Journal of *Imaging*

MDPI

Article

Scalable Database Indexing and Fast Image Retrieval Based on Deep Learning and Hierarchically Nested Structure Applied to Remote Sensing and Plant Biology

Pouria Sadeghi-Tehran [1,*], Plamen Angelov [2], Nicolas Virlet [1] and Malcolm J. Hawkesford [1]

[1] Department of Plant Sciences, Rothamsted Research, Harpenden AL5 2JQ, UK; nicolas.virlet@rothamsted.ac.uk (N.V.); malcolm.hawkesford@rothamsted.ac.uk (M.J.H.)
[2] School of Computing and Communications, InfoLab21, Lancaster University, Lancaster LA1 4WA, UK; p.angelov@lancaster.ac.uk
* Correspondence: pouria.sadeghi-tehran@rothamsted.ac.uk

Received: 6 November 2018; Accepted: 18 February 2019; Published: 1 March 2019

Abstract: Digitalisation has opened a wealth of new data opportunities by revolutionizing how images are captured. Although the cost of data generation is no longer a major concern, the data management and processing have become a bottleneck. Any successful visual trait system requires automated data structuring and a data retrieval model to manage, search, and retrieve unstructured and complex image data. This paper investigates a highly scalable and computationally efficient image retrieval system for real-time content-based searching through large-scale image repositories in the domain of remote sensing and plant biology. Images are processed independently without considering any relevant context between sub-sets of images. We utilize a deep Convolutional Neural Network (CNN) model as a feature extractor to derive deep feature representations from the imaging data. In addition, we propose an effective scheme to optimize data structure that can facilitate faster querying at search time based on the hierarchically nested structure and recursive similarity measurements. A thorough series of tests were carried out for plant identification and high-resolution remote sensing data to evaluate the accuracy and the computational efficiency of the proposed approach against other content-based image retrieval (CBIR) techniques, such as the bag of visual words (BOVW) and multiple feature fusion techniques. The results demonstrate that the proposed scheme is effective and considerably faster than conventional indexing structures.

Keywords: content-based image retrieval; deep convolutional neural networks; information retrieval; data indexing; recursive similarity measurement; deep learning; bag of visual words; remote sensing

1. Introduction

Today, digital images and videos are ubiquitous in every domain. The advancement in multi-media technologies has led to the generation of an enormous number of images and videos. The size of image repositories has increased rapidly in many domains, such as biology, remote sensing, medical, military, and web-searching. The use of automated data acquisition systems, such as modern phenotyping platforms [1–3] has revolutionized the way the data is collected and analyzed. The plant science community is seeking novel solutions to fully exploit all the potential offered by such new platforms equipped with high-resolution remote sensing sensors. Any large-scale dataset in modern biological sciences first and foremost requires reliable data infrastructure and an efficient information retrieval system. For image repositories of large scale, manual tagging is infeasible and is prone to errors, due to users' subjective opinions. Thus, to utilize such unstructured and complex image

collections, there is a substantial need for content-based image retrieval (CBIR) systems for browsing through images at a large scale and to classify, structure, and retrieve relevant information requested by the users.

Information retrieval (IR) refers to finding material (image repositories or documents) of an unstructured nature (image or text) that satisfies an information need from within large collections [4]. There is a fundamental difference between CBIR and search by text and metadata. Searching methods based on metadata rarely examine the content of an image itself but rather rely on manual annotations and tagging. In these systems, words are stored as ASCII character strings to describe image content. However, the high complexity of images cannot be described easily by keywords; thus, retrieval systems which are based solely on manual annotation often lead to unsatisfactory outcomes. In contrast, CBIR does not require keywords (manual annotation) and desired images are retrieved automatically based on their similarity to the query representation [5–7].

Although CBIR techniques are beginning to find a foothold in many applications, such as biology, remote sensing, satellite imaging, etc., the technology still suffers from lack of maturity due to a significant gap towards semantic-aware retrieval from visual content. A major challenge associated with CBIR systems is to extract information from an image which is unique and representative, to overcome the issue of so called *semantic-gap*. The *semantic-gap* refers to low-level features of images such as colors and texture, but those features might not be able to extract a higher level of understanding of the image perceived by humans [8]. Due to the absence of solid evidence on the effectiveness of CBIR techniques for high-throughput datasets with varied collections of images, opinion is still sharply divided regarding the reliability and performance of such systems in real-time. It is essential to standardize CBIR for easy access to data and speed up the retrieval process.

In this paper, a new concept of CBIR is employed to exploit the opportunities presented by large image-based repositories, particularly in remote sensing and plant biology. The proposed approach, which relies solely on the contents of the images, will pave the way for a computationally efficient and real-time image querying through an unstructured image database. An end-to-end CBIR framework is conducted without supervision. First, we utilize a deep CNN model as a feature extractor to obtain the feature representations from the activations of the convolutional layers. In the next step, a hierarchically nested database indexing structure and local recursive density estimation are developed to facilitate an efficient and fast retrieval process. Finally, the key elements of CBIR, accuracy and computational efficiency, are evaluated and compared with the state-of-the-art CBIR techniques.

2. Related Works

The core modules of any CBIR systems include image representation, database indexing, and image scoring, described as detailed below:

2.1. Image Representation

At the core, visual features affect every aspect of computer vision applications, including CBIR. The success of any CBIR system crucially depends on the feature representation of the images extracted by applying an image descriptor. Although over the past decades a variety of feature extraction techniques have been developed to find semantically richer image representations, it still remains one of the key challenges in CBIR applications.

2.1.1. Hand-crafted Feature Extraction Techniques

Handcrafted features are used excessively in conventional CBIR applications to quantify the contents of images. Earlier applications mainly focused on primitive features (global features) which describe an image as a whole to generalise the entire image as a single vector, such as contour representations, texture, or shape features.

- Color properties are extracted directly from the pixel densities over the whole image, segmented regions/bins, or sub-image. Image descriptors that characterize the color properties of an image seek to model the distribution of the pixel intensities in each channel of the image. These methods include color statistics, such as deviation, mean, and skewness, along with color histograms. Since color features are robust to background complications and are invariant to the size or orientation of an image, the color based methods have become one of the most common techniques in CBIR [9–11].
- Texture properties measure visual patterns in images that contain important information about the structural arrangement of surface i.e., fabric, bricks, etc. Texture descriptors seek to model the feel, appearance, and overall tactile quality of an object in an image and are defined as a structure of surfaces formed by repeating a particular element or several elements in different relative spatial distribution and synthetic structure. In general, the repetition involves local variations of scale, orientation, or other geometric and optical features of the elements [12,13].
- Shape properties can also be considered as one of the fundamental perceptual characteristics. Shape properties take on many non-geometric and geometric forms, such as moment invariants, aspect ratio, circularity, and boundary segments. There are difficulties associated with shape representation and descriptors techniques due to noise, occlusion, and arbitrary distortion, which often causes inaccuracies in extracting shape features. Nonetheless, the method has shown promising results to describe the image content [14,15].

Whilst the above techniques focus on primitive features, more recent techniques have been aimed to find semantically richer image representations by extracting a collection of local invariant features. The main advantage of semantic features is locality, which means that the extracted features are local and robust to clutter and occlusion. Also, individual features can be matched to a large database of objects and have close to real-time performance.

One of the most effective techniques is the bag of visual words technique [16,17]. The main reasons that BOVW has gained popularity in classification and retrieval applications are the use of powerful local descriptors, such as Scale Invariant Feature Transform (SIFT) [18], Speeded Up Robust Features (SURF) [19], and Binary Robust Invariant Scalable Keypoints (BRISK) [20]. In addition, the vector representations can be compared with standard distances, and subsequently be used for effective CBIR. However, the main drawback of BOVW is the high dimensional vector representing an image. Although a high-dimensional vector usually provides better exhaustive search results compared to a low-dimensional one, it is more difficult to index efficiently. Aggregated vectors, such as Fisher Vector (FV) [21] and Vector of Locally Aggregated Descriptors (VLAD) [22] aim to address this problem by encoding an image into a single vector, reducing the dimensionality without noticeably impacting the accuracy [16,17].

Nevertheless, despite the robustness of local descriptors techniques, global features are still desirable in a variety of computer vision applications. Ultimately, having an intimate knowledge of the dataset contents will provide a better perspective for which feature extraction techniques might be appropriate. For example, datasets that have relatively different color distributions, color descriptors will be more effective. Nonetheless, the effectiveness of hand-crafted feature representation in CBIR is inherently limited, as these approaches mainly operate at the primitive level. As presented in the following section, higher accuracy will be achieved by extracting semantic features from images based on learning-based features using deep networks.

2.1.2. Learning-based Features Using Deep Convolutional Neural Network

Recent years have witnessed the success of learning based features using Deep Neural Networks (DNNs) [23–25]. Unlike conventional global and local feature extraction methods, which often use shallow architecture and solely rely on human-crafted features, deep Conventional Neural Networks are considered the most well-known architecture for visual analysis [26]. CNN models attempt to model high-level abstractions in images by employing deep architectures composed of multiple

non-linear transformations [27]. In CNNs, features are extracted at multiple levels of abstracts and allow the system to learn complex functions that directly map raw sensory input data to the output, without relying on hand-engineered features using domain knowledge.

CNN has achieved state-of-the-art performance in a variety of applications, including natural language processing [28,29], speech recognition [30], and object recognition [31]. Inspired by the success of CNN in many computer vision applications, it has started to gain a foothold in the research area of CBIR. Subsequently, CNN models have been proposed to improve the image retrieval workflow [16,32,33]. For instance, in Sun et al. [34], features derived from local image regions identified with a general object detector and an adapted CNN model have been evaluated on two public large-scale image datasets. Lai et. al. [35] proposed simultaneous feature learning using deep neural networks and hash coding. The short binary codes resulted from hash coding achieved efficient retrieval and a considerable saving in memory usage. In other techniques, CNN descriptors are combined with conventional descriptors such as the VLAD representation [36,37]. Finally, in Mohedano et al. [38] authors proposed a method based on encoding the convolutional features of CNN and the BOVW aggregation scheme. The approach outperformed the state-of-the-art tested on landmark datasets.

2.2. Feature Indexing and Image Scoring

Another line of research in CBIR focuses on feature indexing and structuring the data vectors extracted from images. Feature indexing refers to structuring a database to facilitate search speed. Since one of the key features in CBIR systems is response time, the importance of feature indexing becomes more vivid, especially in a large-scale image database. An efficient database indexing can significantly accelerate the retrieval process and reduces memory usage substantially [39]. Conventional methods use a similarity metric to compare the feature vector of the query image to each and every single feature vector in the database. However, whilst comparing the query feature vector to the entire image dataset might be feasible for small datasets, this is still an $O(N)$ linear operation; thus, for large-scale datasets of billions of feature vectors, this is not computationally efficient. In [39,40], a hierarchical structure is formed based on low-level feature extraction techniques such as color, texture, and local mean clustering technique. The model is problem-specific and threshold-dependent. The main drawbacks are that the developed primitive features are not effective enough to represent images; in addition, the wrong choice of cluster radius may have a negative impact on the retrieval performance.

Two widely used indexing techniques in CBIR are inverted file index and hashing based indexing. An inverted index (also called "inverted file") is the central component of many search systems [41–45] as it facilitates faster and more scalable querying. Inspired by the field of information retrieval (i.e., text search engine), the inverted index stores mapping of unique word IDs to the document IDs in which the words occur [45]. It is easy to conceptualize an inverted index as a dictionary data structure with the word ID as the key and the value as a list of document IDs that contain the word.

The hashing based index projects images into a common Hamming space, while similar data will be mapped into similar binary codes [46–49]. The main concern of the existing hashing scheme such as locality sensitive hashing (LSH) [50] is an expensive memory cost. The reason is that these methods require to store the raw dataset representation vectors in memory, which is not scalable for a large-scale image database. Inspired by the success of deep networks, deep hashing methods have been proposed for image retrieval systems to take advantage of the deep network's image representation power [35,46,47,51].

3. Methodology

In this paper, we focus on three key challenges of any content-based image retrieval: image representation, database indexing, and image similarity measurement. Figure 1 illustrates an overall view of the proposed framework. The first step in the prescriptive analytics process is to transform the

initial unstructured and structured data sources into analytically prepared data. To achieve a balance between complexity and efficiency, a pre-trained CNN is used to utilize the ability of the model to produce better image representations for the retrieval task. We leverage an existing model trained on the ImageNet dataset [52], known as residual network (ResNet) [53]. The model is used as a fixed feature extractor without the last fully connected layer. The trained model provides access to the visual descriptors previously learnt by the CNN after processing millions of images in the ImageNet dataset without requiring a computational expensive training phase.

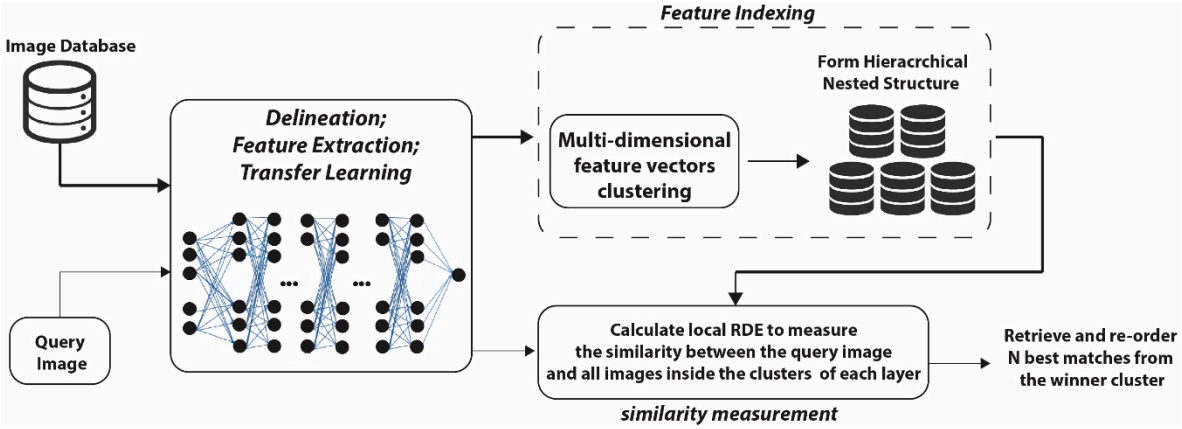

Figure 1. Schematic representation of the retrieval model.

Although the deep learning model is effective in extracting discriminative visual features from images (Section 4.2), it would compute multi-dimensional feature vectors (2048-D in our case) for every image which increases the computational complexity for feature indexing and querying. To address the multi-dimensional complexity caused by the CNN model, a novel nested hierarchical database indexing is proposed to facilitate fast querying. In addition, a recursive calculation based on local density estimation is used to measure the similarity between the given query and all the images from a given image cluster.

3.1. Representation Learning Using Residual Learning Model

CNNs process images through several layers, mainly in two parts of (a) the convolutional layers and max pooling layers and (b) the fully connected layers which are typically a linear classifier, such as softmax regression classifier (Figure 2). The convolutional layers are used to detect features whereas normalization and pooling layers control overfitting and reduce the number of weights. The last fully-connected layers are used for classification. Recent studies [23,54] indicate that it is feasible to adapt CNN models to extract semantic aware features by the activation of different layers in the networks [52]. Such generic descriptors derived from CNN are effective and powerful.

As mentioned, Neural Networks have the ability to learn and discover a good combination of features, even for complex tasks which would otherwise require a lot of human effort to be manually hand-crafted. In practice, it is common to pre-train a CNN on a very large dataset such as ImageNet dataset with 1.2 million images and 1000 categories, and then use the model either as an initialization for fine-tuning the CNN or use it as a fixed feature extractor, which is also known as *Representation Learning (RL)*. The main reason is that it is relatively rare to have a dataset big enough to train an entire CNN from scratch; additionally, training a CNN model from scratch will take considerable time to train across multiple GPUs on a large-scale dataset such as ImageNet.

Representation learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned [55]. In such a model, an existing pre-trained model is used as a starting point for a new task, such as classification. The conventional CNNs are treated as end-to-end image classifiers where an image forward propagates through the

network and the final probabilities are obtained from the end of the network. However, in the *representation learning*, instead of allowing the image to forward propagate through the entire network, we can stop the propagation at an arbitrary layer, such as the last fully connected layer, and extract the values from the network at this time, and then use them as feature vectors.

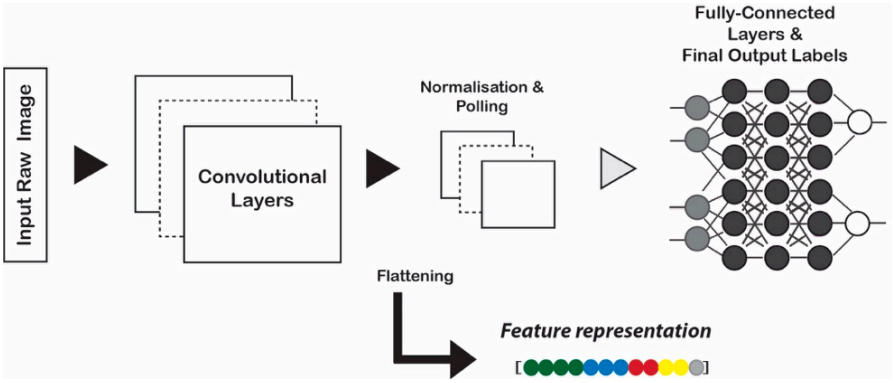

Figure 2. Representation learning scheme. Deep feature extraction from the pretrained Convolutional Neural Network (CNN) model.

In this study, we utilize the convolutional layers merely as a feature extractor. The aim is to generalize a trained CNN in learning discriminative feature representations for the images in our dataset. The trained model is used to derive feature vectors, more powerful than hand-designed algorithms such as SIFT, GIST, HOG, etc. We exploit the ability of a well-known deep convolutional neural network framework known as residual learning (ResNet) [53,56]. Residual learning frameworks ease the training of deeper networks and are a great candidate to capture the discriminative properties of images as a fixed feature extractor model. Network depth is a key element in neural network architecture; however, deeper networks are more difficult to train, as the accuracy gets saturated and then degrades rapidly. When deeper networks start converging, a degradation problem is exposed which is not caused by overfitting, while adding more layers causes even higher training error. In residual learning models, instead of learning a direct mapping of $x \rightarrow y$ with a function $H(x)$, the residual function is defined using $H(x) = F(x) + x$; where $F(x)$ and x represents residual mapping function and the identity function, respectively. The author's hypothesis is that it is easier to optimize $F(x)$ than to optimise the original mapping function, $H(x)$. We refer readers to [53,56] for more details.

The employed ResNet model has been pre-trained on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012, to classify 1.3 million images to 1000 ImageNet classes [52]. The ResNet consists of convolutional layers, pooling layers, and fully connected layers. The network takes images of size 224 $\times$ 224 pixels as input then passes through the network in a forward pass after applying filters to the input image. When treating networks as a fixed feature extractor, we cut off the network at an arbitrary point (normally prior to the last fully-connected layers); thus, all images will be extracted from the activations of convolutional feature maps directly. This would compute a 2048-D feature vector for every image that contains the hidden layer immediately before the classifier. The 2048-D feature vectors will be directly used for computing the similarity between images. The computational complexity and retrieval process may become cumbersome as the dimensionality grows. This requires us to optimize the retrieval process by proposing a hierarchically nested indexing structure and recursive similarity measurements to facilitate faster access and comparison of multi-dimensional feature vectors as described in the following sections.

3.2. Feature Indexing Based on Hierarchical Nested Data Clusters

The success of a CBIR not only depends on image delineation, but feature indexing and similarity measurement matrix also play vital roles to facilitate the execution of queries. In general, feature indexing refers to a database organizing structure to assist fast retrieval process. Whilst it

is feasible to retrieve information from datasets which are small in size by measuring the similarity between a query and every image in the dataset, the computational complexity will soon increase significantly on a larger scale image database.

In an attempt to address the challenges faced by retrieval information on a large-scale dataset, we present a hierarchically nested structure. The introduced database indexing aims at arranging and structuring the image database into a simple yet effective form of data clusters and hierarchies. Although forming a hierarchical structure for retrieval optimization has been explored before [57–60], the method presented in this study is quite different. Hierarchically nested data clusters are structured in which data clusters at higher layers represent one or multiple clusters at a lower layer based on mean values of the cluster centers (Figure 3). The first layer clusters are generated based on feature representations derived from the CNN model. Data clusters are formed by grouping the relevant data points using a partition-based clustering approach known as *K*-means clustering [61]. Figure 3 illustrates how the hierarchical structure of clusters is formed. μ and *X* are abstract values and denote mean values and scalar products explained in Section 3.3.

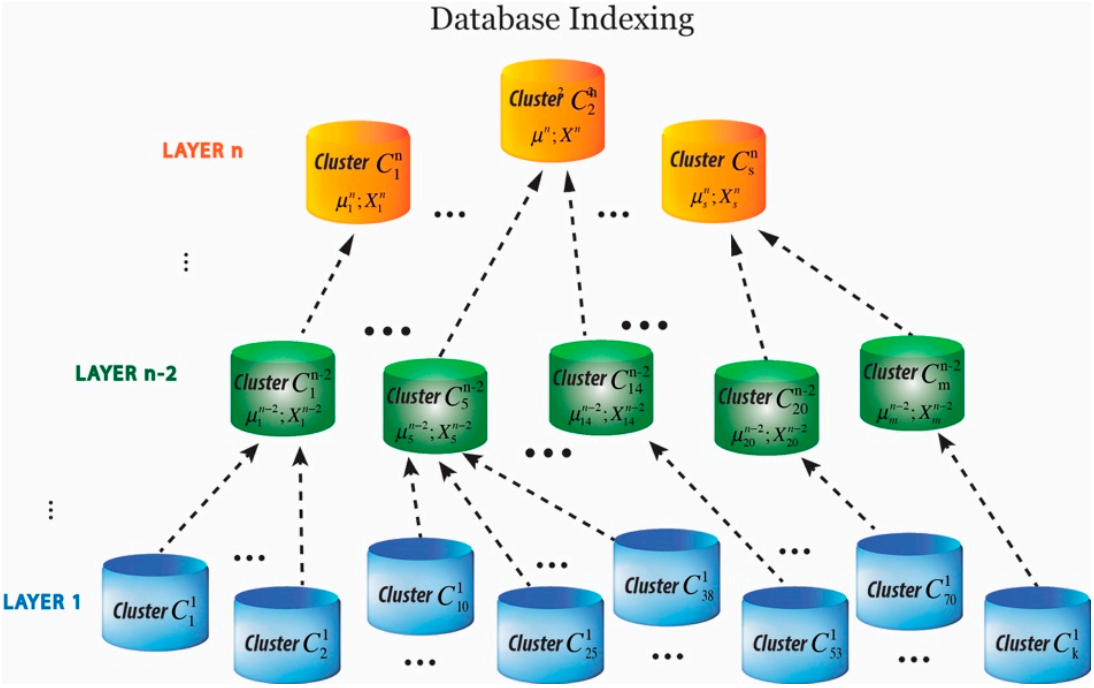

Figure 3. Schematic representation of the hierarchical nested indexing structure.

3.3. Fast Searching and Similarity Measure Based on Recursive Data Density Estimation

The final step after forming the hierarchically nested data clusters is to find the cluster which contains the most similar images to a query image. We applied recursive density estimation [62,63] to measure a similarity between the query image and all images inside each cluster recursively. The main idea of the recursive density function is to estimate the probability density function by a Cauchy type kernel and to recursively calculate it. The method is also applied for novelty detection in real-time data streams and video analytics [64]. The recursive calculation allows us to discard each data once it has been processed and only store the accumulated information in memory concerning the local mean (per cluster), μ and scalar product X. In order to speed up the retrieval process by an order of magnitude, the searching process is performed from the top of the pyramid in an ordered hierarchy based on "winner takes all" principle with maximum local recursive density estimation at each level (Figure 4).

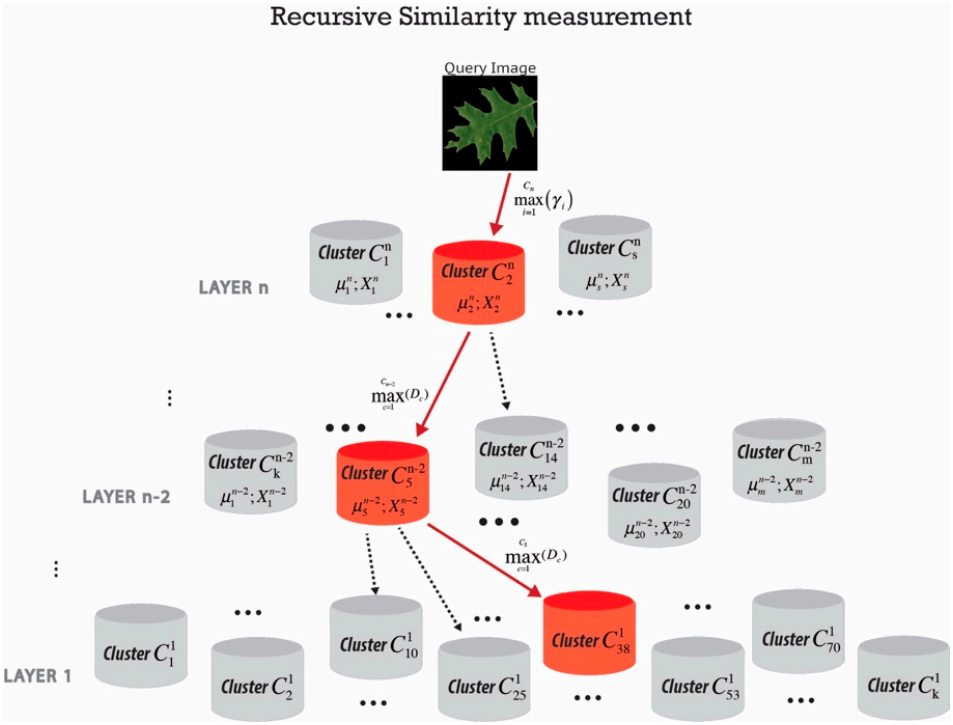

Figure 4. Schematic representation of searching through hierarchical nested structure and retrieve the most similar images (winner cluster) to the query.

The degree of similarity between the query image to images inside each cluster is measured by the relative local density with regards to the query image, which is defined by a suitable kernel over the distance between the current image sample and all the other images inside the cluster:

$$D_i^c = K\left(\sum_{j=1}^{M^c} d_{ij}^c\right) \quad c = [1, C] \tag{1}$$

where M^c is the number of images associated with c^{th} cluster; d_{ij}^c denotes the distance between the query image and any other image of the c^{th} cluster; $i = 1, 2, \ldots, N_c$; N is the number of images within c^{th} cluster.

Different types of distance measures can be used, such as Euclidean or Cosine distance. We used a Cauchy type of kernel to define the local density D_i^c. It can be proven that Cauchy type kernel asymptotically tends to Gaussian, but can be calculated recursively [63]:

$$D_i^c = \frac{1}{1 + \parallel F_i - \mu_i^c \parallel^2 + X_i^c - \parallel \mu_i^c \parallel^2} \tag{2}$$

$F = \{f_1, \cdots, f_{2048}\}$ is the feature vector. $i = 1, 2, \ldots, N_c$; N_c is the number of images within c^{th} cluster.

Both the mean, μ_i and the scalar product, X_i are updated recursively as follows [63]:

$$\mu_i = \frac{i-1}{i}\mu_{i-1} + \frac{1}{i}F_i; \mu_1 = F_1 \tag{3}$$

$$X_i = \frac{i-1}{i}X_{i-1} + \frac{1}{i} \parallel F_i \parallel^2; X_1 = \parallel F_1 \parallel^2 \tag{4}$$

Finally, the cluster with the maximum local density D^c, with respect to the query image, is most likely to contain similar images:

$$C_i^* = argmax_{c=1}^{C}\{D_i^c\} \quad (5)$$

The final step is the similarity measurement between the query image and all the images inside the winning cluster at the lowest layer. The relevance score is defined by distance-based scoring using City Block distance. Images are then ranked accordingly to their obtained scores. A smaller value of City Block distance implies that the corresponding image is more similar to the query image and vice versa. The City Block distance between the query image and images inside the winner cluster is calculated as follows:

$$d\left(I^j, Q\right) = \sum_{k=1}^{K}\left|Q_k - I_k^j\right|; j = 1, \ldots, N_c \quad (6)$$

where N_c is the number of images of winning cloud; K is the number of extracted features (K = 2048); Q denotes the query image; and I is the image in the winning cluster.

4. Experiments and Results

In this section, we present the experiments conducted to evaluate the key elements of CBIR: accuracy and computational efficiency. The deep learning framework and feature indexing were developed in Python using Keras API and MATLAB, respectively. The experiment was carried out on a desktop PC with Intel Core i7, processing power with 3.4 GHz CPU, 24 GB RAM, and GeForce GT 640 GPU running Ubuntu 16.04. Furthermore, the accuracy of the proposed approach is compared with two hand-crafted feature-based methods, known as BOVW and multiple fused global features (MFF). In addition, the computational efficiency and retrieval execution timing are evaluated against inverted file indexing and non-hierarchical searching.

Integrating multiple features: As mentioned, correct selection and utilizing appropriate features to represent an image are key elements for having a more accurate retrieval system [65,66]. The common approach is to combine color and texture properties to generate a robust feature representation [66,67]. In this study, two feature extractor techniques based on color and texture properties known and color correlogram and GIST are integrated. GIST descriptor [68] is widely used in scene classification to represent an image by a vector of spectral values which is based on spatial envelope properties, such as ruggedness, expansion, naturalness, and roughness. Color auto-correlogram, on the other hand, is used to preserve the spatial information of colors in an image. It describes the global distribution of local spatial correlations between identical colors [69].

Bag of visual words: In the BOVW method, the SIFT algorithm [18] is applied as a feature descriptor in addition to Local Linear Constraint (LLC) [70] to project the descriptors into the visual vocabulary and to reduce the computational complexity. In addition, to preserve the spatial relationships of the code vector, Spatial Pyramid Matching (SPM) [71] was developed where the entire image was divided into levels. Each image is divided into spatial sub-regions and histograms of features are computed from each sub-region. Each level divides the image into $2^l \times 2^{l-1}$; where l is level. The features are computed locally for each grid and the spatial information is incorporated into histograms. A three-level SPM is comprised of a single histogram in level 0, 4 histograms in level 1, and 16 histograms in level 2. In the end, the histograms from all the sub-regions are concatenated together to generate the final representation of the image. The result is a feature vector of $(1 + 4 + 16) \times K$; where K is the number of codebooks (K = 2000).

4.1. Datasets

MalayaKew (MK) Leaf-Dataset: This dataset [72] consists of a collection of leaves from 44 species class, with 52 images in each class. The data is in the form of digital images, size 256 × 256 pixels, collected at the Royal Botanic Garden, Kew, England. The dataset has been used solely for supervised image classification, since the dataset is extremely challenging as some of the classes have very similar appearances (Figure 5) making it extremely difficult to distinguish differences between classes with a fully unsupervised model, as was presented in this study. Although the MK dataset is not considered a big dataset, we believe the similarity between classes can be a good example to demonstrate how discriminative the features are between the convolutional neural networks and the hand-crafted methods.

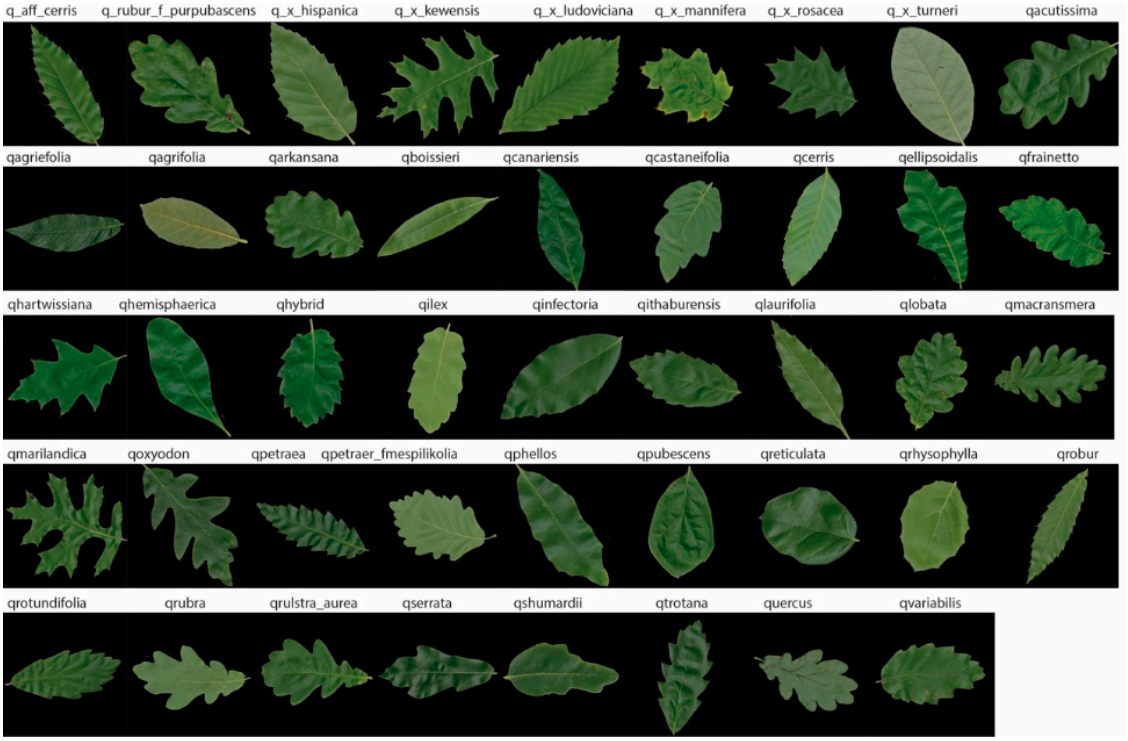

Figure 5. Sample images of MalayaKew 44 leaf collection.

The University of California Merced (UCM) Dataset: UCM dataset [73] consists of 21 land cover, large-scale aerial images from the USGS national map urban area imagery. Each class contains 100 images with 256 × 256 pixels; the spatial resolution of each pixel is 30 cm measured in the RGB spectral space. The dataset has been widely utilized for evaluating the performance of high-resolution remote sensing image scene classification [74–76]. The UCM dataset shows very small inter-class diversity among some categories that share a few similar texture patterns or objects, which makes this dataset very challenging. Some sample image scenes from the UCM dataset are shown in Figure 6.

Figure 6. Sample images of the University of California Merced (UCM) dataset.

4.2. Performance and Accuracy

Throughout this work, we use two evaluation metrics widely used to assess CBIR performance, known as mean Average Precision (*mAP*) and the precision at rank N (*P@N*). Average Precision (*AP*) is one of the most frequent methods used to evaluate the retrieval quality of a single query's retrieval results. *AP* takes consideration of both Precision (*Pr*) and Recall (*Re*). Precision is the fraction of retrieved images that are relevant, whereas Recall is the fraction of relevant images that are retrieved. *AP* averages the precision values from the rank positions where relevant images are retrieved. The mean average precision (*mAP*) is widely used to summaries the retrieval quality, which averages the *AP* over all queries. The definition of the above metrics follows below [4]:

$$AP = \frac{\sum_{k=1}^{n} P(k) \times rel(k)}{R} \tag{7}$$

where *P(k)* denotes the precision of top k retrieval results; *rel(k)* is a binary indicator function equaling 1 if the k^{th} retrieved results are relevant to the current query image and 0 otherwise; and R and n denote the number of relevant results for the current query image and the total number of retrieved results, respectively. Also, the precision at particular rank-N accuracy is another evaluation metric to evaluate CBIR performance. *P@N* score refers to the average number of same retrieved images, within the top-N ranked images. It should be noted that although *mAP* and *P@N* are widely used as evaluation metrics in CBIR, defining a suitable metric to measure the quality of results for an arbitrary query image is not a trivial process. In CBIR, it is hard to define the ground-truth since different users might have a different measure of similarity. If the degree of similarity of some of the images is very low, ignoring or not displaying those images is not critical and does not impact the overall performance of the system. Labelling images as non-relevant is not always satisfactory to the users. Any CBIR should have a certain tolerance for false positives, which often provides useful information.

In this study, to form a hierarchically nested pyramid, at the lower layer, images were grouped into a fixed number of clusters, while at the second layer, the means of the clusters at the first layer were further grouped into smaller numbers of clusters. Since the number of images in both datasets is

in the region of few thousands, two-layer hierarchies are enough to achieve real-time image querying. In MK and UCM datasets, based on our experience, the number of clusters at the first layer was set to 44 and 21 clusters (number of categories) and 10 and 4 clusters at the top layer, respectively.

The retrieval process begins by calculating the local recursive density estimation between the query image and all the clusters at the top layer and selecting the winning cluster with maximum local Recursive Density Estimation (RDE). The search continues at the lower layers, but only with the clusters which associated to the winning cluster at the top layer. Finally, images in the winning cluster at the lowest stage are ranked based on calculating the eigenvector distance to the query image.

4.2.1. Retrieval Performance on MalayaKew Leaf-Dataset

The results of the convolutional neural network as a feature extractor (RL-CNN) are shown in Figure 7 and Table 1. The precision accuracy at rank-20 is compared in Figure 7 based on 20 queries. The queries were selected to tackle every range of visual appearances with a unique shape, such as *qoxyodon*, or similar appearances, like *q-aff-cerris* and *qlaurifolia.*

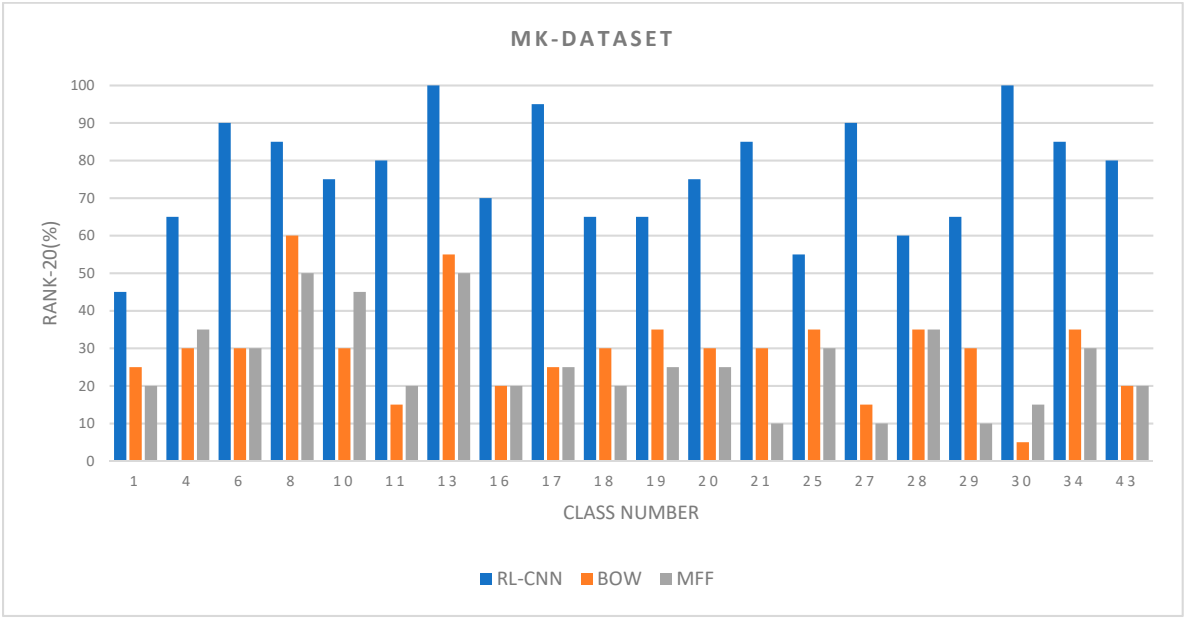

Figure 7. The retrieval Rank-20 accuracy between the Convolutional Neural Network (CNN) as a feature extractor, bag of visual words, and multiple feature fusion (color and texture).

Table 1. The retrieval accuracy *mAP* of convolutional neural network as a feature extractor (RL-CNN), bag of visual words (BOVW), and multiple fused global features (MFF) on Malaya–Kew (MK) and University of California Merced (UCM) datasets.

Dataset	Method	*mAP* (%)
MalayaKew	FE-CNN	88.1%
	BOVW	66.2%
	MFF	52.6%
UCM	FE-CNN	90.5%
	BOVW	86.2%
	MFF	69.8%

Several observations can be achieved from the precision results. The RL-CNN method outperformed the two state-of-the-art techniques by a large margin. The proposed method not only performed well on classes with unique visual appearances, such as *qlobata* or *qpetraea*, but it also distinguished categories with similar appearances, such as *quercus* and *q-x-kewensis*. In RL-CNN

method, *q-x-mannifera*, *qboissieri*, *qellipsoidalis*, *qmacransmera*, and *qpetraea* obtained maximum accuracy with over 90%, whereas *qlaurifolia* and *q-aff-cerris* had the lowest value of 55% and 45%, respectively. The *qlaurifolia* class achieved 55% accuracy, whereas 9 out of 20 images belong to *qcanariensis*, *qrhysophylla*, and *qtrotana* categories (Figure 8A). The accuracy dropped to 35% and 30% in BOVW and MFF, accordingly. The *q-aff-cerris* class obtained the lowest accuracy in RL-CNN with 45% accuracy rate, whereas 11 out of 20 images belong to *qrobur* category, which is visually almost identical to the query image.

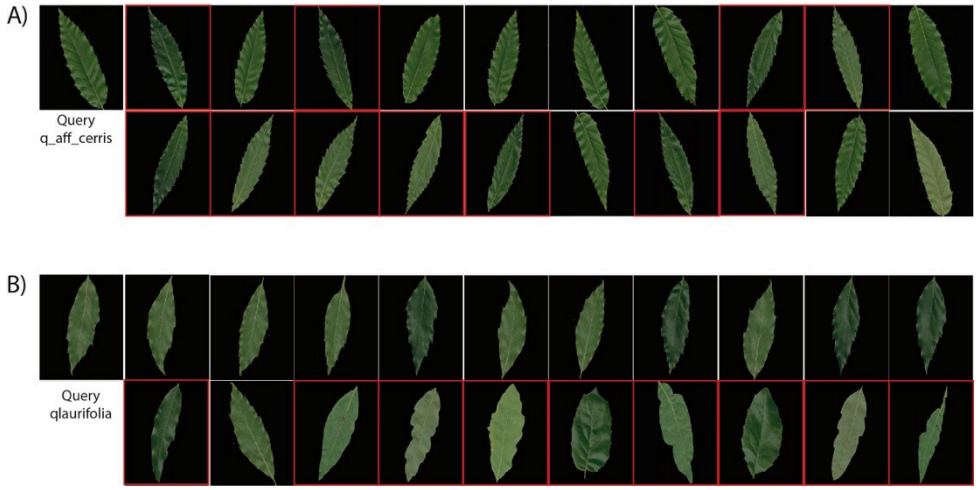

Figure 8. Qualitative evaluation of the proposed image retrieval on the two lowest performance of classes in Malaya–Kew Leaf-Dataset (**A**) Retrieval result from qlaurifolia class (**B**) retrieval result from q-aff-cerris class. The first image is the query and the following images are the images most similar to the query image. The retrieved images wrongly categorized are highlighted in red.

On the other hand, the BOVW and MFF performed poorly in identifying small differences between leaf varieties in MK dataset. Both methods retrieved images with the visual similarity to queries; however, they failed to distinguish small visual differences among classes. As illustrated in Figure 7, BOVW performed better than MFF in most cases, except classes *q_rubur_f_purpubascens*, *qagriefolia*, *qagrifolia*, and *qpetraea*. (The results for each class are presented in the Supplementary Materials).

Table 1 summarizes the *mAP* evaluation of the Malaya–Kew leaf dataset. The results are obtained from 20 queries in which the retrieval system can be tested and evaluated. The best accuracy score is 88.1%, achieved by RL-CNN, followed by BOVW and MFF with 66.2% and 52.6%, respectively.

4.2.2. Retrieval Performance on UCM Dataset

The precision performances at *P*@40 for the UCM dataset are shown in Figure 9. The results show that RL-CNN method outperformed both the BOW and MFF by achieving better accuracy in all categories except *baseball diamond* category. In RL-CNN, high accuracy results obtained in *agricultural*, *beach*, *forest*, *harbor*, *chaparral*, and *airplane* (Figure 10) categories. On the other hand, in the *baseball-diamond*, *dense residential* (Figure 11), and *freeway* (Figure 12), RL-CNN achieved the lowest accuracy with 41%, 35%, and 50%, respectively (Figure 9).

Figure 11 shows the retrieval results of the *dense building* class on a randomly given query. The class achieved 35% accuracy, whereas 14 out of 40 images belong to the same class as the query image. However, the rest of the images are still visually similar to the query retrieved from *medium residential* and *mobile home park* classes. The *freeway* class with 50% accuracy has a similar performance, whereas half of the retrieved images belong to *runway* and *overpass* classes, which are still visually very similar to the *freeway* class (Figure 12).

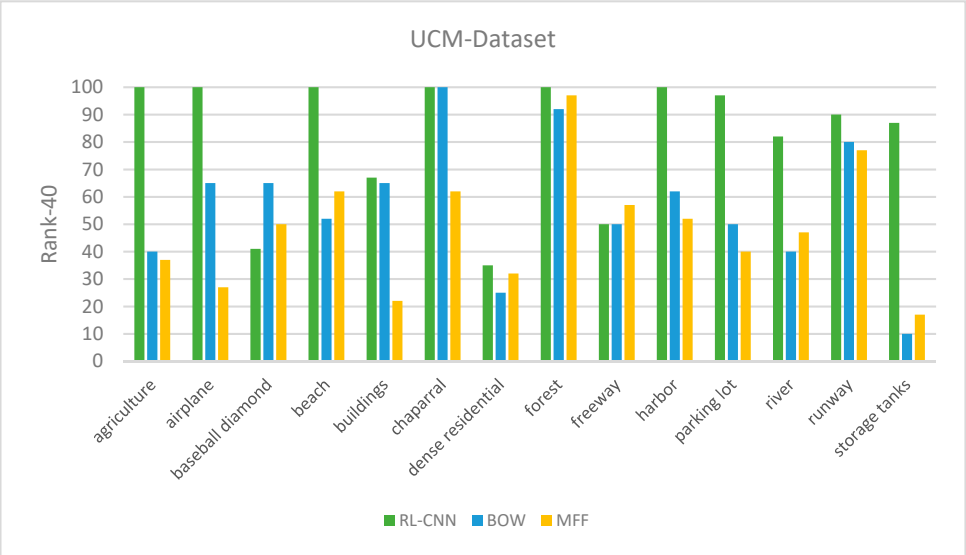

Figure 9. The retrieval Rank-40 accuracy between feature extractor using convolutional neural network, bag of visual words, and multiple feature fusion (color and texture).

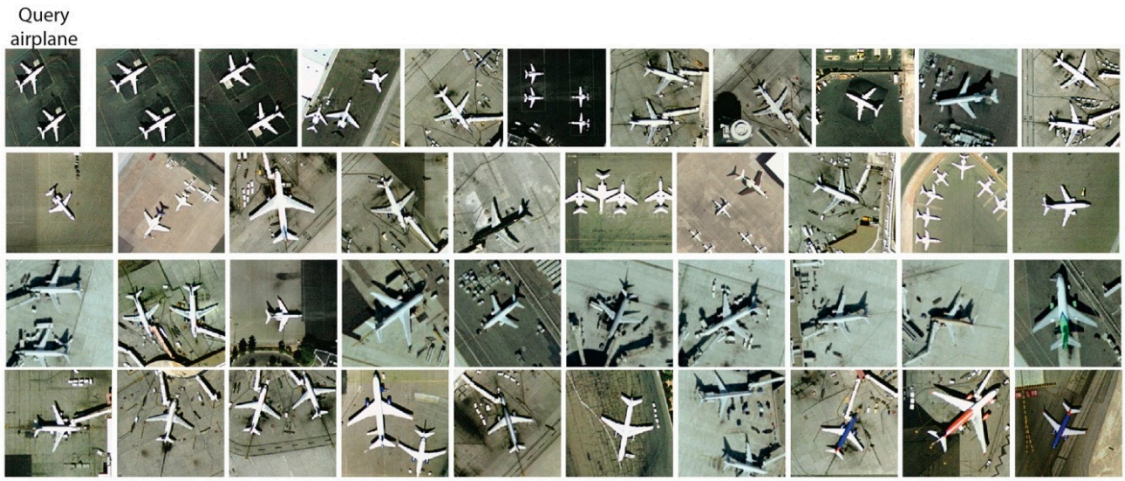

Figure 10. Retrieval results of airplane category using convolutional neural network as a feature extractor (RL-CNN). The methods obtained 100% retrieval accuracy.

Figure 11. Retrieval results of dense-building category using convolutional neural network as a feature extractor (RL-CNN). The green rectangles indicate correct retrieval results.

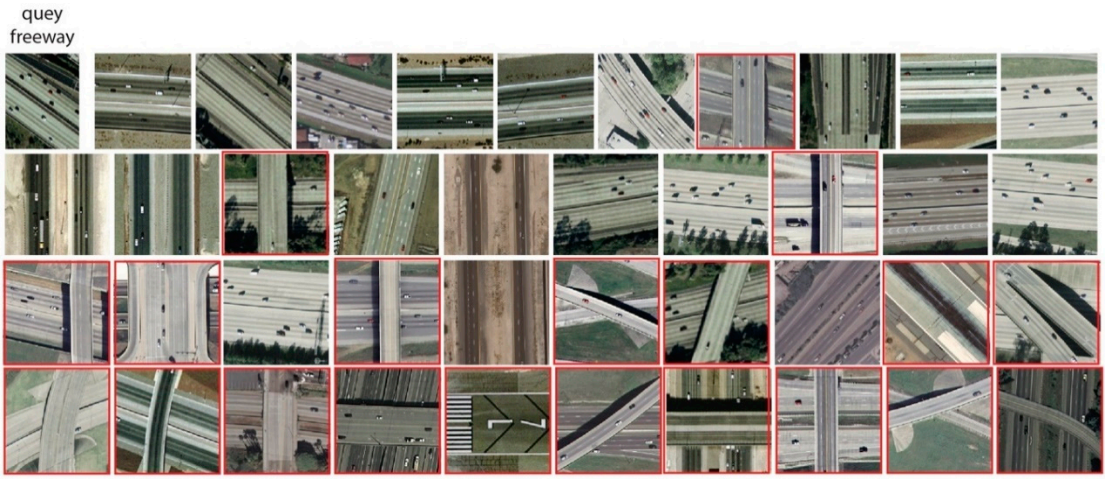

Figure 12. Retrieval results of freeway category using convolutional neural network as a feature extractor (RL-CNN). The red rectangles indicate incorrect retrieval results.

The retrieval *mAP* of different models on the UCM image dataset are listed in Table 1. As shown in the table, the RL-CNN outperformed both the state-of-the-art techniques. The *mAP* measure in RL-CNN is 90.1%, whereas the BOW and MFF achieved 86.2% and 69.8%, respectively.

4.3. Retrieval Time Per Query

Commercial CBIR applications are often assessed for requirements in computational capacity and memory efficiency. As mentioned earlier, the proposed hierarchically nested structure is beneficial for the retrieval performance in terms of search time and memory size required to store the indexed images. In this section, we provide more details on the retrieval time per query between the proposed method, the inverted index file method, and the non-hierarchical searching with a single layer. The non-hierarchical searching technique processes each image by scanning all image patches and computing similarity values for every individual image, unlike the nested hierarchical indexing described in Section 3.2.

Figure 13 shows the execution time of CNN-hierarchically nested structure, CNN sequential searching, BOVW-inverted indexing technique, and BOVW-non-hierarchical retrieval method. The non-hierarchical method processes each image by scanning all images and computing similarity values between query images and all images. In contrast, the hierarchically nested indexing avoids comparing the query image with every image in the dataset by grouping similar images together and measures the similarity based on recursive density estimation. The inverted index technique also avoids performing a linear search over all images in the dataset, helping us to speed up the querying process. In the inverted index method, we query our inverted index to find images that contain the same visual word as the query image. Then, we only compare images in the dataset that contain a significant number of visual words as the query.

As illustrated in Figure 13, the proposed hierarchical indexing method achieved the fastest retrieval performance. The average retrieval time of RL-CNN with hierarchical indexing scheme on the MK and UCM image datasets are 0.039 and 0.025 seconds, respectively. RL-CNN with sequential searching came second with 0.164 and 0.142 seconds, respectively. Nevertheless, this is an *O(N)* linear operation; thus, the execution time increases considerably in the sequential searching if the number of images increases to hundreds of thousands or millions. BOVW with inverted index and BOVW without indexing had the slowest retrieval time, with 0.29 and 1.8 seconds and 0.66 and 3.9 seconds in MK and UCM datasets, respectively. Moreover, the RL-CNN with sequential searching showed faster performance compared to the BOW with a similar structure. This can be justified since the RL-CNN computes a 2048-D feature vector, whereas BOW generates a 42,000-D feature representation (Section 4,

the bag of visual words). As a result, it takes more time to compute similarity for every individual image in the dataset.

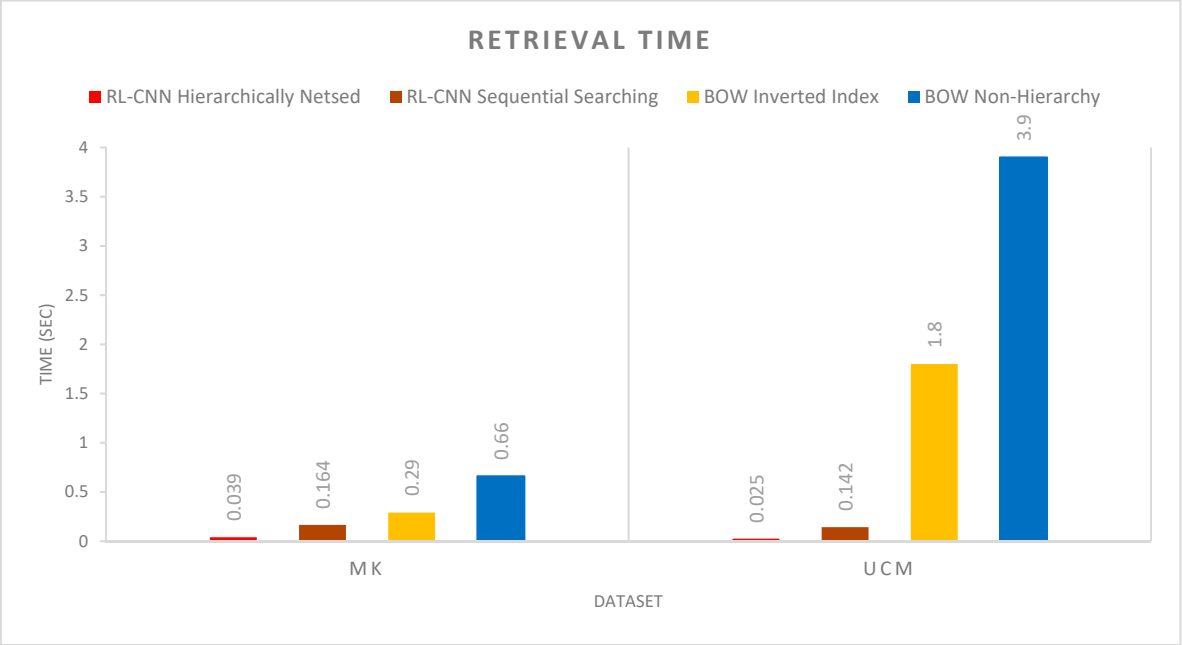

Figure 13. Execution times in seconds of the hierarchically nested indexing, inverted index, and non-hierarchical content-based image retrieval (CBIR), tested on Malaya–Kew and The University of California Merced datasets.

4.4. Discussion, Challenges and Future Work

Although the ability to retrieve digital images with relatively high accuracy and low computational efficiency was presented in this study, challenges remain in terms of optimizing the CNN model to derive better feature representations as well as developing a dynamic clustering technique to group similar images and form a hierarchically nested pyramid.

In this study, we applied pre-existing network architecture pre-trained on data of some "related" domain and use it as feature extractor. However, if the testing dataset is not related to the training dataset that the pre-existing network is trained on (for example, hyperspectral or medical imagery), the pre-trained model will most likely have difficulty deriving discriminative features from the testing dataset. There is a type of transfer learning (TL) called fine-tuning that exists to leverage unlabeled data. Typically, these techniques attempt to pre-train the weights of the classification network, by iteratively training each layer to reconstruct the images. A combination of these techniques and pre-trained network is often used to improve convergence.

In terms of database structure and querying, the proposed indexing technique is an approximation based on visual similarity. Since the similarity measurement is based on data density estimation, the nearest neighbor will be either in the winning cluster or in the edge/border of another cluster, but in most cases in the same cluster, as the high accuracy achieved by the proposed methods presented in Section 4.2 indicates. However, the assumption can be modified from "winner takes all" to "few winners take all" to also include similar images fall into border clusters.

As shown, cluster/group feature vectors extracted from images based on their similarities will reduce the computational complexity of CBIR; however, in feature vectors with high dimension, data becomes very sparse and distance measures become increasingly meaningless, resulting in low performance of CBIR. Moreover, in some applications where the number of image categories is unknown, the difficulties become more vivid when the clustering method has a static nature and pre-defined structure, such as *K*-means. Future work will tackle dynamic clustering methods without

the requirement of pre-defining the number of clusters in advance. The advantage of using such a model is that if new images are added to the dataset, the clustering images and forming the hierarchical structure will not be repeated from scratch.

Another improvement will be adding relevant feedback which enables users to have more interaction with the system and provide feedback on the relevance of the retrieved images. The feedback can be used for learning and improving the performance of the CBIR.

5. Conclusions

The research scope for this paper focused on highly scalable and memory efficient image retrieval system. The aim was to overcome the limitations of conventional retrieval methods in the field of plant biology and remote sensing to significantly boost the retrieval performance in terms of accuracy and computational efficiency. The challenge was to preserve multi-dimensional and high discriminative image representations derived by the CNN model and still maintain the computational efficiency of the querying process. It is worth highlighting the following advantages of the proposed method:

- **Fast Retrieval time**: The proposed approach improves the retrieval process and is over 16 times faster than the traditional brute-force sequential searching which is vital for large-scale databases.
- **Scalability**: The model is constructed in a hierarchical structure. The feature indexing in a hierarchical form can handle a dynamic image database and can be easily integrated into the server-client architecture.
- **Unsupervised data mining**: The proposed technique does not require any prior knowledge of image repositories or any human intervention. However, in future work, human input/feedback can potentially improve the performance.
- **Recursive similarity measurement**: The similarity measurements are done recursively, which significantly reduces memory cost in high-scale multimedia CBIR systems.
- **Discriminative power for quantifying images**: Transfer learning is applied by utilizing a pre-trained deep neural network model merely as a feature extractor. The results indicate that the generic descriptors extracted from the CNNs are effective and powerful, and performed consistently better than conventional content-based retrieval systems.

Furthermore, although the visual content was the main focus of this study, integrating keywords and text to the CBIR pipeline can capture images' semantic content and describe images which are identical by linguistic clues.

Supplementary Materials: The following are available online at http://www.mdpi.com/2313-433X/5/3/33/s1, Additional file 1: The retrieval results of the CNN, BOW, and MFF methods tested on MK dataset. Additional file 2: The retrieval results of the CNN, BOW, and MFF methods tested on MK dataset. The results are presented in both HTML and PNG formats. It should be noted that the retrieval images in the HTML files will not be displayed due the missing link with the local hard drive stored the images; however, the class labels of the retrieved data can be validated in the HTML source code.

Author Contributions: Conceptualization, P.S.T. and P.A.; methodology, P.S.T. and P.A.; software, P.S.T.; validation, P.S.T.; formal analysis, P.S.T.; investigation, P.S.T.; resources, P.S.T.; data curation, P.S.T.; writing—original draft preparation, P.S.T.; writing—review and editing, P.S.T., N.V., M.J.H., and P.A.; visualization, P.S.T.; supervision, M.J.H., and P.A.; project administration, M.J.H.; funding acquisition, M.J.H.

Funding: Rothamsted Research receives support from the Biotechnology and Biological Sciences Research Council (BBSRC) of the UK as part of the Designing Future Wheat (BBS/E/C/000I0220) project.

Acknowledgments: Rothamsted Research receives support from the Biotechnology and Biological Sciences Research Council (BBSRC) of the UK as part of the Designing Future Wheat (BBS/E/C/000I0220) project).

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

Abbreviations

The following abbreviations are used in this manuscript:

CBIR	Content Based Image Retrieval
BOVW	Bag of Visual Words
SIFT	Scale Invariant Feature Transform
CNN	Convolutional Neural Network
DNN	Deep Neural Network
TL	Transfer Learning
RDE	Recursive Density Estimation
SPM	Spatial Pyramid Matching
LLC	Local Linear Constraint
FV	Fisher Vector
VLAD	Vector of Locally Aggregated Descriptors
MFF	Multiple Feature Fusion
IR	Information Retrieval
RL	Representation Learning

References

1. Virlet, N.; Sabermanesh, K.; Sadeghi-Tehran, P.; Hawkesford, M.J. Field Scanalyzer: An automated robotic field phenotyping platform for detailed crop monitoring. *Funct. Plant Biol.* **2017**, *44*, 143. [CrossRef]
2. Busemeyer, L.; Mentrup, D.; Möller, K.; Wunder, E.; Alheit, K.; Hahn, V.; Maurer, H.P.; Reif, J.C.; Würschum, T.; Müller, J.; et al. BreedVision—A Multi-Sensor Platform for Non-Destructive Field-Based Phenotyping in Plant Breeding. *Sensors* **2013**, *13*, 2830–2847. [CrossRef] [PubMed]
3. Kirchgessner, N.; Liebisch, F.; Yu, K.; Pfeifer, J.; Friedli, M.; Hund, A.; Walter, A. The ETH field phenotyping platform FIP: A cable-suspended multi-sensor system. *Funct. Plant Biol.* **2017**, *44*, 154. [CrossRef]
4. Larson, R.R. Introduction to Information Retrieval. *J. Am. Soc. Inf. Sci.* **2010**, *61*, 852–853. [CrossRef]
5. Datta, R.; Joshi, D.; Li, J.; Wang, J.Z. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.* **2008**, *40*, 5–60. [CrossRef]
6. Lew, M.; Sebe, N.; Djeraba, C.; Jain, R. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimed. Comput. Commun. Appl.* **2006**, *2*, 1–19. [CrossRef]
7. Smeulders, A.W.M.; Worring, M.; Santini, S.; Gupta, A.; Jain, R. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1349–1380. [CrossRef]
8. Alzu'bi, A.; Amira, A.; Ramzan, N. Semantic content-based image retrieval: A comprehensive study. *J. Vis. Commun. Image Represent.* **2015**, *32*, 20–54. [CrossRef]
9. Yu, H.; Li, M.; Zhang, H.-J.; Feng, J. Color texture moments for content-based image retrieval. In Proceedings of the International Conference on Image Processing, Rochester, NY, USA, 22–25 September 2002; pp. 929–932.
10. Lin, C.-H.; Chen, R.-T.; Chan, Y.-K. A smart content-based image retrieval system based on color and texture feature. *J. Image Vis. Comput.* **2009**, *27*, 658–665. [CrossRef]
11. Singh, S.M.; Hemach, K.; Hemachandran, K. Content-Based Image Retrieval using Color Moment and Gabor Texture Feature. *IJCSI Int. J. Comput. Sci.* **2012**, *9*, 299–309.
12. Guo, Y.; Zhao, G.; Pietikainen, M. Discriminative features for texture description. *Pattern Recognit.* **2012**, *45*, 3834–3843. [CrossRef]
13. Ahonen, T.; Matas, J.; He, C.; Pietikainen, M. Rotation invariant image description with local binary pattern histogram fourier features. In Proceedings of the 16th Scandinavian Conference on Image Analysis (SCIA 2009), Oslo, Norway, 15–18 June 2009; Springer: Berlin/Heidelberg, Germany, 2009.
14. Mezaris, V.; Kompatsiaris, I.; Strintzis, M.G. An ontology approach to object-based image retrieval. In Proceedings of the 2003 International Conference on Image Processing (Cat. No.03CH37429), Barcelona, Spain, 14–17 September 2003.
15. Nikkam, P.S.; Reddy, B.E. A Key Point Selection Shape Technique for Content based Image Retrieval System. *Int. J. Comput. Vis. Image Process.* **2016**, *6*, 54–70. [CrossRef]

16. Zhou, W.; Li, H.; Tian, Q. Recent Advance in Content-based Image Retrieval: A Literature Survey. *arXiv* **2017**, arXiv:1706.06064.
17. Tsai, C.F. Bag-of-words representation in image annotation: A review. *ISRN Artif. Intell.* **2012**, *2012*. [CrossRef]
18. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
19. Bay, H.; Tuytelaars, T.; Gool, L. Surf: Speeded Up Robust Features. In Proceedings of the 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 404–417.
20. Leutenegger, S.; Chli, M.; Siegwart, R.Y. Brisk: Binary Robust Invariant Scalable Keypoints. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2548–2555.
21. Perronnin, F.; Liu, Y.; Sánchez, J. Large-scale image retrieval with compressed fisher vectors. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010.
22. Jegou, H.; Douze, M.; Schmid, C. Aggregating local descriptors into a compact image representation. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010.
23. Bengio, Y. Learning Deep Architectures for AI. *Found. Trends®Mach. Learn.* **2009**, *2*, 1–127. [CrossRef]
24. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
25. 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, 7–12 June 2015; pp. 1–9.
26. Tzelepi, M.; Tefas, A. Deep convolutional learning for Content Based Image Retrieval. *Neurocomputing* **2018**, *275*, 2467–2478. [CrossRef]
27. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef] [PubMed]
28. Johnson, R.; Zhang, T. Semi-supervised Convolutional Neural Networks for Text Categorization via Region Embedding. In Proceedings of the Twenty-Ninth Conference on Neural Information Processing Systems (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015.
29. Shen, Y.; He, X.; Gao, J.; Deng, L.; Mesnil, G. A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval. In Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, Shanghai, China, 3–7 November 2014.
30. Abdel-Hamid, O.; Mohamed, A.R.; Jiang, H.; Deng, L.; Penn, G.; Yu, D. Convolutional Neural Networks for Speech Recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2014**, *22*, 1533–1545. [CrossRef]
31. Borji, A.; Cheng, M.-M.; Jiang, H.; Li, J. Salient Object—A Benchmark. *IEEE Trans. Image Process.* **2015**, *24*, 5706–5722. [CrossRef] [PubMed]
32. Tzelepi, M.; Tefas, A. Deep convolutional image retrieval: A general framework. *Signal Process. Image Commun.* **2018**, *63*, 30–43. [CrossRef]
33. Wan, J.; Wang, D.; Hoi, S.; Wu, P.; Zhu, J. Deep learning for content-based image retrieval: A comprehensive study. In Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 157–166.
34. Sun, S.; Zhou, W.; Tian, Q.; Li, H. Scalable Object Retrieval with Compact Image Representation from Generic Object Regions. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2016**, *12*, 29. [CrossRef]
35. Lai, H.; Pan, Y.; Liu, Y.; Yan, S. Simultaneous Feature Learning and Hash Coding with Deep Neural Networks. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3270–3278.
36. Gong, Y.; Wang, L.; Guo, R.; Lazebnik, S. Multi-scale Orderless Pooling of Deep Convolutional Activation Features. In Proceedings of the 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Springer: Cham, Switzerland, 2014; Volume 8695, pp. 392–407.

37. Ng, J.Y.-H.; Yang, F.; Davis, L.S. Exploiting local features from deep networks for image retrieval. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Boston, MA, USA, 7–12 June 2015; pp. 53–61.
38. Mohedano, E.; McGuinness, K.; O'Connor, N.E.; Salvador, A.; Marques, F.; Giro-i-Nieto, X. Bags of Local Convolutional Features for Scalable Instance Search. In Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, New York, NY, USA, 6–9 June 2016; ACM Press: New York, NY, USA, 2016; pp. 327–331.
39. Angelov, P.; Sadeghi-Tehran, P. Look-a-Like: A Fast Content-Based Image Retrieval Approach Using a Hierarchically Nested Dynamically Evolving Image Clouds and Recursive Local Data Density. *Int. J. Intell. Syst.* **2016**, *32*, 82–103. [CrossRef]
40. Angelov, P.; Sadeghi-Tehran, P. A Nested Hierarchy of Dynamically Evolving Clouds for Big Data Structuring and Searching. *Procedia Comput. Sci.* **2015**, *53*, 1–8. [CrossRef]
41. Cai, J.; Liu, Q.; Chen, F.; Joshi, D.; Tian, Q. Scalable Image Search with Multiple Index Tables. In Proceedings of the International Conference on Multimedia Retrieval, Glasgow, UK, 1–4 April 2014; p. 407.
42. Nister, D.; Stewenius, H. Scalable Recognition with a Vocabulary Tree. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; pp. 2161–2168.
43. Zhou, W.; Lu, Y.; Li, H.; Song, Y.; Tian, Q. Spatial coding for large scale partial-duplicate web image search. In Proceedings of the 18th ACM International Conference on Multimedia, Firenze, Italy, 25–29 October 2010; pp. 511–520.
44. Wu, Z.; Ke, Q.; Isard, M.; Sun, J. Bundling features for large scale partial-duplicate web image search. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 25–32.
45. Bartolini, I.; Patella, M. Windsurf: the best way to SURF. *Multimed. Syst.* **2018**, *24*, 459–476. [CrossRef]
46. Zhang, J.; Peng, Y.; Ye, Z. Deep Reinforcement Learning for Image Hashing. *arXiv* **2018**, arXiv:1802.02904.
47. Liu, H.; Wang, R.; Shan, S.; Chen, X. Deep Supervised Hashing for Fast Image Retrieval. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2064–2072.
48. Jiang, K.; Que, Q.; Kulis, B. Revisiting Kernelized Locality-Sensitive Hashing for Improved Large-Scale Image Retrieval. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 4933–4941.
49. Tang, J.; Li, Z.; Wang, M. Neighborhood discriminant hashing for large-scale image retrieval. *IEEE Trans. Image Process.* **2015**, *24*, 2827–2840. [CrossRef] [PubMed]
50. Datar, M.; Immorlica, N.; Indyk, P.; Mirrokni, V.S. Locality-sensitive hashing scheme based on p-stable distributions. In Proceedings of the Twentieth Annual Symposium on Computational Geometry, Brooklyn, NY, USA, 8–11 June 2004; pp. 253–262.
51. Cao, Z.; Long, M.; Wang, J.; Yu, P.S. HashNet: Deep Learning to Hash by Continuation. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5609–5618.
52. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems 25 (NIPS 2012), Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1097–1105.
53. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity Mappings in Deep Residual Networks. In Proceedings of the 14th European Conference Computer Vision (ECCV 2016), Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; Volume 9908, pp. 630–645.
54. Sharif Razavian, A.; 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 Workshops, Columbus, OH, USA, 24–27 June 2014; pp. 806–813.
55. Olivas, E.S. *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*; IGI Global: Hershey, PA, USA, 2009.
56. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

57. Yang, L.; Qi, X.; Xing, F.; Kurc, T.; Saltz, J.; Foran, D.J. Parallel content-based sub-image retrieval using hierarchical searching. *Bioinformatics* **2013**, *30*, 996–1002. [CrossRef] [PubMed]
58. Distasi, R.; Vitulano, D.; Vitulano, S. A Hierarchical Representation for Content-based Image Retrieval. *J. Vis. Lang. Comput.* **2000**, *11*, 369–382. [CrossRef]
59. Jiang, F.; Hu, H.M.; Zheng, J.; Li, B. A hierarchal BoW for image retrieval by enhancing feature salience. *Neurocomputing* **2016**, *175*, 146–154. [CrossRef]
60. You, J.; Li, Q. On hierarchical content-based image retrieval by dynamic indexing and guided search. In Proceedings of the 2009 8th IEEE International Conference on Cognitive Informatics (ICCI'09), Hong Kong, China, 15–17 June 2009; pp. 188–195.
61. Lloyd, S. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **1982**, *28*, 129–137. [CrossRef]
62. Angelov, P. Anomalous System State Identification. U.S. Patent US9390265B2, 15 May 2012.
63. Angelov, P. *Evolving Rule-Based Models: A Tool for Design of Flexible Adaptive Systems*; Springer: Berlin/Heidelberg, Germany, 2002.
64. Angelov, P.; Sadeghi-Tehran, P.; Ramezani, R. A Real-time Approach to Autonomous Novelty Detection and Object Tracking in Video Stream. *Int. J. Intell. Syst.* **2011**, *26*, 189–205. [CrossRef]
65. Zhang, C.; Huang, L. Content-Based Image Retrieval Using Multiple Features. *J. Comput. Inf. Technol.* **2014**, *22*, 1–10. [CrossRef]
66. Wang, X.-Y.; Zhang, B.-B.; Yang, H.-Y. Content-based image retrieval by integrating color and texture features. *Multimed. Tools Appl.* **2012**, *68*, 545–569. [CrossRef]
67. Yue, J.; Li, Z.; Liu, L.; Fu, Z. Content-based image retrieval using color and texture fused features. *Math. Comput. Model. Int. J.* **2011**, *54*, 1121–1127. [CrossRef]
68. Oliva, A.; Torralba, A. Building the Gist of A Scene: The Role of Global Image Features in Recognition. *Prog. Brain Res.* **2006**, *155*, 23–36. [PubMed]
69. Huang, J.; Kumar, S.R.; Mitra, M.; Zhu, W.-J.; Zabih, R. Image indexing using color correlograms. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, USA, 17–19 June 1997; pp. 762–768.
70. Wang, J.; Yang, J.; Yu, K.; Lv, F.; Huang, T.; Gong, Y. Locality-Constrained Linear Coding For Image Classification. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 3360–3367.
71. Lazebnik, S.; Schmid, C.; Ponce, J. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; pp. 2169–2178.
72. Han, S.; Chee, L.; Chan, S.; Wilkin, P.; Remagnino, P. Deep-Plant: Plant Identification with Convolutional Neural Networks. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015.
73. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; ACM: New York, NY, USA; pp. 270–279.
74. Yu, H.; Yang, W.; Xia, G.-S.; Liu, G. A Color-Texture-Structure Descriptor for High-Resolution Satellite Image Classification. *Remote Sens.* **2016**, *8*, 259. [CrossRef]
75. Li, Y.; Tao, C.; Tan, Y. Unsupervised multilayer feature learning for satellite image scene classification. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 157–161. [CrossRef]
76. Romero, A. Unsupervised deep feature extraction for remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 1349–1362. [CrossRef]