}^{(M)} & \mathbf{I}\end{bmatrix},\\ \hat{\mathbf{\Phi}}^{(N)} &= \mathbf{\Theta}^{-\frac{1}{2}}\mathbf{\Phi}^{(N)}\mathbf{\Theta}^{-\frac{1}{2}} = \begin{bmatrix}\mathbf{I} & \mathbf{J}^{(N)}\\ \mathbf{J}^{(N)} & \mathbf{I}\end{bmatrix},\end{aligned}$$

where

$$\begin{aligned}\mathbf{J}^{(M)} &= \mathbf{\Sigma}_{auto}^{-\frac{1}{2}}\mathbf{\Sigma}^{(M)}_{XY}\mathbf{\Sigma}_{auto}^{-\frac{1}{2}} = \mathbf{\Sigma}_{auto}^{-\frac{1}{2}}\mathbf{\Sigma}^{(M)}_{YX}\mathbf{\Sigma}_{auto}^{-\frac{1}{2}},\\ \mathbf{J}^{(N)} &= \mathbf{\Sigma}_{auto}^{-\frac{1}{2}}\mathbf{\Sigma}^{(N)}_{XY}\mathbf{\Sigma}_{auto}^{-\frac{1}{2}} = \mathbf{\Sigma}_{auto}^{-\frac{1}{2}}\mathbf{\Sigma}^{(N)}_{YX}\mathbf{\Sigma}_{auto}^{-\frac{1}{2}}.\end{aligned}$$

Step 4: Apply eigen-decomposition [31] on $\mathbf{J}^{(M)}$:

$$\mathbf{J}^{(M)} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T,$$

where $\mathbf{U}$ is a unitary matrix, and $\mathbf{\Lambda}$ is a diagonal matrix with the diagonal entries being the eigenvalues of $\mathbf{J}^{(M)}$. The columns of $\mathbf{U}$ are exactly the sought-after canonical vectors. The blockwise left- and right-multiplication of both $\hat{\mathbf{\Phi}}^{(M)}$ and $\hat{\mathbf{\Phi}}^{(N)}$ by $\mathbf{U}^T$ and $\mathbf{U}$, respectively, gives the following pair of matrices:

$$\begin{bmatrix} \mathbf{U}^T\mathbf{U} & \mathbf{U}^T\mathbf{J}^{(M)}\mathbf{U} \\ \mathbf{U}^T\mathbf{J}^{(M)}\mathbf{U} & \mathbf{U}^T\mathbf{U} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{\Lambda} \\ \mathbf{\Lambda} & \mathbf{I} \end{bmatrix}, \tag{3}$$

$$\begin{bmatrix} \mathbf{U}^T\mathbf{U} & \mathbf{U}^T\mathbf{J}^{(N)}\mathbf{U} \\ \mathbf{U}^T\mathbf{J}^{(N)}\mathbf{U} & \mathbf{U}^T\mathbf{U} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{\Pi} \\ \mathbf{\Pi} & \mathbf{I} \end{bmatrix}, \tag{4}$$

where $\mathbf{\Pi} = \mathbf{U}^T\mathbf{J}^{(N)}\mathbf{U}$. The off-diagonal block $\mathbf{\Lambda}$ in Equation (3) is a diagonal matrix whereas $\mathbf{\Pi}$ in Equation (4) is not necessarily so. Nevertheless, it will be seen that in practice $\mathbf{\Pi}$ is often close to a zero matrix (as two non-matching image features tend to be uncorrelated) and thus is approximately diagonal as well.

2.3. Chernoff Information for Canonical Vector Selection

Note that the learned canonical vectors of matching image features form an orthonormal basis of $\mathbb{R}^{512}$. These vectors are not necessarily equally useful for the purpose of measuring the similarity between two feature vectors of an unknown pair of images; therefore, it is of considerable interest to quantify how *discriminative* each canonical vector is. To this end, the off-diagonal blocks of the covariance matrix of non-matching image features can be brought into play. Evaluating Chernoff information (CI) [32,33] with respect to the diagonal elements of both $\mathbf{\Lambda}$ and $\mathbf{\Pi}$ yields a ranking of the most different diagonal element pairs, which can be used to guide the selection of canonical vectors.

Define the following set of 2 × 2 matrices

$$\mathbf{S}_t^{(M)} = \begin{bmatrix} 1 & c_t^{(M)} \\ c_t^{(M)} & 1 \end{bmatrix},$$
$$\mathbf{S}_t^{(N)} = \begin{bmatrix} 1 & c_t^{(N)} \\ c_t^{(N)} & 1 \end{bmatrix},$$

using matching coefficient $c_t^{(M)} = [\mathbf{\Lambda}]_{tt}$ and non-matching coefficient $c_t^{(N)} = [\mathbf{\Pi}]_{tt}$, $t \in \{1, 2, \ldots, 512\}$, determined by the diagonal elements of $\mathbf{\Lambda}$ and $\mathbf{\Pi}$:

$$\mathbf{\Lambda} = \begin{bmatrix} c_1^{(M)} & 0 & \ldots & 0 \\ 0 & c_2^{(M)} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & c_{512}^{(M)} \end{bmatrix},$$
$$\mathbf{\Pi} = \begin{bmatrix} c_1^{(N)} & \pi_{1,2} & \ldots & \pi_{1,512} \\ \pi_{2,1} & c_2^{(N)} & \ldots & \pi_{2,512} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{512,1} & \pi_{512,2} & \ldots & c_{512}^{(N)} \end{bmatrix}.$$

Now let $\mathbf{S}_t^{(\lambda_t)} = (\lambda_t(\mathbf{S}_t^{(M)})^{-1} + (1-\lambda_t)(\mathbf{S}_t^{(N)})^{-1})^{-1}$, $\lambda_t \in [0,1]$ and define

$$D(\mathbf{S}_t^{(\lambda_t)} || \mathbf{S}_t^{(M)}) = \frac{1}{2}\log_e \frac{|\mathbf{S}_t^{(M)}|}{|\mathbf{S}_t^{(\lambda_t)}|} + \frac{1}{2}\operatorname{tr}((\mathbf{S}_t^{(M)})^{-1}\mathbf{S}_t^{(\lambda_t)}) - 1,$$

$$D(\mathbf{S}_t^{(\lambda_t)} || \mathbf{S}_t^{(N)}) = \frac{1}{2}\log_e \frac{|\mathbf{S}_t^{(N)}|}{|\mathbf{S}_t^{(\lambda_t)}|} + \frac{1}{2}\operatorname{tr}((\mathbf{S}_t^{(N)})^{-1}\mathbf{S}_t^{(\lambda_t)}) - 1,$$

where $\mathrm{tr}(\cdot)$ is the trace operator. Let $\lambda_t = \lambda_t^*$ be the solution of $D(\mathbf{S}_t^{(\lambda_t)}||\mathbf{S}_t^{(M)}) = D(\mathbf{S}_t^{(\lambda_t)}|| \mathbf{S}_t^{(N)})$. The Chernoff information $CI(\mathbf{S}_t^{(M)}||\mathbf{S}_t^{(N)})$ is defined as

$$CI(\mathbf{S}_t^{(M)}||\mathbf{S}_t^{(N)}) = D(\mathbf{S}_t^{(\lambda_t^*)}||\mathbf{S}_t^{(M)}) = D(\mathbf{S}_t^{(\lambda_t^*)}||\mathbf{S}_t^{(N)}).$$

An expression for individual λ_t^* is derived in Appendix A.

Given λ_t^*, CI of all pairs $(\mathbf{S}_t^{(M)}, \mathbf{S}_t^{(N)})$ can be evaluated, leading to a ranking (greater CI corresponds to higher rank) of the most different pairs of diagonal elements $(c_t^{(M)}, c_t^{(N)})$ and, consequently, the most discriminative canonical vectors of $\mathbf{U}$. Let the k most discriminative vectors serve as the columns of the new canonical vector matrix $\tilde{\boldsymbol{U}}$. Moreover, select the top k different pairs of diagonal elements $(\tilde{c}_i^{(M)}, \tilde{c}_i^{(N)})$ and the corresponding $(\tilde{\boldsymbol{S}}_i^{(M)}, \tilde{\boldsymbol{S}}_i^{(N)})$, where $i \in \{1, 2, \ldots, k\}$.

2.4. Similarity Measurement

The selected canonical vectors can be leveraged to measure the similarity between an arbitrary pair of images through a binary hypothesis test. Let $(\mathbf{x}_r, \mathbf{y}_c)$ be an arbitrary pair of global feature vectors. The exact joint distribution of $(\mathbf{x}_r, \mathbf{y}_c)$ likely varies from one dataset to another and does not admit an explicit characterization. Here we make the simplifying assumption that $\mathbf{x}_r$ and $\mathbf{y}_c$ are jointly Gaussian. Specifically, we assume that $(\mathbf{x}_r, \mathbf{y}_c) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Phi}^{(M)})$ if they come from two matching images, and $(\mathbf{x}_r, \mathbf{y}_c) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Phi}^{(N)})$ otherwise, where $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ denotes a multivariate Gaussian distribution [34] with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma}$. Given $(\mathbf{x}_r, \mathbf{y}_c)$, the transformed feature vectors are computed as follows:

$$\mathbf{w} = [w_1, w_2, \ldots, w_k]^T = \tilde{\boldsymbol{U}}^T \boldsymbol{\Sigma}_{auto}^{-\frac{1}{2}} \mathbf{x}_r,$$
$$\mathbf{v} = [v_1, v_2, \ldots, v_k]^T = \tilde{\boldsymbol{U}}^T \boldsymbol{\Sigma}_{auto}^{-\frac{1}{2}} \mathbf{y}_c.$$

Since $\boldsymbol{\Lambda}$ is a diagonal matrix, it follows that $(w_1, v_1), (w_2, v_2), \ldots, (w_k, v_k)$ are mutually independent with $(w_i, v_i) \sim \mathcal{N}(\mathbf{0}, \tilde{\boldsymbol{S}}_i^{(M)})$ for $i \in \{1, 2, \ldots, k\}$ in the case where $(\mathbf{x}_r, \mathbf{y}_c)$ is a matching pair. We shall further assume that $\boldsymbol{\Pi}$ is also a diagonal matrix, which is justified by the fact that in practice $\boldsymbol{\Pi}$ is often very close to a zero matrix (see Figure 3 and 4 for some empirical evidences). As a consequence, $(w_1, v_1), (w_2, v_2), \ldots, (w_k, v_k)$ are mutually independent with $(w_i, v_i) \sim \mathcal{N}(\mathbf{0}, \tilde{\boldsymbol{S}}_i^{(N)})$ for $i \in \{1, 2, \ldots, k\}$ in the case where $(\mathbf{x}_r, \mathbf{y}_c)$ is a non-matching pair. To check whether the given two images match or not, one can perform a binary hypothesis test regarding the underlying distribution of $(\mathbf{w}, \mathbf{v})$: $\otimes_{i=1}^k \mathcal{N}(\mathbf{0}, \tilde{\boldsymbol{S}}_i^{(M)})$ vs. $\otimes_{i=1}^k \mathcal{N}(\mathbf{0}, \tilde{\boldsymbol{S}}_i^{(N)})$.

Note that $\mathcal{N}(\mathbf{0}, \tilde{\boldsymbol{S}}_i^{(M)})$ has probability density

$$P_M(w_i, v_i) = \frac{e^{-\frac{1}{2}[w_i \quad v_i] \begin{bmatrix} 1 & \tilde{c}_i^{(M)} \\ \tilde{c}_i^{(M)} & 1 \end{bmatrix}^{-1} \begin{bmatrix} w_i \\ v_i \end{bmatrix}}}{\sqrt{(2\pi)^2 \begin{vmatrix} 1 & \tilde{c}_i^{(M)} \\ \tilde{c}_i^{(M)} & 1 \end{vmatrix}}} \tag{5}$$

while $\mathcal{N}(\mathbf{0}, \tilde{\boldsymbol{S}}_i^{(N)})$ has probability density

$$P_N(w_i, v_i) = \frac{e^{-\frac{1}{2}\begin{bmatrix} w_i & v_i \end{bmatrix}\begin{bmatrix} 1 & \tilde{c}_i^{(N)} \\ \tilde{c}_i^{(N)} & 1 \end{bmatrix}^{-1}\begin{bmatrix} w_i \\ v_i \end{bmatrix}}}{\sqrt{(2\pi)^2\begin{vmatrix} 1 & \tilde{c}_i^{(N)} \\ \tilde{c}_i^{(N)} & 1 \end{vmatrix}}}. \tag{6}$$

We are now in a position to conduct a binary hypothesis test based on the confidence score given below:

$$score_G = \log \frac{\otimes_{i=1}^{n} P_M(w_i, v_i)}{\otimes_{i=1}^{n} P_N(w_i, v_i)} = \sum_{i=1}^{k} \log \frac{P_M(w_i, v_i)}{P_N(w_i, v_i)}. \tag{7}$$

Substituting Equations (5) and (6) into Equation (7) gives

$$\begin{aligned} score_G &= \sum_{i=1}^{k} (\log P_M(w_i, v_i) - \log P_N(w_i, v_i)) \\ &= \sum_{i=1}^{k} \left(-\frac{w_i^2 - 2w_i v_i \tilde{c}_i^{(M)} + v_i^2}{2\pi\sqrt{(1 - (\tilde{c}_i^{(M)})^2)}} + \frac{w_i^2 - 2w_i v_i \tilde{c}_i^{(N)} + v_i^2}{2\pi\sqrt{(1 - (\tilde{c}_i^{(N)})^2)}} + \log \frac{\sqrt{1 - (\tilde{c}_i^{(N)})^2}}{\sqrt{1 - (\tilde{c}_i^{(M)})^2}} \right), \end{aligned}$$

which is equivalent to

$$\sum_{i=1}^{k} \left(-\frac{w_i^2 - 2w_i v_i \tilde{c}_i^{(M)} + v_i^2}{\sqrt{(1 - (\tilde{c}_i^{(M)})^2)}} + \frac{w_i^2 - 2w_i v_i \tilde{c}_i^{(N)} + v_i^2}{\sqrt{(1 - (\tilde{c}_i^{(N)})^2)}} \right) \tag{8}$$

as the log term and the scalar 2π have no effect on rankings. This confidence score reflects the degree of similarity between the two given images. The higher the score is, the more likely the images match each other.

3. Experimental Results

3.1. Training Datasets

We resort to two datasets for training, namely, 120k-Structure from Motion (120k-SfM) and 30k-Structure from Motion (30k-SfM) [35]. Both are preprocessed to eliminate overlaps with the evaluation datasets. A succinct description of these two datasets can be found below:

3.1.1. 120k-Structure from Motion

120k-Structure from Motion (120k-SfM) dataset is constructed from the one used in the work of Schonberger et al. [36], which contains 713 3D models with nearly 120k images. The maximum size of each image is 1024 × 1024. The original dataset includes all image from Oxford5k and Paris6k. Those images are removed to avoid overlaps (in total 98 clusters are eliminated).

3.1.2. 30k-Structure from Motion

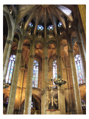
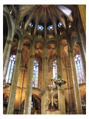
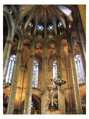
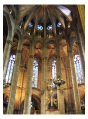
30k-Structure from Motion (30k-SfM) dataset is a subset of 120k-SfM, which contains approximately 30k images and 551 classes. The maximum size of images are resized to 362 × 362.

Each dataset serves its own purpose; 30k-SfM is a small dataset while 120k-SfM is a big one. This enables us to investigate the pros and cons of different datasets in terms of their sizes. Compared to 30k-SfM, 120k-SfM supplies richer features to be explored by the methods being tested.

3.2. Training Details

Using each dataset, two lists of matching and non-matching pairs of images are created for training—feature space analysis not CNN training. Table 1 shows some examples of matching and non-matching pairs. Specifically, we randomly select 10,960 raw pairs from 30k-SfM and 58,502 raw pairs from 120k-SfM. We double the number matching and non-matching pairs by simultaneously using each raw pair and its flipped version to ensure that the diagonal/off-diagonal blocks of the data covariances in Equations (1) and (2) are identical and symmetric. This could also be seen from Table 1: each pair is used twice but with its image order flipped.

Table 1. Examples of matching/non-matching pairs.

Matching Pair						
						
Non-Matching Pair						
						

The feature vector of a given image is extracted from the very last pooling layer of a pre-trained VGG16 via one of the following three pooling strategies: Global Max (MAC) pooling, Global Average (AVE) pooling, and global Standard Deviation (SD) pooling (global Max (MAC) pooling, Global Average (AVE) pooling, and global Standard Deviation (SD) pooling compute, respectively, the maximum value, the average value, and the standard deviation of the feature map in each channel). We conducted separate training for each of these strategies in order to compare performances.

For benchmarking, the proposed method (G-CCA) and its variant (S-CCA) were trained along with three alternative feature-space analysis methods, i.e., PCAw [21], Supervised PCA (SPCA) [37] and Multiclass LDA (MLDA) [38]. G-CCA is depicted in Figure 2 while S-CCA is the same as G-CCA except that in the final step the scalar similarity measure is used instead (namely, in the last block of Figure 2, $score_G$ is replaced with $score_S = \mathbf{w}^T \cdot \mathbf{v}$). PCAw infers a basis matrix of the feature space from the covariance matrix of the training image features. This basis matrix is used to whiten and compress new image features, which are then leveraged to make a matching/non-matching decision based on the scalar similarity measure. See [12] for a detailed description of the PCAw method and its performance. Furthermore, we compared the proposed method with SPCA, which is a weighted PCA method. It uses a Laplacian matrix to characterize the relationship among the classes in the dataset. We implement SPCA by following the steps in [39]. As to LDA [40], its application to image retrieval has also been thoroughly investigated [41], which is hardly surprising given its popularity in statistical analysis. Here we use its variant MLDA [38] as a competing feature-space analysis method. MLDA is trained using the classes provided by both training datasets. It derives a set of projection vectors that offer the best linear separation of the classes (full separation is achievable if the classes are linearly separable, otherwise, MLDA produces some overlaps between the classes). These projection vectors are employed to transform and compress (in the sense of dimensionality reduction) new feature vectors. Scalar similarity is then evaluated for the transformed features to determine whether or not they match.

3.3. Implementation Details

In the experiment, we compare G-CCA, S-CCA with PCAw, SPCA, and MLDA. The G-CCA and S-CCA are presented in this paper while the PCAw, SPCA, and MLDA are implemented by following procedures in [21,38,39]. Here, we discuss some detailed issues in the implementation.

Firstly, S-CCA, PCAw, SPCA, and MLDA use scalar similarity score to calculate the confidence score while G-CCA uses the proposed score in Equation (8). Secondly, for all these methods, the feature vectors are obtained via MAC, AVE, and SD pooling, and centralization and normalization are performed. Thirdly, the performance comparisons are conducted for eight dimensions: 512, 450, 400, 300, 200, 100, 50, and 25. Lastly, we calculate the scores between the query image and each image in the test dataset, and obtain the image retrieval results by ranking scores from high to low. All the methods are evaluated by the mean Average Precision (mAP) (we calculatethe mAP without enforcing the monotonicity for Precision (Recall) relationship). which can be formulated as follows:

$$\mathrm{mAP} = \frac{\sum_{i=1}^{m} \mathrm{AP_i}}{m} \qquad \text{with } \mathrm{AP_i} = \sum_{k=1}^{n} P(k)\Delta r(k),$$

where $\mathrm{AP_i}$ is the average precision for the i-th query image, m is the total number of query images, and n is the total number of images in the testing dataset, $P(k)$ is the precision of top k results, and $\Delta r(k) = R(k) - R(k-1)$ with $R(k)$ being the recall of top k results. For calculating the precision $P(K)$ and recall $R(k)$, the positive labels for each query image are provided by the test datasets.

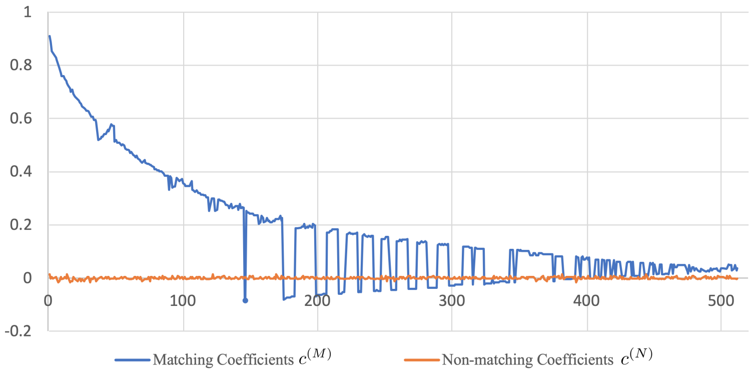


Figure 3. Profile of the diagonal elements of $\mathbf{\Lambda}$ and $\mathbf{\Pi}$ (i.e., $c_t^{(M)}$ and $c_t^{(N)}$, where $t \in \{1, 2, \ldots, 512\}$) using AVE features. The CCA training was performed on the 120k-SfM dataset.

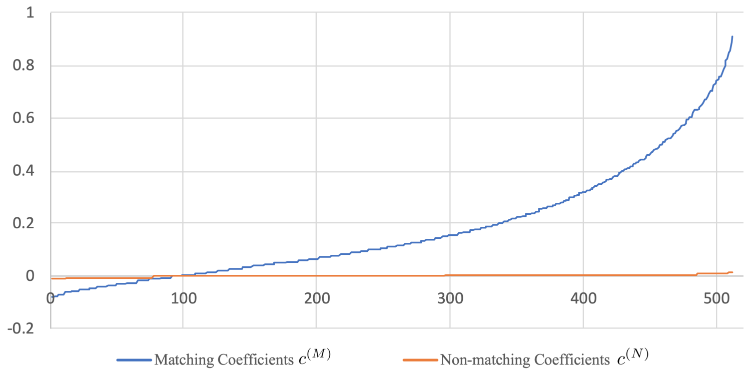


Figure 4. Profile of sorted diagonal elements of $\mathbf{\Lambda}$ and $\mathbf{\Pi}$ (i.e., $\tilde{c}_t^{(M)}$ and $\tilde{c}_t^{(N)}$, where $t \in \{1, 2, \ldots, 512\}$) using AVE features. The CCA training was performed on the 120k-SfM dataset.

3.4. Evaluation Datasets and Details

Four datasets, namely, Oxford5k [42], Paris6k [43], $\mathcal{R}$Oxford [44], and $\mathcal{R}$Paris [44], are used to assess the performance of each retrieval method. As the first two datasets are contained in the large raw 120k-SfM dataset, they are excluded from the training dataset via preprocessing. The last two datasets contain new annotations and more difficult query images, and consequently create more challenges for image retrieval; therefore, they can help test the reliability of our approach. A short description of each dataset is given below.

3.4.1. Oxford5k

Oxford5k dataset contains 5063 images and 55 query images for 11 different buildings. It is annotated with bounding boxes for the main objects.

3.4.2. $\mathcal{R}$Oxford

$\mathcal{R}$Oxford dataset contains 4993 images and 70 query images for 11 different buildings. Query images are excluded from the retrieval images. Same as Oxford5k, it is annotated with bounding boxes for the main objects.

3.4.3. Paris6k

Paris6k dataset contains 6412 images and 55 query images for 11 different buildings. It is also annotated with bounding boxes.

3.4.4. $\mathcal{R}$Paris

$\mathcal{R}$Paris dataset contains 6322 images and 70 query images for 11 different buildings. Query images are excluded from the retrieval images. Same as Paris6k, it is annotated with bounding boxes.

The performance of each retrieval method is evaluated using mean Average Precision (mAP) [42]. The positive labels of each query image are provided by the datasets. The standard evaluation protocol is followed for Oxford5k and Paris6k. As for the $\mathcal{R}$Oxford and $\mathcal{R}$Paris datasets, the medium protocol setups in [44] are adopted. We crop all the query images with the provided bounding boxes before feeding them to VGG16. Each method undergoes training and evaluation twice. The first training used the small dataset, 30k-SfM, followed by evaluation. Then it was trained with the large dataset, 120k-SfM, before evaluation. This enables us to study the effect of dataset size and diversity on the methods under comparison.

3.5. Performance Evaluation and Analysis

Before getting into the performance evaluation of the proposed method, it is useful to have some insights about how discriminative the canonical vectors are. Figures 5 and 6 show the profile of the diagonal elements of the off-diagonal blocks in Equations (3) and (4). It can be seen that the values of $c_t^{(N)}$ fluctuate around zero whereas those of $c_t^{(M)}$ range between -0.1 and 0.9. This observation suggests that there exists a set of canonical vectors that can effectively tell apart matching from non-matching pairs of images. This is shown in the rest of this subsection.

Table 2 reports the baseline performances of MAC, AVE, and SD without dimensionality reduction. Specifically, for these baselines, we directly calculate the scalar similarity between the pooling features (after centralization and normalization) of the query image and each image in the testing dataset. In the evaluation, we consider the proposed method (G-CCA) and its variant with Gaussian-distribution-based hypothesis testing replaced by scalar similarity (S-CCA). From Table 2, we observe that G-CCA achieves better performance than S-CCA in most cases except for Paris6k and AVE on $\mathcal{R}$Paris.

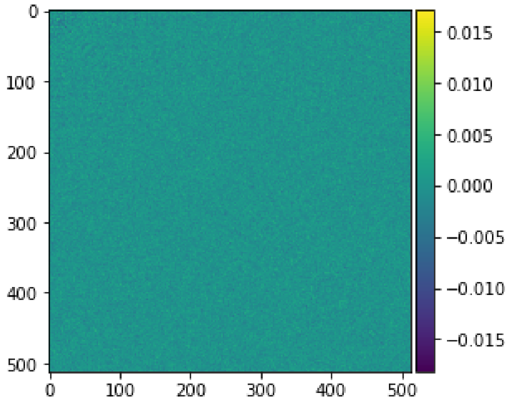


Figure 5. A 2D visualization of matrix **Π**.

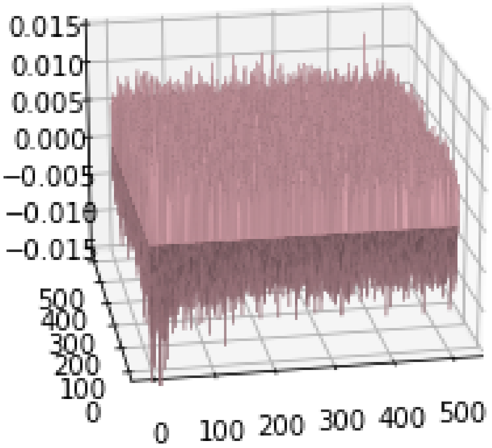


Figure 6. A 3D visualization of matrix **Π**.

Table 2. Performance comparison of the baseline, S-CCA, and G-CCA on Oxford5k, $\mathcal{R}$Oxford, Paris6k, and $\mathcal{R}$Paris without dimension reduction.

Method	Oxford5k	$\mathcal{R}$Oxford	Paris6k	$\mathcal{R}$Paris
MAC	0.5296	0.3295	0.7455	0.5122
S-CCA + MAC	0.5800	0.3575	**0.7726**	0.5408
G-CCA + MAC	**0.6275**	**0.3996**	0.7455	**0.5939**
AVE	0.5312	0.2884	0.6467	0.4653
S-CCA + AVE	0.6845	0.4303	**0.7845**	**0.5936**
G-CCA + AVE	**0.7146**	**0.4444**	0.7507	0.5812
SD	0.6095	0.3834	0.7355	0.5311
S-CCA + SD	0.6943	0.4503	**0.8191**	0.6199
G-CCA + SD	**0.7419**	**0.4806**	0.8164	**0.6403**

1. The evaluation results are based on 120k-SfM. 2. For the same type of features, the best performances are highlighted in **bold**.

By considering three different pooling strategies, three image retrieval methods are trained on the 30k-SfM dataset and evaluated on all four test sets. Table 3 provides a comprehensive depiction of the experimental results for each retrieval method with different pooling strategies and feature dimensionality choices (compression levels). The results for MLDA are not reported there, for MLDA cannot be trained on the 30k-SfM dataset, which is a consequence of the fact that the difference between classes is too small as far as MLDA training is concerned. From Table 3, four observations can be made. The first is regarding the effect of the pooling strategy. Specifically, SD pooling appears to result in the most competitive performance for all methods at every choice of feature dimensionality while MAC renders G-CCA superior to SPCA and PCAw at low dimensions over all test sets. The second observation is that for MAC, AVE, and SD pooling strategies, the proposed method outperforms PCAw at low feature dimensionality. As such, the proposed method is a better choice for producing compact features than PCAw regardless of the pooling strategy. The last observation is that G-CCA is more robust against dimensionality reduction than S-CCA.

The performance of the proposed method can be improved by replacing 30k-SfM with 120k-SfM, which is a larger training set. Table 4 shows the corresponding evaluation results for all the methods with different pooling strategies and dimensionality choices (the only exception is SPCA for which the training on 120k-SfM is computationally infeasible as its Laplacian matrix is too large to be stored on our computer). It is clear that the increased-size training set leads to an improved mAP performance on all test sets and for all pooling strategies. It is also interesting to note that the proposed method outperforms all others on Oxford5k. This uniform superiority across all dimensions is only attained on Paris6k using SD pooling. Although AVE and MAC improve mAP, they cause G-CCA to lose its edge at high dimensions on Paris6k. In contrast, with SD pooling, the proposed method retains its dominating performance at all feature dimensions. On $\mathcal{R}$Oxford and $\mathcal{R}$Paris, the performance of G-CCA is better than MLDA at almost all dimensions with MAC. G-CCA almost outperforms PCAw in every dimensions with all three pooling strategies.

Based on Tables 3 and 4, there are three notable advantages of G-CCA over MLDA, PCAw, and SPCA. The first is that the CCA-based methods can be trained using datasets with small differences between classes whereas MLDA cannot be trained on such datasets. The second advantage is that G-CCA typically shows a more graceful performance degradation than PCAw after dimensionality reduction. The last is that SPCA can not be trained on large datasets as compared with G-CCA.

Tables 5–7 present some retrieval results for visual illustration. In Table 5, a query image from the Oxford5k set is presented to PCAw, SPCA, and G-CCA, trained on the 30k-SfM set, while in Tables 6 and 7, a query image from the Oxford5k set is presented to PCAw, MLDA, and G-CCA, trained on the 120k-SfM set. We list top 10 matches for each method with each list ranked using the matching score associated with the corresponding method. Tables 5 and 6 show the top 10 retrieved images for different methods with SD pooling while Table 7 gives examples for G-CCA with different pooling strategies.

Table 3. Evaluation results from 30k-SfM on Oxford5k, $\mathcal{R}$Oxford, Paris6k, and $\mathcal{R}$Paris.

	Dim	MAC				AVE				SD			
		SPCA	PCAw	S-CCA	G-CCA	SPCA	PCAw	S-CCA	G-CCA	SPCA	PCAw	S-CCA	G-CCA
Oxford5k	25	0.3589	0.3555	0.2431	**0.3873**	0.4474	0.4443	0.3091	**0.4873**	0.4757	0.4838	0.3439	**0.4979**
	50	0.4412	0.4258	0.3174	**0.4487**	0.4930	0.4933	0.3782	**0.5127**	0.5086	0.5074	0.4403	**0.5690**
	100	0.5016	0.5027	0.4122	**0.5043**	0.5599	0.5697	0.5447	**0.6034**	0.6002	0.6041	0.5191	**0.6164**
	200	**0.5628**	0.5583	0.4818	0.5501	0.6083	0.6086	0.6157	**0.6445**	0.6635	0.6619	0.6280	**0.6772**
	300	**0.5723**	0.5672	0.5280	0.5379	0.6416	0.6307	0.6428	**0.6552**	0.6753	0.6736	0.6513	**0.6830**
	400	**0.5728**	0.5715	0.5505	0.5405	0.6517	0.6385	0.6373	**0.6525**	**0.6811**	**0.6811**	0.6703	0.6745
	450	**0.5670**	0.5654	0.5609	0.5364	**0.6544**	0.6422	0.6393	0.6538	0.6839	**0.6849**	0.6740	0.6746
	512	**0.5615**	0.5601	0.5580	0.5363	0.6506	0.6388	0.6493	**0.6537**	**0.6766**	0.6763	0.6764	0.6743
	Dim	MAC				AVE				SD			
		SPCA	PCAw	S-CCA	G-CCA	SPCA	PCAw	S-CCA	G-CCA	SPCA	PCAw	S-CCA	G-CCA
$\mathcal{R}$Oxford	25	0.2070	0.2226	0.1495	**0.2276**	0.2702	**0.2709**	0.1939	0.2590	0.2883	0.2856	0.2116	**0.3031**
	50	0.2823	0.2771	0.1886	**0.2914**	0.2731	0.2757	0.2206	**0.2876**	0.3117	0.3123	0.2590	**0.3485**
	100	0.3259	0.3281	0.2484	**0.3282**	0.3304	0.3197	0.3083	**0.3372**	**0.3885**	0.3795	0.3007	0.3848
	200	0.3462	0.3545	0.3071	**0.3569**	0.3569	0.3531	0.3759	**0.4002**	0.4399	0.4368	0.4021	**0.4417**
	300	**0.3595**	0.3593	0.3290	0.3413	0.3901	0.3771	0.3911	**0.4057**	**0.4507**	0.4420	0.4173	0.4484
	400	**0.3576**	0.3568	0.3424	0.3400	0.3905	0.3796	0.3798	**0.4065**	0.4526	0.4381	0.4454	**0.4538**
	450	**0.3551**	0.3544	0.3466	0.3398	0.4002	0.3772	0.3876	**0.4052**	0.4498	0.4382	0.4435	**0.4499**
	512	0.3442	**0.3469**	0.3444	0.3396	0.4042	0.3767	0.3963	**0.4077**	0.4417	0.4383	0.4412	**0.4419**
	Dim	MAC				AVE				SD			
		SPCA	PCAw	S-CCA	G-CCA	SPCA	PCAw	S-CCA	G-CCA	SPCA	PCAw	S-CCA	G-CCA
Paris	25	0.4878	0.5084	0.4133	**0.5464**	0.4944	0.4330	0.4182	**0.4990**	0.5633	0.5858	0.4758	**0.5969**
	50	0.6027	0.6208	0.5391	**0.6347**	0.5692	0.5893	0.5898	**0.6153**	0.6415	0.6555	0.6084	**0.6746**
	100	0.6691	0.6750	0.5848	**0.6808**	0.6441	0.6736	0.6559	**0.6790**	0.7290	0.7267	0.6988	**0.7426**
	200	0.7035	0.6942	0.6384	**0.7166**	0.6931	0.6994	0.7049	**0.7106**	0.7719	0.7620	0.7501	**0.7811**
	300	0.7004	0.6980	0.6701	**0.7067**	0.7109	**0.7328**	0.7297	0.7118	0.7834	0.7819	0.7739	**0.7892**
	400	**0.7076**	0.7057	0.6893	0.7052	0.7375	**0.7586**	0.7418	0.7120	**0.8010**	0.7970	0.7885	0.7867
	450	**0.7091**	0.7073	0.6964	0.7027	0.7482	**0.7679**	0.7472	0.7130	**0.8067**	0.8066	0.7969	0.7871
	512	**0.7032**	0.7060	0.7039	0.7029	0.7508	**0.7732**	0.7520	0.7133	0.8020	0.8031	**0.8036**	0.7874

Table 3. *Cont.*

	Dim	MAC				AVE				SD			
		SPCA	PCAw	S-CCA	G-CCA	SPCA	PCAw	S-CCA	G-CCA	SPCA	PCAw	S-CCA	G-CCA
$\mathcal{R}$Paris	25	0.3966	0.3944	0.3225	**0.4212**	0.3877	0.3981	0.3085	**0.4212**	0.4361	0.4410	0.3602	**0.4433**
	50	0.4725	0.4738	0.4063	**0.4781**	0.4354	0.4442	0.4385	**0.4538**	0.5006	0.5015	0.4524	**0.5056**
	100	0.5007	0.5021	0.4311	**0.5106**	0.4820	0.4886	0.4946	**0.5082**	0.5457	0.5501	0.5258	**0.5653**
	200	0.5183	0.5182	0.4668	**0.5370**	0.5118	0.5129	0.5302	**0.5355**	0.5822	0.5827	0.5635	**0.5985**
	300	0.5206	0.5200	0.4894	**0.5285**	0.5281	0.5306	**0.5507**	0.5377	0.5966	0.5964	0.5829	**0.6045**
	400	0.5224	0.5219	0.5040	**0.5272**	0.5504	0.5507	**0.5577**	0.5379	0.6064	**0.6070**	0.5958	0.6024
	450	0.5222	0.5200	0.5109	**0.5255**	0.5587	0.5590	**0.5620**	0.5383	0.6119	**0.6121**	0.6013	0.6027
	512	0.5169	0.5168	0.5154	**0.5256**	0.5579	0.5588	**0.5646**	0.5384	0.6051	**0.6067**	0.6048	0.6028

1. The best performances in each dimension are highlighted in **bold**.

Table 4. Evaluation results from 120k-SfM on Oxford5k, $\mathcal{R}$Oxford, Paris6k, and $\mathcal{R}$Paris.

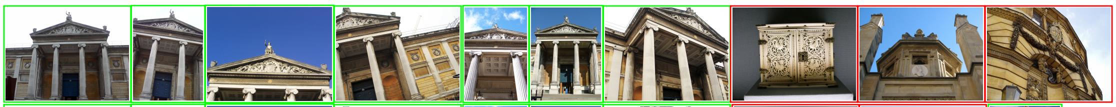
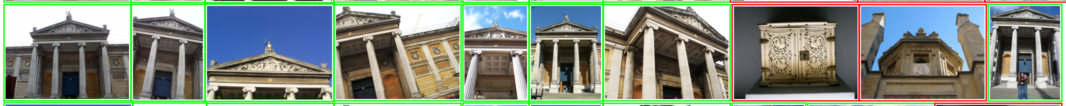
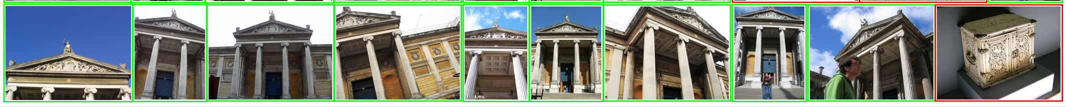
	Dim	MAC				AVE				SD			
		MLDA	PCAw	S-CCA	G-CCA	MLDA	PCAw	S-CCA	G-CCA	MLDA	PCAw	S-CCA	G-CCA
Oxford5k	25	0.3603	0.3906	0.2677	**0.4019**	0.4758	0.4266	0.2644	**0.4821**	0.4759	0.4790	0.3400	**0.5212**
	50	0.4760	0.4319	0.3802	**0.4987**	**0.5612**	0.5033	0.4293	0.5572	0.5375	0.5355	0.4667	**0.5956**
	100	0.5157	0.5275	0.4537	**0.5481**	0.6017	0.5756	0.5529	**0.6402**	0.6429	0.6240	0.5593	**0.6688**
	200	0.5887	0.5453	0.5562	**0.6231**	0.6571	0.6437	0.6498	**0.6964**	0.6861	0.6410	0.6620	**0.7244**
	300	0.6028	0.5669	0.5697	**0.6306**	0.6643	0.6474	0.6658	**0.7102**	0.7030	0.6711	0.6754	**0.7382**
	400	0.5974	0.5810	0.5768	**0.6275**	0.6688	0.6681	0.6758	**0.7139**	0.7020	0.6970	0.6864	**0.7422**
	450	0.5939	0.5840	0.5820	**0.6279**	0.6678	0.6728	0.6781	**0.7144**	0.6972	0.6986	0.6939	**0.7412**
	512	0.5868	0.5799	0.5800	**0.6275**	0.6613	0.6711	0.6845	**0.7146**	0.6958	0.6946	0.6943	**0.7419**

Table 4. *Cont.*

	Dim	MAC				AVE				SD			
		MLDA	PCAw	S-CCA	G-CCA	MLDA	PCAw	S-CCA	G-CCA	MLDA	PCAw	S-CCA	G-CCA
$\mathcal{R}$Oxford	25	0.2330	**0.2503**	0.1543	0.2459	**0.2712**	0.2533	0.1422	0.2441	0.2666	**0.3037**	0.1782	0.2853
	50	0.2989	0.2664	0.2366	**0.3025**	**0.3522**	0.2802	0.2337	0.3366	**0.3418**	0.3357	0.2636	0.3254
	100	**0.3470**	0.3521	0.2724	0.3437	0.3981	0.3318	0.3412	**0.4290**	0.4002	0.4075	0.3269	**0.4073**
	200	0.3924	0.3510	0.3482	**0.3991**	0.4324	0.3911	0.3913	**0.4411**	0.4497	0.4192	0.4085	**0.4622**
	300	**0.4006**	0.3557	0.3497	0.3986	0.4404	0.3920	0.4056	**0.4430**	0.4645	0.4454	0.4335	**0.4796**
	400	0.3964	0.3625	0.3526	**0.4001**	0.4412	0.4106	0.4215	**0.4462**	0.4673	0.4609	0.4429	**0.4812**
	450	0.3941	0.3613	0.3587	**0.3998**	0.4363	0.4159	0.4215	**0.4443**	0.4624	0.4604	0.4394	**0.4807**
	512	0.3881	0.3570	0.3575	**0.3996**	0.4267	0.4136	0.4303	**0.4444**	0.4597	0.4501	0.4503	**0.4806**
	Dim	MAC				AVE				SD			
		MLDA	PCAw	S-CCA	G-CCA	MLDA	PCAw	S-CCA	G-CCA	MLDA	PCAw	S-CCA	G-CCA
Paris6k	25	0.5781	0.4878	0.5109	**0.6270**	0.5553	0.5013	0.4442	**0.5693**	0.6204	0.5543	0.5269	**0.6611**
	50	0.6384	0.6153	0.5416	**0.6679**	0.6362	0.5893	0.5467	**0.6314**	0.6900	0.6575	0.5935	**0.6968**
	100	0.6916	0.6788	0.6226	**0.7339**	0.6994	0.6736	0.6657	**0.6910**	0.7502	0.7313	0.7105	**0.7641**
	200	0.7244	0.7124	0.6765	**0.7674**	0.7162	0.6994	0.7220	**0.7491**	0.7845	0.7842	0.7760	**0.8043**
	300	0.7493	0.7214	0.6900	**0.7719**	0.7299	0.7328	**0.7538**	0.7491	0.8030	00.8046	0.7973	**0.8160**
	400	0.7548	0.7230	0.7146	**0.7729**	0.7247	0.7586	**0.7729**	0.7507	0.8042	0.8143	0.8067	**0.8164**
	450	0.7540	0.7222	**0.7729**	0.7455	0.7197	0.7679	**0.7775**	0.7508	0.8003	0.8144	0.8096	**0.8161**
	512	0.7549	0.7288	**0.7726**	0.7455	0.7111	0.7732	**0.7845**	0.7507	0.7971	0.8159	0.8164	**0.8191**
	Dim	MAC				AVE				SD			
		MLDA	PCAw	S-CCA	G-CCA	MLDA	PCAw	S-CCA	G-CCA	MLDA	PCAw	S-CCA	G-CCA
$\mathcal{R}$Paris	25	0.4321	0.3728	0.3956	**0.4787**	0.4524	0.3745	0.3607	**0.4455**	0.4817	0.4136	0.4075	**0.5032**
	50	0.4910	0.4685	0.4214	**0.5156**	**0.4944**	0.4495	0.4229	0.4877	0.5373	0.4998	0.4611	**0.5415**
	100	0.5339	0.5096	0.4681	**0.5631**	0.5003	0.5101	0.5052	**0.5340**	0.5796	0.5596	0.5472	**0.5970**
	200	0.5526	0.5346	0.5066	**0.5910**	0.5656	0.5310	0.5437	**0.5678**	0.6066	0.6002	0.5928	**0.6317**
	300	0.5520	0.5425	0.5124	**0.5942**	0.5809	0.5566	0.5639	**0.5799**	0.6195	0.6156	0.6021	**0.6408**
	400	0.5437	0.5406	0.5282	**0.5941**	0.5843	0.5738	0.5824	**0.5857**	0.6165	0.6228	0.6072	**0.6401**
	450	0.5399	0.5369	0.5313	**0.5941**	0.5829	0.5796	**0.5844**	0.5813	0.6136	0.6187	0.6118	**0.6401**
	512	0.5333	0.5387	0.5408	**0.5939**	0.5830	0.5828	**0.5936**	0.5812	0.6093	0.6178	0.6199	**0.6403**

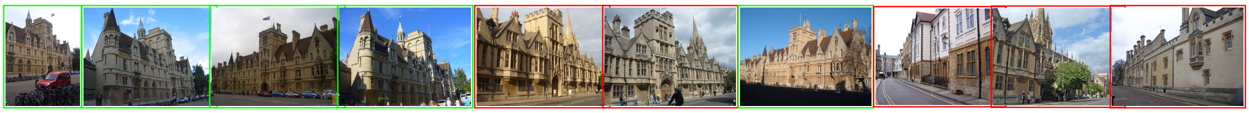
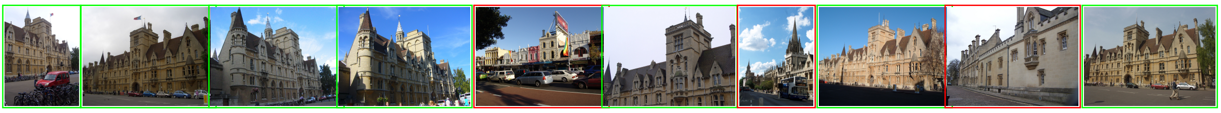
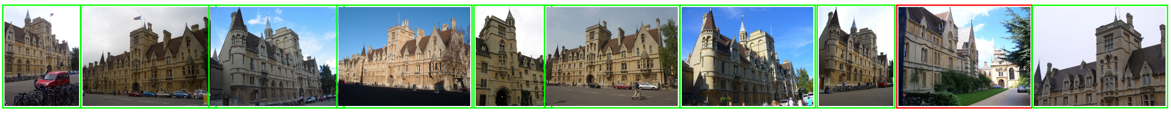
1. The best performances in each dimension are highlighted in **bold**.

Table 5. Image retrieval comparison of PCAw, SPCA, and G-CCA.

	Query	TOP 10 Retrieved Images
A		
B		
C		

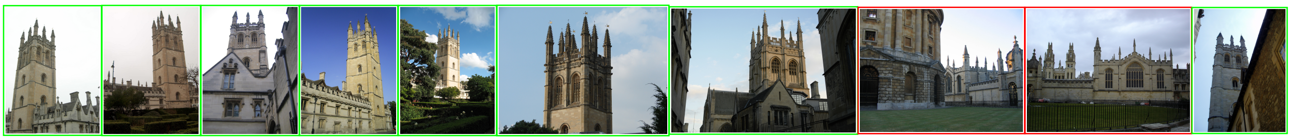
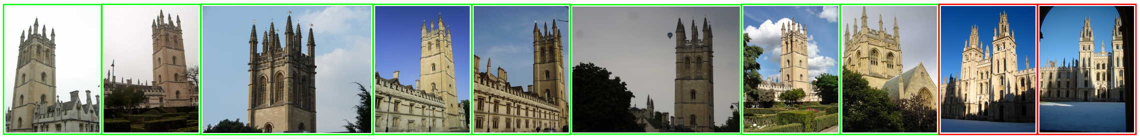
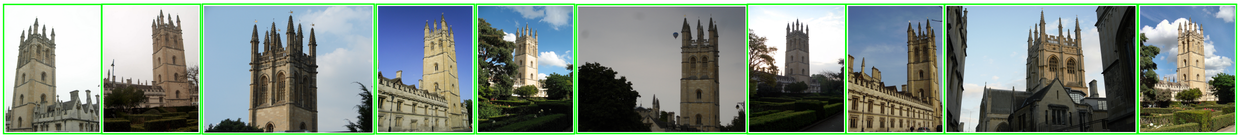
1. Top 10 retrieved images from Oxford5k. (A) SD + PCAw. (B) SD + SPCA (C) SD + G-CCA. 2. Correct images are bounded with green boxes, wrong images are bounded with red boxes.

Table 6. Image retrieval comparison of PCAw, MLDA, and G-CCA.

	Query	TOP 10 Retrieved Images
A		
B		
C		

1. Top 10 retrieved images from Oxford5k. (A) SD + PCAw. (B) SD + MLDA (C) SD + G-CCA. 2. Correct images are bounded with green boxes, wrong images are bounded with red boxes.

Table 7. Image retrieval comparison of G-CCA with MAC, AVE, and SD feature.

	Query	TOP 10 Retrieval Images
A		
B		
C		

1. top 10 retrieved images from Oxford5k. (A) MAC + G-CCA. (B) AVE + G-CCA (C) SD + G-CCA. 2. Correct images are bounded with green box, wrong images are bounded with red box.

4. Conclusions

In view of the success of DL in image classification, a CCA-based method is proposed to exploit DL features for image retrieval applications. By adopting an OTS CNN without fine-tuning, it achieves good retrieval accuracy with a minimal computational overhead. As shown by the experimental results on standard evaluation datasets, the proposed method is performance-wise competitive against traditional and other OTS-CNN-based methods. Moreover, it exhibits improved robustness against dimensionality reduction and enhanced sensitivity to feature mismatch.

Author Contributions: Data curation, K.S.; Methodology, J.L. and J.C.; Software, K.S., X.L., M.A., X.G. and H.L.; Supervision, J.C.; Writing–original draft, K.S.; Writing–review and editing, M.A., J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada through a Discovery Grant.

Data Availability Statement: The source code is available at https://github.com/ShiKangdi/Image-Retrieval-via-Canonical-Correlation-Analysis-and-Binary-Hypothesis-Testing.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Chernoff Information between Two 2-Dimensional Gaussian Distributions

For notational simplicity, we suppress subscript t in the following derivation. Consider

$$\mathbf{S}^{(M)} = \begin{bmatrix} 1 & c^{(M)} \\ c^{(M)} & 1 \end{bmatrix}, \quad \mathbf{S}^{(N)} = \begin{bmatrix} 1 & c^{(N)} \\ c^{(N)} & 1 \end{bmatrix},$$

where $c^{(M)}$ and $c^{(N)}$ are two corresponding coefficients. Let $\mathbf{S}^{(\lambda)} = (\lambda(\mathbf{S}^{(M)})^{-1} + (1-\lambda)(\mathbf{S}^{(N)})^{-1})^{-1}$, $\lambda \in [0,1]$. Now we proceed to find the solution $\lambda = \lambda^*$ of the equation $D(\mathbf{S}^{(\lambda)}||\mathbf{S}^{(M)}) = D(\mathbf{S}^{(\lambda)}||\mathbf{S}^{(N)})$.

Note that

$$\begin{aligned} & D(\mathbf{S}^{(\lambda)}||\mathbf{S}^{(M)}) = D(\mathbf{S}^{(\lambda)}||\mathbf{S}^{(N)}) \\ \Leftrightarrow\ & \log_e \frac{|\mathbf{S}^{(M)}|}{|\mathbf{S}^{(N)}|} = \mathrm{tr}(((\mathbf{S}^{(N)})^{-1} - (\mathbf{S}^{(M)})^{-1})\mathbf{S}^{(\lambda)}). \end{aligned}$$

We have

$$\begin{aligned} & ((\mathbf{S}^{(N)})^{-1} - (\mathbf{S}^{(M)})^{-1})\mathbf{S}^{(\lambda)} \\ & = \frac{1}{\lambda}((\lambda(\mathbf{S}^{(M)})^{-1}\mathbf{S}^{(N)} + (1-\lambda)\mathbf{I})^{-1} - \mathbf{I}). \end{aligned}$$

It can be verified that

$$\begin{aligned} & (\lambda(\mathbf{S}^{(M)})^{-1}\mathbf{S}^{(N)} + (1-\lambda)\mathbf{I})^{-1} \\ & = \frac{1}{\theta}\begin{bmatrix} \frac{\lambda(1-c^{(M)}c^{(N)})}{1-(c^{(M)})^2} + 1 - \lambda & \frac{\lambda(c^{(M)}-c^{(N)})}{1-(c^{M})^2} \\ \frac{\lambda(c^{(M)}-c^{(N)})}{1-(c^{(M)})^2} & \frac{\lambda(1-c^{(M)}c^{(N)})}{1-(c^{(M)})^2} + 1 - \lambda \end{bmatrix}, \end{aligned}$$

where

$$\theta = -\frac{(c^{(N)} - c^{(M)})^2}{1-(c^{(M)})^2}\lambda^2 + \frac{2c^{(M)}(c^{(M)} - c^{(N)})}{1-(c^{(M)})^2}\lambda + 1.$$

As a consequence,

$$\begin{aligned}&\mathrm{tr}(((\mathbf{S}^{(N)})^{-1}-(\mathbf{S}^{(M)})^{-1})\mathbf{S}^{(\lambda)})\\&=\frac{2}{\theta}\Big(\frac{(c^{(N)}-c^{(M)})^2}{1-(c^{(M)})^2}\lambda+\frac{c^{(M)}(c^{(N)}-c^{(M)})}{1-(c^{(M)})^2}\Big).\end{aligned}$$

Therefore, $\lambda=\lambda^*$ is a root in $[0,1]$ of the following quadratic equation:

$$\alpha\lambda^2+\beta\lambda+\gamma=0, \tag{A1}$$

where

$$\begin{aligned}\alpha&=\frac{(c^{(N)}-c^{(M)})^2}{1-(c^{(M)})^2}\log_e\frac{|\mathbf{S}^{(M)}|}{|\mathbf{S}^{(N)}|},\\ \beta&=\frac{2(c^{(N)}-c^{(M)})^2}{1-(c^{(M)})^2}-\frac{2c^{(M)}(c^{(M)}-c^{(N)})}{1-(c^{(M)})^2}\log_e\frac{|\mathbf{S}^{(M)}|}{|\mathbf{S}^{(N)}|},\\ \gamma&=\frac{2c^{(M)}(c^{(N)}-c^{(M)})}{1-(c^{(M)})^2}-\log_e\frac{|\mathbf{S}^{(M)}|}{|\mathbf{S}^{(N)}|}.\end{aligned}$$

We shall show that Equation (A1) has a unique root in $[0,1]$, which is given by

$$\lambda^*=\frac{-\beta+\sqrt{\beta^2-4\alpha\gamma}}{2\alpha}. \tag{A2}$$

Clearly, Equation (A1) must have a root in $[0,1]$ since $D(\mathbf{S}^{(\lambda)}||\mathbf{S}^{(M)})|_{\lambda=0}>0$, $D(\mathbf{S}^{(\lambda)}||S^{(N)})|_{\lambda=1}>0$, and $D(\mathbf{S}^{(\lambda)}||\mathbf{S}^{(M)})|_{\lambda=1}=D(\mathbf{S}^{(\lambda)}||\mathbf{S}^{(N)})|_{\lambda=0}=0$. So it remains to prove the uniqueness of this root.

First consider the case $(c^{(N)})^2>(c^{(M)})^2$. It is clear that $\alpha>0$ and

$$\begin{aligned}\gamma&=\frac{2c^{(M)}(c^{(N)}-c^{(M)})}{1-(c^{(M)})^2}-\log_e\frac{|\mathbf{S}^{(M)}|}{|\mathbf{S}^{(N)}|}\\&=\frac{2c^{(M)}(c^{(N)}-c^{(M)})}{1-(c^{(M)})^2}-\log_e\frac{1-(c^{(M)})^2}{1-(c^{(N)})^2}\\&\le\frac{2c^{(M)}(c^{(N)}-c^{(M)})}{1-(c^{(M)})^2}-\frac{(c^{(N)})^2-(c^{(M)})^2}{1-(c^{(M)})^2}\\&=-\frac{(c^{(N)}-c^{(M)})^2}{1-(c^{(M)})^2}\\&<0,\end{aligned}$$

where the first inequality is due to $\log_e x\ge\frac{x-1}{x}$. Therefore, the two roots of Equation (A1) must be of different signs, which implies that there exists a unique root in $[0,1]$ with the expression given by Equation (A2).

Next consider the case $(c^{(N)})^2<(c^{(M)})^2$. Define $\overline{\lambda}=1-\lambda$. Equation (A1) can be written equivalently as

$$\alpha(1-\overline{\lambda})^2+\beta(1-\overline{\lambda})+\gamma=0,$$

i.e.,

$$\alpha\overline{\lambda}^2-(2\alpha+\beta)\overline{\lambda}+(\alpha+\beta+\gamma)=0. \tag{A3}$$

Note that

$$\begin{aligned}&2\alpha+\beta\\&\quad=\frac{2(c^{(N)}-c^{(M)})^2}{1-(c^{(M)})^2}-\frac{2c^{(N)}(c^{(M)}-c^{(N)})}{1-(c^{(M)})^2}\log_e\frac{|\mathbf{S}^{(M)}|}{|\mathbf{S}^{(N)}|}\end{aligned}$$

and

$$\begin{aligned}&\alpha+\beta+\gamma\\&\quad=\frac{2c^{(N)}(c^{(N)}-c^{(M)})}{1-(c^{(M)})^2}-\frac{1-(c^{(N)})^2}{1-(c^{(M)})^2}\log_e\frac{|\mathbf{S}^{(M)}|}{|\mathbf{S}^{(N)}|}.\end{aligned}$$

Therefore, Equation (A3) can be rewritten as

$$\overline{\alpha}\overline{\lambda}^2+\overline{\beta\lambda}+\overline{\gamma}=0, \tag{A4}$$

where

$$\begin{aligned}\overline{\alpha}&=\frac{(c^{(M)}-c^{(N)})^2}{1-(c^{(N)})^2}\log_e\frac{|\mathbf{S}^{(N)}|}{|\mathbf{S}^{(M)}|},\\ \overline{\beta}&=\frac{2(c^{(M)}-c^{(N)})^2}{1-(c^{(N)})^2}-\frac{2c^{(N)}(c^{(N)}-c^{(M)})}{1-(c^{(N)})^2}\log_e\frac{|\mathbf{S}^{(N)}|}{|\mathbf{S}^{(M)}|},\\ \overline{\gamma}&=\frac{2c^{(N)}(c^{(M)}-c^{(N)})}{1-(c^{(N)})^2}-\log_e\frac{|\mathbf{S}^{(N)}|}{|\mathbf{S}^{(M)}|}.\end{aligned}$$

A similar argument to that for the case $(c^{(N)})^2 > (c^{(M)})^2$ can be used to prove that Equation (A4) has one root in $[0,1]$ and the other root in $(-\infty,0)$. This implies that Equation (A1) must have one root in $[0,1]$ and the other root in $(1,\infty)$; the one in $[0,1]$ must be given by Equation (A2) (note that $\alpha < 0$ when $(c^{(N)})^2 < (c^{(M)})^2$).

References

1. Wengang, Z.; Houqiang, L.; Qi, T. Recent advance in content-based image retrieval: A literature survey. *arXiv* **2017**, arXiv:1706.06064.
2. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
3. Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987. [CrossRef]
4. Tan, X.; Triggs, B. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Trans. Image Process.* **2010**, *19*, 1635–1650. [PubMed]
5. Ojansivu, V.; Heikkilä, J. Blur insensitive texture classification using local phase quantization. In *International Conference on Image and Signal Processing*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 236–243.
6. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. *IEEE Conf. Comput. Vis. Pattern Recognit.* **2005**, *1*, 886–893.
7. Batool, A.; Nisar, M.W.; Shah, J.H.; Khan, M.A.; El-Latif, A.A.A. iELMNet: integrating novel improved extreme learning machine and convolutional neural network model for traffic sign detection. *Big Data* 2022, *ahead of print*. [CrossRef]
8. Nawaz, M.; Nazir, T.; Javed, A.; Tariq, U.; Yong, H.-S.; Khan, M.A.; Cha, J. An efficient deep learning approach to automatic glaucoma detection using optic disc and optic cup localization. *Sensors* **2022**, *22*, 434. [CrossRef]
9. Razavian, A.S.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN features off-the-shelf: an astounding baseline for recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; Institute of Electrical and Electronics Engineers: Columbus, OH, USA, 2014; pp. 806–813.
10. Khan, M.A.; Muhammad, K.; Sharif, M.; Akram, T.; Kadry, S. Intelligent fusion-assisted skin lesion localization and classification for smart healthcare. *Neural Comput. Appl.* **2021**, 1–16. [CrossRef]
11. Tolias, G.; Sicre, R.; Jégou, H. Particular object retrieval with integral max-pooling of CNN activations. *arXiv* **2015**, arXiv:1511.05879.
12. Babenko, A.; Lempitsky, V. Aggregating local deep features for image retrieval. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1269–1277.

13. Lin, J.; Duan, L.-Y.; Wang, S.; Bai, Y.; Lou, Y.; Chandrasekhar, V.; Huang, T.; Kot, A.; Gao, W. Hnip: Compact deep invariant representations for video matching, localization, and retrieval. *IEEE Trans. Multimed.* **2017**, *19*, 1968–1983. [CrossRef]
14. Kalantidis, Y.; Mellina, C.; Osindero, S. Cross-dimensional weighting for aggregated deep convolutional features. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 685–701.
15. Azhar, I.; Sharif, M.; Raza, M.; Khan, M.A.; Yong, H.-S. A decision support system for face sketch synthesis using deep learning and artificial intelligence. *Sensors* **2021**, *21*, 8178. [CrossRef] [PubMed]
16. Khan, S.; Khan, M.A.; Alhaisoni, M.; Tariq, U.; Yong, H.-S.; Armghan, A.; Alenezi, F. Human action recognition: A paradigm of best deep learning features selection and serial based extended fusion. *Sensors* **2021**, *21*, 7941. [CrossRef]
17. Noh, H.; Araujo, A.; Sim, J.; Weyand, T.; Han, B. Largescale image retrieval with attentive deep local features. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3456–3465.
18. Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; Sivic, J. NetVLAD: CNN architecture for weakly supervised place recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26–30 June 2016; pp. 5297–5307.
19. Radenović, F.; Tolias, G.; Chum, O. CNN image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 3–20.
20. Gordo, A.; Almazan, J.; Revaud, J.; Larlus, D. End-to-end learning of deep visual representations for image retrieval. *Int. J. Comput. Vis.* **2017**, *124*, 237–254. [CrossRef]
21. Hyvärinen, A.; Hurri, J.; Hoyer, P.O. Principal components and whitening. In *Natural Image Statistics*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 93–130.
22. Izenman, A.J. Linear discriminant analysis. In *Modern Multivariate Statistical Techniques*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 237–280.
23. Johnson, R.A.; Wichern, D.W. Canonical correlation analysis. In *Applied Multivariate Statistical Analysis*, 6th ed.; Pearson: Upper Saddle River, NJ, USA, 2018; pp. 539–574.
24. Gong, Y.; Ke, Q.; Isard, M.; Lazebnik, S. A multi-view embedding space for modeling internet images, tags, and their semantics. *Int. J. Comput. Vis.* **2014**, *106*, 210–233. [CrossRef]
25. Yan, F.; Mikolajczyk, K. Deep correlation for matching images and text. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3441–3450.
26. Dorfer, M.; Schlüter, J.; Vall, A.; Korzeniowski, F.; Widmer, G. End-to-end cross-modality retrieval with cca projections and pairwise ranking loss. *Int. J. Multimed. Inf. Retr.* **2018**, *7*, 117–128. [CrossRef]
27. Yu, Y.; Tang, S.; Aizawa, K.; Aizawa, A. Category-based deep cca for fine-grained venue discovery from multimodal data. *arXiv* **2018**, arXiv:1805.02997.
28. Lin, Z.; Peltonen, J. An information retrieval approach for finding dependent subspaces of multiple views. In *International Conference on Machine Learning and Data Mining in Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 1–16.
29. Yair, O.; Talmon, R. Local canonical correlation analysis for nonlinear common variables discovery. *IEEE Trans. Signal Process.* **2017**, *65*, 1101–1115. [CrossRef]
30. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv*, **2014**, arXiv:1409.1556.
31. Abdi, H. The eigen-decomposition: eigenvalues and eigenvectors. In *Encyclopedia of Measurement and Statistics*; SAGE Publications, Inc.: Thousand Oaks, CA, USA 2007; pp. 304–308.
32. Nielsen, F. An information-geometric characterization of chernoff information. *IEEE Signal Process. Lett.* **2013**, *20*, 269–272. [CrossRef]
33. Nielsen, F. Chernoff information of exponential families. *arXiv* **2011**, arXiv:1102.2684.
34. Prince, S.J. Common probability distribution. In *Computer Vision: Models, Learning and Inference*; Cambridge University Press: Cambridge, England, 2012; pp. 35–42.
35. Radenović, F.; Tolias, G.; Chum, O. Fine-tuning CNN image retrieval with no human annotation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*; Institute of Electrical and Electronics Engineers: Manhattan, NY, USA, 2018.
36. Schonberger, J.L.; Radenovic, F.; Chum, O.; Frahm, J.-M. From single image query to detailed 3d reconstruction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5126–5134.
37. Koren, Y.; Carmel, L. Robust linear dimensionality reduction. *IEEE Trans. Vis. Comput. Graph.* **2004**, *10*, 459–470. [CrossRef] [PubMed]
38. Li, T.; Zhu, S.; Ogihara, M. Using discriminant analysis for multi-class classification. In *Third IEEE International Conference on Data Mining*; IEEE Computer Society: Los Alamitos, CA, USA, 2003; pp. 589–589.
39. Mirkes, E.M.; Gorban, A.N.; Zinoviev, A. A Supervised PCA. 2016. Available online: https://github.com/Mirkes/SupervisedPCA (accessed on 10 Sepetember 2021).
40. Fisher, R.A. The use of multiple measurements in taxonomic problems. *Ann. Eugen.* **1936**, *7*, 179–188. [CrossRef]
41. Swets, D.L.; Weng, J.J. Using discriminant eigenfeatures for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **1996**, *8*, 831–836. [CrossRef]
42. Philbin, J.; Chum, O.; Isard, M.; Sivic, J.; Zisserman, A. Object retrieval with large vocabularies and fast spatial matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8.

43. Philbin, J.; Chum, O.; Isard, M.; Sivic, J.; Zisserman, A. Lost in quantization: Improving particular object retrieval in large scale image databases. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008.
44. Radenovic, F.; Iscen, A.; Tolias, G.; Avrithis, Y.; Chum, O. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In Proceedings of the IEEE Computer Vision and Pattern Recognition Conference, Salt Lake City, UT, USA, 18–23 June 2018.