

# Efficient Framework for Content-Based Image Retrieval using CNN Classification Scores

S. A. Angadi<sup>1</sup> and Hemavati C. Purad<sup>\*2</sup>

Submitted: 11/11/2022

Accepted: 12/02/2023

**Abstract:** Content-Based Image retrieval(CBIR) is a technique to search and retrieve similar images from large multimedia databases and an IR system is regarded as efficient if it can retrieve all the images to meet the user's needs. There are many advanced machine-learning technologies such as deep neural networks(DNN), convolutional neural networks(CNN), and transfer-learning(TL), which are gaining greater importance in image-related tasks. In this paper an efficient framework for content-based image retrieval system adapting transfer-learning on pre-trained CNNs (ResNet18, GoogLeNet, AlexNet) using query-by-image method is proposed, the method explores classification-score descriptors for IR and employ distance metrics for similarity matching. The framework prescribes transfer-learning for efficient retraining of pre-trained CNNs on small datasets chosen from the Wang database. Thirty-plus experiments are designed for finding optimal values of the hyper-parameters and exploring the suitability of six popular distance metrics namely Euclidean, seucleidean, Cityblock, Cosine, Mahalanobis, and Chebychev. After extensive experimentation, a new efficient framework for CBIR using CNN classification scores is proposed and the new framework of CBIR achieves the image retrieval accuracy of 99.45% on natural scene images of 20 classes of the Wang dataset. The experimentations show that the proposed framework is efficient for content-based image retrieval system.

**Keywords:** AlexNet, CBIR, GoogLeNet, ResNet18, Transfer Learning

## 1. Introduction

The advancement in information technology has resulted in an ever-increasing quantity of multimedia data. Automated systems for analyzing, classifying, indexing, and retrieval of multimedia data are necessary, and the available computer-assisted automatic systems are facing the challenge of a "semantic gap" in the retrieval of multimedia information. The works in image retrieval date back to the 1980s with image annotation and keyword-based querying [1] on image database management (IDBM) systems. The keyword-based search is dependent on the quality and completeness of image annotations. In order to increase the efficiency of image retrieval and to reduce the work of annotation and its related discrepancies, content-based image retrieval (CBIR) systems are introduced. But these required the process of feature extraction to understand the contents of the image queries. Such visual interactions via the query-by-visual example and query-by-subjective descriptions for CBIR are explained in [2] and CBIR systems (IBM's QBIC to Chabot) are summarized in [3], [4] to increase the search

efficiency and retrieval accuracy. But the use of visual features such as color, texture, shapes (the low-level features), etc. to retrieve relevant images, results in lack of understanding the human perceptions such as objects, events, etc. Since the performance of CBIR systems heavily depends upon the feature descriptors, matching the low-level visual contents of images to high-level understanding (Bridging the semantic gap) becomes essential to meet the user needs. The semantic gap can be reduced by creating the object ontology, using supervised and unsupervised machine learning methods, relevance feedback, generating semantic templates, using web info [5], etc., and are summarized in [6] & [7].

The advancements in machine learning techniques such as deep neural networks and convolutional neural networks (CNN) have achieved greater importance in image-related tasks with their powerful feature learning functionality and hence have received considerable attention in CBIR systems in the generation of binary codes or hash codes for retrieval. The use of deep learning to build binary codes known as deep hashing techniques, for image retrieval are discussed in [8]. The use of hash codes is best suited for increasing the search quality and the choice of network is dependent upon the retrieval type, such as object retrieval and semantic retrieval. The comprehensive survey of deep learning methods such as the evolution of deep learning methods (years 2011 to 2020), network types, types of

<sup>1</sup> Visveswaraya Technological University, Belagavi, Karnataka – 590018, INDIA, saangadi@vtu.ac.in  
ORCID ID : 0000-0001-9756-9786

<sup>2</sup> Visveswaraya Technological University, Belagavi, Karnataka – 590018, INDIA, hemacpurad@gmail.com  
ORCID ID : 0000-0001-7032-1780

\* Corresponding Author Email: hemacpurad@gmail.com

descriptors, retrieval types, etc. for CBIR, is given by [9]. The feature extraction quality is increased by the use of CNNs and hash codes. The CNNs, generally with their deeper networks yield high dimensional features. The deepest features [9], [10], [11], [12], [13] are extracted from CNNs to analyze the images in the image retrieval tasks. The applications using these deep, high dimensional features will always suffer from “Curse of Dimensionality”. In order to have dimension efficient, robust, and fast features in the retrieval of the images, the “classification scores” [14] are utilized in this work. The classification scores in this work are obtained after tuning of hyper-parameters by conducting thirty-plus experiments on three pre-trained CNNs and the best framework is recommended for efficient retrieval of images on natural scene images using 3 pre-trained CNNs.

The deep learning techniques and CNNs are heavily dependent upon availability of massive training data. The performance of CNNs is directly proportional to the size of the dataset. Accuracy will be increased with the increase in the size of the dataset. In many cases, it is difficult to create or collect training datasets that are large enough. The transfer-learning is the solution to the problem of insufficient training data of CNNs. Transfer-learning is the method to improve the performance of CNNs with small datasets by bringing the knowledge gained in one task to another task in the same (homogeneous) or different (heterogeneous) domain without redesigning the architecture from scratch and this process also consumes less time. The mechanism, performance, and applications of transfer-learning are discussed in [15]. Unlike homogeneous transfer-learning, heterogeneous transfer-learning [16] is rarely used in real-world applications because of its differing feature spaces. There are four types in transfer-learning as mentioned in [17], instances based (specific weight adjustment strategy in the same domain), mapping based (bringing knowledge to new data space), network-based (reuse the partial network that pre-trained in the source domain), and adversarial based (introduce adversarial technology to find transferable representations). In this work, network-based transfer-learning is employed which means the transfer-learning is performed on pre-trained CNN models with adaptation also referred to as refining the network. Deploying augmentation methods prevents overfitting and improves the accuracy of transfer networks by increasing the size of the dataset with augmented images. There are many augmentation methods such as rotation, flipping, zooming, brightness, contrast, translation, deformation, reflections, etc. [18]. In this proposed work, the pre-trained models ResNet18 [19], GoogLeNet [20], and AlexNet [21] are chosen for retrieval of images. The application of deep CNNs started presenting considerable improvement over older networks by having the deep CNN architecture

proposed by Alex Krizhevsky et al. [21] named AlexNet after the ImageNet Large Scale Visual Recognition Challenge (ILSVRC competition dated on September 30, 2012). Later to improve the practical applications and with the influence of increased image classification efficiency, Google come up with GoogLeNet proposed by Szegedy et al. [20] and participated in ILSVRC 2014 for classification and detection challenges. Both of the networks have comparatively fewer layers and are faster in learning large datasets and hence are used in this work.

The introduction of CNN with the residual network (ResNet) has achieved a breakthrough in image classification. Microsoft has come up with a series of deep networks as ResNets proposed in [19] to improve the accuracy in image-related tasks. The ResNet and its variants are used in various applications, such as object detection [22], image classification [23], [24], image retrieval [25], etc. The performance comparison of data classification based on modern CNNs for non public datasets (Lead isotope) is explained in [26] and mentioned that ResNet18 which has moderate layers, takes less training time, with minimum error rate, and produces the best classification results. The paper [27] has implemented five CNN architectures (Yolo-v2, Yolo-Conv, GoogLeNet, ResNet18 & ResNet50) in vehicle occupancy detection, and the performance of these models are compared and it is reported that ResNet18 has performed better than other CNNs and uses less computation time and it is further stated that the ResNet18 can be used in real-time vehicle occupancy detection. The literature explores that the transfer-learning is suitable for small datasets and transfer-net with the pre-trained ResNet achieves greater accuracy by preventing overfitting with the deeper network and compared to other pre-trained models (Yolo-v2, Yolo-Conv, ZFNet, VGG16, VGG19, DenseNet, AlexNet, GoogLeNet, ResNet101, ResNet50, ResNet40, ResNet25, ResNet10, ResNet7) and ResNet18 is feasible with its moderate layers and yields better accuracy [19].

The pre-trained models in transfer-learning, extract high-level features and ResNet18 [19], GoogLeNet [20], and AlexNet [21] with relatively shallow networks, yields better accuracy, hence in this work image retrieval system based on the transfer-learning with ResNet18, GoogLeNet, and AlexNet using classification scores are explored and an efficient framework is proposed. The contributions of this paper can be summarized as below,

- i) A new framework using transfer-learning for CBIR using classification scores of CNN
- ii) Efficient training and retrieval using tuned hyper-parameters of CNN that employs transfer-learning technique using classification scores from Softmax layer and

iii) Experimentation leading to a new and efficient framework

The framework aims at providing an efficient content-based image retrieval system and the core of this framework is to provide dimensionally efficient image descriptors and suitable pre-trained CNN for IR and the suitable distance metric to find similar images using classification scores from CNN.

This work involved conducting experiments to ascertain the optimal values for the hyper-parameters of pre-trained CNN's, and optimal distance measures for content-based image retrieval. The experimentation on hyper-parameters, and distance measures for the efficient framework of CBIR, involved designing 30 experiments for hyper-parameters optimization and working with six different distance measures namely Euclidean, seclidean, Cityblock, Cosine, Mahalanobis, and Chebychev. The experimentations explored one of the two optimizer algorithms namely, Adaptive Moment Estimation (Adam) and Stochastic Gradient Descent with Momentum (sgdm), various values of Mini Batch Size (10), Maximum Epoch (3), Initial Learning Rate (4). This resulted in arriving at the optimal values for these hyper-parameters. Further, the different previously named distance measures were used to verify the retrieval accuracy of the system and it is found that Cosine similarity gives the best results of 99.25% for the CBIR system using classification scores of the CNNs (transfer-learning networks).

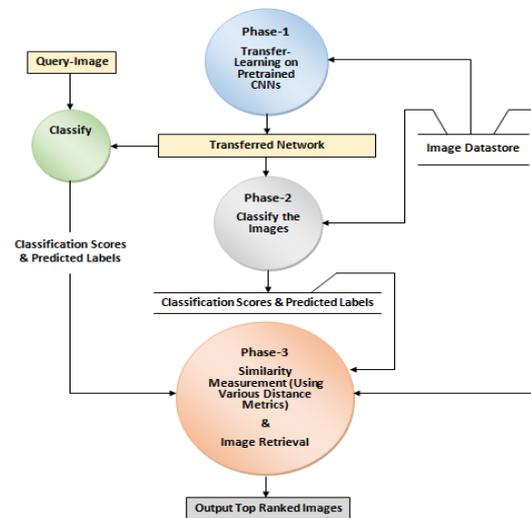
The extensive experimentation on three pre-trained CNNs has led to the proposal of an efficient framework of CBIR which consists of a transfer-net of ResNet18, with the optimized values of hyper-parameters having Adam optimizer, with a mini-batch size(MBS) of 32 and max epochs(ME) of 10 and initial learning rate(ILR) of 1e-4. Cosine similarity for matching and retrieval of images is the most efficient measure. And the recommended framework of CBIR achieves recall accuracy of 99.45% on natural scene images of 20 classes of the Wang dataset. These results are promising and present a good approach for Content-Based Image Retrieval.

The paper is organized into 5 sections, section 2 explains the proposed methodology and section 3 describes the experimentation. The results and recommendations are presented in section 4 and finally, section 5 brings up the conclusion.

## 2. Proposed Method

The proposed method devises the framework for CBIR and is comprised of three phases. Phase one focuses on transfer-learning using pre-trained CNNs (ResNet18, GoogLeNet, and AlexNet), phase two focuses on classifying the images and obtaining the classification

scores, and phase three on similarity measurement and image retrieval using the query-by-image method and various distance metrics. Details of the three phases are described in the following subsections and the data flow diagram of the proposed system with three phases is shown in Figure 1.



**Fig.1.** Framework of proposed CBIR system using classification scores

The framework in Figure 1 presents the flow of data between the phases. In phase 1 transfer-learning is performed separately on each dataset (subsets obtained from the Wang database) with ResNet18. And the transfer-learning is also done on GoogLeNet and AlexNet using wang20 and wangO20 datasets. Finally, the validated transfer-nets are saved for further use. In phase 2 all the images in datasets are classified using transfer-net and the resultant classification scores and predicted labels are saved in the datastore. In the third phase, the query-by-image method is employed, classification scores are obtained and the label is predicted by classifying the query-image using transfer-net, and image retrieval is performed through similarity measurement by employing the distance metrics between classification scores of query-image to the classification scores pre-computed for training/validation images. Finally, the required numbers of top-ranked images are retrieved. The detailed process is explained in the following subsections.

### 2.1. Transfer-Learning with Pre-trained CNNs (ResNet18, GoogLeNet, & AlexNet)-(Phase-1)

The transfer-learning is applied by deploying 18-layers deep pre-trained CNN the ResNet18, 22 layers deep GoogLeNet, and 8 layers deep AlexNet, to classify datasets of images selected from the Wang database also known as Corel database [33]. The transfer-learning on ResNet18 is employed for each dataset. And the GoogLeNet and AlexNet have been trained on wang20 and wangO20 datasets (the datasets are explained in section 3).

Once the transferred networks are built, the models are validated by using input images kept for validation. The trained and validated transferred networks are saved in the data store for later use in the next phase. The methodology of transfer-learning is given in the Algorithm-1.

#### Algorithm 1: TRANSFER-LEARNING ON PRE-TRAINED\_CNN

Input: Image Datastore 'Ids', Pretrained\_CNN 'P'

Output: Transferred Network 'T<sub>n</sub>' obtained from transfer learning on the Pre-trained\_CNN 'P'

Begin

Step 1: Split the image datastore 'Ids' into training & validation sets

Step 2: Apply pre-processing on training & validation sets

Step 3: Create new learnable & new class layers for the pre-trained\_CNN 'P'

Step 4: Replace the fully connected & classification output layers of the

Pre-trained\_CNN 'P' with the newly created layers

Step 5: Find optimal values for hyper-parameters of the Pretrained\_CNN 'P'

Step 6: Retrain & validate the modified Pretrained\_CNN on the dataset

Step 7: Store the transferred network 'T<sub>n</sub>' of the pre-trained\_CNN 'P'

End

The Algorithm-1 explains the process of transfer-learning on pre-trained CNNs (Resnet18, GoogLeNet, and AlexNet) in the first phase. The datastore images are partitioned into training and validation sets to train and validate the pre-trained CNNs for the new dataset. The pre-processing is performed over the training dataset before feeding to the network. The input images are resized to accommodate the size of the input layer of the pre-trained CNNs (for ResNet18 & GoogLeNet it is 224-by-224-by-3 & for AlexNet it is 227-by-227-by-3). The image augmentation (pre-processing-involves flipping the images along the vertical axis, randomly translating the pixels horizontally and vertically, etc.) is performed to make transfer-net to be invariant to changes in the images (translation, viewpoint, size, or illumination etc.) in order to learn the relevant pattern and boost overall performance. The two new layers ('newLearnableLayer' and 'newClassLayer') are created to learn and classify the images and to output the labels of the new dataset. The final layers, the fully connected layer ('fc1000' of ResNet18, 'loss3-classifier' of GoogLeNet, and 'fc8' of Alexnet) is replaced with the 'new\_fc' to reflect the

features of the new small dataset and the classification output layer ("classificationLayer\_Predictions" of ResNet18 and 'output' of GoogLeNet and AlexNet), is replaced with 'new\_classoutput' to output the predicted class labels of the new datasets. The hyper-parameters are tuned to obtain optimal values and finally transferred network is obtained by retraining and validating the modified pre-trained CNN and stored to use in the next phase.

The hyper-parameters namely, the optimizer algorithm (OA), mini-batch size (MBS), max epochs (ME), initial learning rate (ILR), validation frequency (VF) are normally used in the training process of transfer-learning. The optimal values set to these parameters will play a crucial role in obtaining an effective transferred network. The features obtained from an effective transferred network can efficiently retrieve the images. And hence the experimentations are conducted to fine-tune the values of these hyper-parameters. And so-obtained transfer-net is used in the subsequent stage.

#### 2.2 Classify the Images (Phase-2)

Transfer-learning phase is followed by the classification phase, the second phase of the proposed system. The classification scores are extracted by classifying all the images of the dataset using transfer-net in order to index the images for retrieval in the subsequent phase. All the images in the database are fed to the transfer-net and the Softmax layer of the transfer-net classifies the images and obtains the classification scores ( $\alpha$ ) and the 'new\_classoutput' layer outputs the classification label predictions ( $\beta$ ) of all the images of the database and are stored in the datastore for further use in the next stage, and the process is presented in Algorithm-2. The fully connected layer and classification layers contain information on how to combine the features that the network extracts into class probabilities (classification scores) & predicted labels.

#### Algorithm 2: CLASSIFY\_IMAGES

Input: Image Datastore 'Ids', Transferred Network 'T<sub>n</sub>'

Output: Classification scores ' $\alpha$ ' & Predicted labels ' $\beta$ '

Begin

Step 1: Input the image datastore 'Ids' to the transferred network 'T<sub>n</sub>'

Step 2: Apply Softmax function 'S' of the transferred network 'T<sub>n</sub>' to classify the images of the image datastore 'Ids' to obtain classification scores ' $\alpha$ ' & Predicted label ' $\beta$ '

Step 3: Store the Classification scores ' $\alpha$ ' & Predicted label ' $\beta$ ' for each image 'I' of the image datastore 'Ids'

End

**CLASSIFICATION SCORES.** Normally when CNN techniques are used for image related tasks, the deepest features are obtained from different layers of the network to analyze the images for classification. Further various descriptors [8], [9] such as hash codes, binary descriptors, real-valued descriptors, aggregated descriptors (fusion of features at different layers of the network(s)), etc. are used in the retrieval of images. In this work, the dimensionally efficient, “classification scores” are utilized as the descriptors for the retrieval of images. The “classification scores” form the vector of probabilities of the image being classified as belonging to each class and such vectors are generated for each image at the Softmax layer (the last activation function) [14], and are extracted. The Softmax activation function scales (normalizes the output of the network to a probability distribution over predicted output classes) “numbers/logits” into probabilities. In mathematics ‘logit’ is a function that maps probabilities (0, 1) to real numbers in  $(-\infty, +\infty)$ . The output of a Softmax is a vector with probabilities of each possible outcome. The probabilities in vector sums to one for all the possible image classes. Mathematically Softmax is defined as in equation (1).

$$S(y)_i = \frac{\exp(y_i)}{\sum_{j=1}^m \exp(y_j)} \quad (1)$$

Where ‘y’ is an input vector to a Softmax function, S. It consist of ‘m’ elements for ‘m’ image categories (possible outcomes). ‘y<sub>i</sub>’ is the i<sup>th</sup> element of the input vector. It can take any value between  $-\infty$  to  $+\infty$ . The ‘exp(y<sub>i</sub>)’ is the standard exponential function applied to ‘y<sub>i</sub>’. The result of ‘exp(y<sub>i</sub>)’ is small value (close to ‘0’ but not ‘0’) if y<sub>i</sub><0 & a large value if ‘y<sub>i</sub>’ is large. And the term ‘ $\sum_{j=1}^m \exp(y_j)$ ’ is a normalization term. It ensures that the values of output vector ‘S(y)<sub>i</sub>’ sums to ‘1’ for the j<sup>th</sup> class & each of them is in the range ‘0’ and ‘1’ which makes up a valid probability distribution. And ‘m’ indicates number of image classes (possible outcomes).

When images are classified, the ‘m’ numbers of scores are generated for each image in the dataset, where ‘m’ indicates the number of classes available in that dataset. Hence for the ‘m’ number of classes, the dimension of the feature vector containing classification scores for each image contains ‘m’ values. And these classification scores (α) of the images are utilized in the image retrieval phase.

$$\text{Let, } f(I) = \sum_{i=1}^n \left( \sum_{j=i}^m \alpha_{I_j^i} \right) \quad (2)$$

$$\text{And let, } f(Q) = \sum_{j=1}^m \alpha_{Q_j} \quad (3)$$

Where f(I) & f(Q) in equation (2) & (3) indicates the classification output functions for database images &

Query-image respectively. And the  $\alpha_{I_j^i} = \{\alpha_{I_1^i}, \alpha_{I_2^i}, \dots, \alpha_{I_m^i}; \alpha_{I_1^j}, \alpha_{I_2^j}, \dots, \alpha_{I_m^j}; \dots; \alpha_{I_1^n}, \dots, \alpha_{I_m^n}\}$  is the vector of ‘m’ classification scores of i<sup>th</sup> input image of the j<sup>th</sup> class in the dataset and the  $\alpha_{Q_j} = \{\alpha_{Q_1}, \alpha_{Q_2}, \dots, \alpha_{Q_m}\}$  is the vector of ‘m’ classification scores for query-image (Q) respectively. There are 5 datasets namely wang5, wang10, wang15, wang20, and wangO20, and ‘n’ is the total number of images, and ‘m’ is the number of classes in each dataset. The dataset wang5 has total n=500 images, m=5 classes, wang10 has n=1000 images, m=10 classes, wang15 has n=1500 images, m=15 classes, wang20 has n=2000 images, m=20 classes respectively and each class in the dataset contains a maximum of 100 images (MAX\_CLASS\_SIZE). The details of datasets are provided in Table 3 of section 3. Once the classification scores are obtained the image retrieval is performed and is explained in the next section.

### 2.3 Similarity Measurement & Image Retrieval (Phase-3)

Image features and distance metrics will play a vital role in similarity matching and retrieval of images. The image retrieval is performed by utilizing classification scores and the similarity measurement is carried out by employing the most popular and efficient distance metrics, the Euclidean, Standardized Euclidean (seuclidean), Cityblock, Cosine, Mahalanobis and Chebychev, to analyze the behavior of metrics in the retrieval of images using classification score descriptors on three pre-trained CNNs. These distance metrics have already proven their utility in the retrieval of images [28], [29] and hence these metrics are chosen for experimentation to easily discriminate the efficacy of the proposed image descriptor in the retrieval of images and identify the best distance metric for the framework of CBIR using query-by-image.

**QUERY-BY-IMAGE.** Given a query-image Q, the required ‘L’ number ( $\leq$ Max\_Class\_Size) of images are retrieved from the top-ranked (indexed) images by processing the query-image as illustrated in Algorithm-3.

#### Algorithm 3: SIMILARITY MEASUREMENT & IMAGE RETRIEVAL

Input: Query Image ‘Q’, Transferred network ‘T<sub>n</sub>’, Distance metric ‘D<sub>m</sub>’, Classification scores ‘ $\alpha_{I_m^n}$ ’ of all ‘m’ classes of ‘n’ number of images of image datastore ‘Ids’

Output: Retrieve & display top ‘L’ images ( $\leq$ Max\_Class\_Size) from the set of sorted retrieved images

Begin

Step 1: Input the Query image ‘Q’ to the transferred network ‘T<sub>n</sub>’

Step 2: Apply Softmax function ‘S’ of the transferred network ‘T<sub>n</sub>’ to classify & obtain classification scores ‘ $\alpha_{Q_m}$ ’ of the query image ‘Q’

Step 3: Compute the pair-wise distance between classification scores ‘ $\alpha_{Q_m}$ ’ of query image ‘Q’ to classification scores ‘ $\alpha_{I_m^n}$ ’ of each image ‘I’ of the datastore ‘Ids’ by applying distance metric ‘D<sub>m</sub>’

Step 4: Sort the vector of resultant distances in an ascending order along with the corresponding predicted labels ‘ $\beta$ ’ and images ‘I’ of image datastore ‘Ids’

Step 5: Finally retrieve & display top ‘L’ images ( $\leq$ Max\_Class\_Size) from the set of sorted retrieved images

End

The Algorithm-3 gives the process of similarity measurement & image retrieval in this phase. In order to retrieve the images, the vector of classification scores  $\alpha_{Q_j}$

= {  $\alpha_{Q_1}, \alpha_{Q_2}, \dots, \alpha_{Q_m}$  } are obtained from classifying the query-image (Q) using transferred network. And the vector of classification scores  $\alpha_{I_j^i} = \{ \alpha_{I_1^i}, \alpha_{I_2^i}, \dots, \alpha_{I_m^i}; \alpha_{I_1^2}, \alpha_{I_2^2}, \dots, \alpha_{I_m^2}; \dots; \alpha_{I_1^n}, \dots, \alpha_{I_m^n} \}$  of each image I<sup>i</sup> of the dataset which were stored in the form of matrix, is retrieved from the datastore for similarity measurement. The distance metric is employed, the pair-wise distance between the two sets of vector of classification scores ( $\alpha_{Q_j}$  to  $\alpha_{I_j^i}$ ) are computed to measure the similarity between query image to the database images using various distance metrics as in equation (4) to (9) of Table 1. The application of distance metrics will result in the vector of distances. The length of the resultant vector corresponds to the number of images in the database. The resultant vector of distances is sorted in ascending order and ranked as 1 to N number of images in the database and their respective database images are indexed. Finally, the specified number ‘L’ of images ( $\leq$ Max\_Class\_Size) are retrieved from the set of sorted retrieved images and displayed.

**Table 1.** The distance metrics used in retrieval of images

Sl. No	Distance Metric	Equation	Description	Characteristics	Eq. No.
1	Euclidean	$d(\alpha_Q, \alpha_I) = \sqrt{\sum_{i=1}^n \sum_{j=1}^m (\alpha_{I_j^i} - \alpha_{Q_j})^2}$	Where ‘d’ is Euclidean distance, and ‘ $\alpha_Q$ ’ is a classification-scores( $\alpha$ ) for Query Image(Q) and ‘ $\alpha_I$ ’ is Classification-scores( $\alpha$ ) of all the database images(I <sup>i</sup> )	This is the most common distance measure. It is not scale-invariant so is necessary to normalize the data before using it. It works great with low dimensional data & when the magnitude of the vectors is important to be measured [28], [29], [30]. The proposed image descriptor is dimension efficient and finds the pair-wise distance between the vectors of classification score descriptors (coordinates) of query-image to the database images.	(4)
2	Standardized Euclidean	$d_s(\alpha_Q, \alpha_I) = \sqrt{\sum_{i=1}^n \sum_{j=1}^m \frac{1}{s_j^2} (\alpha_{I_j^i} - \alpha_{Q_j})^2}$	Where ‘d <sub>s</sub> ’ is seclidean (Standardized Euclidean) distance and S <sub>j</sub> is the sample standard deviation of the j <sup>th</sup> variable.	This is Standardized Euclidean distance metric where in which each coordinate difference between the vector of classification scores of query image to database images is scaled by dividing by the corresponding element of the standard deviation. And continues to find such distance for all the images of the database and finally results in seclidean distances.	(5)

3	Cityblock	$d_c = \sum_{i=1}^n \sum_{j=1}^m  \alpha_{I_j^i} - \alpha_{Q_j} $	Where 'd <sub>c</sub> ' is the city-block distance.	This distance metric is also called the Manhattan or taxicab distance. This finds the distance between two vectors only if they would move right angles. And suits for high dimensional data [28], [29], & [31]. Despite using dimensionally-efficient image descriptors, the Cityblock metric performs well. The Cityblock distance computes the absolute difference between the classification scores (probabilities or coordinates) of the pair of vectors belonging to the query-image & the database image. And finally, obtains the Cityblock distance of each image (nothing but a vector of Cityblock distances).	(6)
4	Cosine	Cosine distance=1-cosine similarity	Where cosine similarity= $\frac{\sum_{i=1}^n \sum_{j=1}^m \alpha_{I_j^i} \alpha_{Q_j}}{\sqrt{\sum_{i=1}^n \sum_{j=1}^m (\alpha_{I_j^i})^2} \sqrt{\sum_{j=1}^m (\alpha_{Q_j})^2}}$	Cosine similarity has often been used as a way to counteract Euclidean distance's problem with high dimensionality. The cosine similarity is simply the cosine of the angle between two vectors. The cosine is often used when there is high dimensional data and when the magnitude of the vectors is not of importance means in practice the differences in values are not fully taken into account [29], [32]. In this work the cosine of the angle between the vector of classification scores of query image and database image is performed. The smaller the angle, higher the cosine similarity. And the images which placed far apart by the Euclidean distance can be brought close with the smaller angle between them with the cosine distance metric.	(7)
5	Mahalano bis	$d_M(x, y) = \sqrt{(x - y)^T S^{-1} (x - y)}$	Where 'd <sub>M</sub> ' is a Mahalanobis distance, S is covariance matrix, and $x = \sum_{i=1}^n \sum_{j=1}^m \alpha_{I_j^i}$ and $y = \sum_{j=1}^m \alpha_{Q_j}$	Mahalanobis distance is an effective multivariate distance metric that measures the distance between a point and a distribution. It is an extremely useful metric having, excellent applications in multivariate anomaly detection, classification on highly imbalanced datasets and one-class classification [29], [32]. It computes the distance according to the statistical variation of each component using the covariance matrix of the vector of classification	(8)

				scores of the database images and results in the vector of Mahalanobis distances.	
6	Chebychev	$d_{cheb}(x, y) = \max_i( x_i - y_i )$	Where ' $d_{cheb}$ ' is a Chebychev distance, and $x = \sum_{i=1}^n \sum_{j=1}^m \alpha_{ij}$ and $y = \sum_{j=1}^m \alpha_{Qj}$	Chebychev distance is defined as the greatest of difference between two vectors along any coordinate dimension. In other words, it is simply the maximum distance along one axis. Due to its nature, it is often referred to as Chessboard distance [28],[ 29]. In this work the vector of Chebychev distances, is obtained by computing the greatest of difference between query-image's vectors of classification scores to the vector of classification scores of database images.	(9)

The vector of 'm' classification scores are the probabilities of the classified image belonging to the corresponding class. The distance metrics, Euclidean, seuclidean, City-block, Cosine, Mahalanobis and Chebychev defines a distance between the two (query image and database

image) vectors of classification scores ( $\alpha_{Qj}$  to  $\alpha_{ij}$ ). The application of these distance metrics will yield the vector of single distances and are as shown in the instance given in Table 2.

**Table 2.** The instance of classification scores obtained for the database images, the query image-BUS, & the sample of resultant vector of distances of Euclidean distance metric on wang5 dataset

Sl. No	The sample classification scores obtained for wang5 dataset are,				
1	{0.99998260	1.5573902e-05	8.7092258e-07	7.0329541e-08	9.3770132e-07;
2	0.99992704	3.1920321e-05	2.6577220e-05	1.3114803e-06	1.3145765e-05;
3	0.99999452	3.8131013e-06	1.2904632e-06	7.5322347e-08	1.8233963e-07;
4	1	3.5422183e-08	2.2425489e-08	2.4634079e-09	1.9584863e-09;
5	0.99999750	1.4847376e-06	1.0171109e-06	2.4384854e-09	2.2980675e-08; & soon 500 rows of classification scores}
	The classification scores obtained for query-image: 'Bus' are				
6	{0042403e-08	2.0793327e-06	0.99999344	8.8135881e-09	4.4501253e-06}
	The sample Euclidean distances obtained are,				
7	{2.3090647e-11;	1.0066730e-06;	1.0680699e-06;	1.2641023e-06;	1.3906412e-06; & soon 500 distance values}

### 3. Experimentation

This section discusses the experiments conducted in order to evaluate the performance of the proposed methodology and explores the use of various distance metrics. Initially a brief description of the datasets, the evaluation metrics and subsequently experimentations conducted and the observations are presented.

**DATASETS.** The datasets are obtained from the Wang database (also known as Corel-database). This database comes in different flavors Corel-1k, Corel-5K & Corel-10K [33, 34]. Each of which contains one, five & ten

thousand images categorized into ten, fifty and one-hundred classes respectively, whereas, each class contains one hundred images (MAX\_CLASS\_SIZE). These datasets contain diverse (natural) scenes such as, Africans, beach, monuments, horses, cars, dinosaurs, mountains, flowers etc. In order to evaluate the retrieval efficacy of the transfer-net with diversity of scenes the Wang database is chosen in this proposed work. Five small datasets containing 5, 10, 15 & 20 classes are formed from Corel-1K and Corel-10K databases. The proposed method is validated on these datasets. The datasets formed are detailed in Table 3.

**Table 3.** The small datasets formed of 5, 10, 15 and 20 classes from the Wang database

Sl. No	Dataset Names Chosen	No. of Classes(m)	No of Images Available in the dataset(n)	Classes Chosen
1	wang5	5	500	{Beach, Monument, Bus, Butterfly, Car}
2	wang10	10	1000	{Beach, Monument, Bus, Dinosaur, Elephant, Food, Horse, Mountain, Rose, Sports}
3	wang15	15	1500	{Beach, Monument, Bus, Butterfly, Car, Dinosaur, Elephant, Food, Forest, Horse, Mountain, Planet, Rose, Sports, Sunrise}
4	wang20	20	2000	{Africans, Beach, Monument, Bus, Butterfly, Car, Dinosaur, Elephant, Food, Forest, Horse, Map, Mountain, , Planet, Rose, Space, Sports, Sunrise, Text, Yacht}
5	wangO20 (O - for Other)	20	2000	{Africans, Beach, Bus, Butterfly, Church, Dinosaur, Flag, Forestlife, Logo, Medication, Monument2, Nature, Planet, Rose, Space, Testwork, Traintrack, Vibrantlighting, Waterfalls}

**EVALUATION METRICS.** The performance of the proposed system is measured by validation accuracy and recall rate. The validation accuracy is computed as in equation (10) on part of the dataset which is not used for training but used for validating or testing the generalization ability of (transfer-net in order to validate) the model during training of transfer-learning. And predictions made on test set is used for finding the validation accuracy. The validation accuracy of transferred network is calculated for new classes and the equation is as follows,

$$VA = \frac{C}{T} \quad (10)$$

Where ‘VA’ indicates validation accuracy, ‘C’ indicates the correctly classified labels of the test query-images and ‘T’ is the total number of labels of the test query-images.

In order to evaluate the retrieval accuracy, the query images are chosen randomly from each class and recall rate of each distance metric, on dataset of 20 classes is reported. The recall (also known as sensitivity) as defined in equation (11) is the ratio of the total number of relevant instances to the total images in each class and is given as follows,

$$Recall = \frac{\{Relevant\ Images\} \cap \{Retrieved\ Images\}}{\{Relevant\ Images\}} \quad (11)$$

Where the term  $(\{\text{Relevant images}\} \cap \{\text{Retrieved Images}\})$  in the numerator, indicates the number of relevant images retrieved out of the total number of relevant images in the database. The term  $(\{\text{Relevant Images}\})$  in the denominator indicates the total number of relevant images in the database. The implementation details of the framework are given in the following subsection.

**IMPLEMENTATION.** The proposed methodology is implemented using Matlab 2020b. The three deep pre-trained CNNs, the Restnet18, GoogLeNet, and AlexNet, which have the shallow networks, and have reduced complexity, memory and execute in less time, are chosen for retrieval of images. In order to obtain the effective transfer-net, which in turn yields efficient features for retrieval of images, the thirty experiments (E1 to E30) are designed for ResNet18 to optimize the values of hyper-parameters for retrieval of the images on the given dataset. Once the optimal values are obtained for ResNet18, those experimental values, are used on GoogLeNet and AlexNet to check whether those experimental values also give better performance on these pre-trained CNNs. And also additional experiments Eg1, Eg2 of Table 5, and Ea1, Ea2, Ea3 of Table 6 are conducted to optimize the values of the parameters for GoogLeNet and AlexNets respectively.

The hyper-parameters which have greater impact on model capacity such as, the learning rate, optimization algorithm, Mini Batch Size, Max Epochs are chosen for optimization during experimentation of transfer-learning, to minimize the error rate and maximize the accuracy. The parameter, optimizer-algorithm is assigned with two values namely, the Adaptive moment estimation (Adam) [35] and the Stochastic Gradient Descent with Momentum (sgdm/SGDM) [36]. These two are most widely used and are practical optimizers to minimize the error rate in the network. To increase the learning rate in the early layers of transfer-net, the learn rate factor of weights and bias are set to the high value that is to the value ten. And to keep minimal cost for revalidation of the transfer-net for new datasets the validation frequency is fixed to the value three. The experimentations conducted by deploying ResNet18, GoogLeNet, and AlexNet are tabulated in Table 4, Table 5 and Table 6 respectively. And the abbreviations used in the experimentations are Adam-Adaptive moment estimation; sgdm-Stochastic Gradient Descent with momentum; OA-Optimizer Algorithm; MBS-Mini Batch Size; ME-Max Epochs; ILR-Initial Learning Rate; VA-Validation Accuracy.

**Table 4.** The 30 experiments to optimize hyper-parameters for transfer-learning of ResNet18

Exp. No.	OA	MBS	ME	ILR	VA(in%) of wang5 dataset	VA(in%) of wang10 dataset	VA(in%) of wang15 dataset	VA(in%) of wang20 dataset	Mean VA(in%) of all datasets
E1	Adam	8	6	1e-4	98.68	96.87	96.51	93.63	96.42
E2	Adam	10	6	1e-4	97.35	97.49	98.69	96.73	97.56
E3	Adam	10	8	1e-4	97.35	97.18	96.94	94.77	96.56
E4	Adam	10	10	1e-4	98.68	96.55	97.38	94.28	96.72
E5	Adam	16	8	1e-4	100	99.06	97.38	97.06	98.37
E6	Adam	32	8	1e-4	99.34	98.43	99.34	98.69	98.95
E7	Adam	32	10	1e-4	97.35	98.75	98.51	97.39	98
E8	Adam	32	10	1e-3	90.07	94.67	91.70	96.90	93.33
E9	Adam	32	10	1e-5	94.04	97.49	98.91	96.24	96.67
E10	Adam	32	10	1e-6	45.03	52.98	40.34	39.54	44.47
E11	Adam	38	10	1e-4	100	98.75	98.91	97.06	98.68

E12	Adam	40	10	1e-3	90.73	87.46	92.36	90.85	90.35
E13	Adam	40	10	1e-4	98.01	98.75	98.47	98.04	98.31
E14	Adam	40	10	1e-5	98.01	98.12	97.38	93.95	96.86
E15	Adam	50	10	1e-4	98.68	97.49	98.03	97.39	97.89
E16	Adam	100	10	1e-4	Error	Error	Error	Error	!
E17	Sgdm	8	6	1e-4	98.01	98.12	98.69	96.90	97.93
E18	Sgdm	8	8	1e-4	96.03	98.75	98.69	96.57	97.51
E19	Sgdm	10	6	1e-4	95.36	95.30	96.72	95.92	95.82
E20	Sgdm	10	8	1e-4	98.68	97.18	97.82	95.75	97.35
E21	Sgdm	10	10	1e-3	96.69	97.18	99.34	94.93	97.03
E22	Sgdm	10	10	1e-4	98.01	97.49	98.47	98.04	98.00
E23	Sgdm	10	10	1e-6	26.49	36.05	16.16	11.27	22.49
E24	Sgdm	16	8	1e-4	94.70	97.81	97.82	96.90	96.80
E25	Sgdm	18	8	1e-4	97.35	98.12	98.47	95.10	97.26
E26	Sgdm	18	10	1e-4	96.03	96.87	98.69	95.92	96.87
E27	Sgdm	20	10	1e-4	98.01	97.49	98.91	96.90	97.82
E28	Sgdm	32	8	1e-4	96.69	97.81	95.20	92.65	95.58
E29	Sgdm	32	10	1e-4	98.98	97.49	97.82	95.75	97.51
E30	Sgdm	100	10	1e-6	error	Error	Error	Error	!

**Table 5.** The 7 experiments and the results obtained on wang20 dataset using GoogLeNet

Exp. No.	OA	MBS	ME	ILR	VA(in%)of wang20 dataset
E5	Adam	16	8	1e-4	92.54
E6	Adam	32	8	1e-4	95.56
E11	Adam	38	10	1e-4	95.71
E17	Sgdm	8	6	1e-4	94.44

E22	Sgdm	10	10	1e-4	96.19
Eg1	Adam	50	6	1e-4	Error
Eg2	Sgdm	50	6	1e-4	92.70

**Table 6.** The 8 experiments and the results obtained on wang20 dataset using AlexNet

Exp. No.	OA	MBS	ME	ILR	VA(in%)of wang20 dataset
E5	Adam	16	8	1e-4	Error
E6	Adam	32	8	1e-4	Error
E11	Adam	38	10	1e-4	Error
E17	Sgdm	8	6	1e-4	93.97
E22	Sgdm	10	10	1e-4	95.56
Ea1	Sgdm	38	8	1e-4	95.40
Ea2	Sgdm	40	6	1e-4	Error
Ea3	Sgdm	20	6	1e-4	Error

The colors Green, Yellow, and Red in the above tables will indicate most efficient, least efficient and error caused experiments respectively.

**OBSERVATIONS ON EXPERIMENTATION CONDUCTED ON TUNING OF HYPER-PARAMETERS OF THE PRE-TRAINED CNNs.** Tuning the values of the hyper-parameters during transfer-learning is performed by conducting 30, 7, and 8 experimentations on ResNet18, GoogLeNet, and AlexNet respectively to validate the performance of the transfer-net and to obtain the best model for retrieval of images. The best experiments of ResNet18 are chosen and conducted to prove their applicability in the efficient transfer-learning on GoogLeNet and AlexNet. The observations with respect to the hyper-parameters are discussed below.

**1. Optimizer Algorithm(OA).** The experiments of Table 4, Table 5 and Table 6 are observed for the efficiency of the optimizers. The Adam optimizer yields better results in combination with larger values of MBS as in experiments E6 (OA-Adam, MBS-32) & E13 (OA-Adam, MBS-40) on ResNet18, E11 (OA-Adam, MBS-38) on GoogLeNet and due to the requirement of larger memory AlexNet gave an error to handle larger MBS with any (adam/sgdm) optimizer. The sgdm optimizer better suited for all the 3

CNNs and is efficient despite using small MBS (as in E22 (OA-sgdm, MBS-10) of Table 4, Table 5 & Table 6).

**2. Initial Learning Rate(ILR).** The experiments E8, E12 and E21 on ResNet18 are assigned with the value 1e-3 to the ILR, and has obtained good results on using both the optimizers. But the small value (1e-6) to the ILR, greatly degrades the performance as observed by the experiments E10 and E23 on Resnet18. The value 1e-4 is found optimal for transfer-learning on all the 3 networks. Choosing the proper ILR is challenging, if too small value is chosen it results in too long training process and efficiency is less. And if too large value is chosen, it results in unstable training process. Choosing moderate ILR gives efficient transfer network such as ILR with the value 1e-4.

**4. Mini Batch Size(MBS).** The largest MBS value of 100 is chosen in experiments E16 and E30, which is the size of each class in the datasets. Choosing this value caused an error during training as GPU is low on memory. It is observed that larger main memory is required to process the large MBS of images. The MBS is an important hyper-parameter that influences the dynamics of the learning algorithm. Smaller batch sizes make it easier to fit one batch worth of training data in memory (when using GPU) and are protracted. Hence for this reason training process

caused an error during handling MBS=100. In spite of using small MBS and ME values, the transfer-learning on GoogLeNet and AlexNet is very protracted. The larger value of MBS on these networks starves for memory & causes an error during training. And hence the moderate values are required for MBS (32 for ResNet18, 10 for GoogLeNet & AlexNet) in order to have an efficient transfer-learning on all the three CNNs.

**5. Maximum Epochs (ME).** Choosing the large values to “maximum epoch” is time consuming and it is protracted for large datasets and performance will be saturated for the large values of ME. For this purpose the ME values are kept small (6 to 10) for all the experiments (on ResNet18, GoogLeNet, AlexNet). The value 8 to the ME on all the three CNNs results in efficient transfer-learning.

Hence the optimal values of the hyper-parameters will influence in yielding the efficient transferred network. And the observation on pre-trained CNNs during transfer-learning is given below.

**ResNet18.** Consumes less time during transfer-learning on very small datasets (of 5 & 10 classes). Adam and/or sgd both optimizers are suitable to use with ResNet18. The experiment E6 (OA-Adam, MBS-32, ME-8, ILR=1e-4) &

E22 (OA-sgdm, MBS-10, ME-10, ILR=1e-4) both are best for the ResNet18.

**GoogLeNet.** Is very protracted to transfer learn on datasets of 20 classes. The sgd optimizer is suitable to use with GoogLeNet. It requires larger main memory to handle larger MBS. The experiment E22 (OA-sgdm, MBS-10, ME-10, ILR=1e-4) is suitable to use with this network.

**AlexNet.** Is also very time consuming to transfer-learn on 20 classes. The Adam optimizer is not suitable to use with. It works on small values of MBS and the experiment E22 (OA-sgdm, MBS-10, ME-10, ILR=1e-4) is suitable for AlexNet.

The selection of optimal values of the hyper-parameters that is the optimizer, MBS, ME and ILR values greatly impacts on the performance of transfer-learning on the pre-trained networks. And the experimented values of hyper-parameters have been shown to have considerable effect on the performance of the proposed retrieval system. The various pre-trained CNNs and hyper-parameters selected and fine-tuning of those hyper-parameters found in state-of-the-art are tabulated in Table 7.

**Table 7.** The fine tuning of hyper-parameters found in state-of-the-art

Paper	Pretrained CNNs Used	Hyper-parameters and values tested	Dataset	Remarks
Li et al. (2020) [37]	ResNet101-V2 DenseNet MobileNet	Learning Rate(LR) = (0.1, 0.05, 0.01, 0.005, 0.001, 0.0001) ; Momentum = (0.9, 0.99, 0.95, 0.9, 0.8, 0.0) ; Weight Decay = (0.0, 0.0001, 0.0005, 0.001); Batch Size = 256 (8GPUs*batch size 32); Epochs = 60	ImageNet, Places-365, iNaturalist  (Aircrafts, Birds, Caltech, Cars, Dogs, Flower, Indoor)	Extensive empirical evaluation for fine-tuning of hyper-parameters, and similarity matching using Earth Mover’s distance is performed and concluded that choice of hyper-parameters (especially effective learning rate) are dependent on dataset & also on similarity between source & target domains & also that the momentum can affect the performance of Transfer Learning. No particular value is concluded for any of the selected hyper-parameters.
Zhou et al. (2020) [38]	ResNet18 ResNet34 ResNet50 ResNet101	Mini-Batch Size = (10, 20, 30, 40, 50, 60, 70, 80, 90, 100); learning rate = (0.001 0.005 0.01 0.05 0.1) ; Learning rate drop factor = (0.9 0.8 0.7 0.6 0.5 0.4); Dropout factor = (0, 10%, 30%, 70%, 90% ); LReLU factor = (0.001, 0.005, 0.05, 0.1, 0.5); Optimizer	Laser scanned range images	Determining the optimal hyper-parameter sets for the training of DCNN classifiers (the mini-batch size, learning rate, dropout factor, and LReLU factor) for roadway crack classification task is performed and mini-batch size of 60, initial learning rate of 0.01 and a drop factor of 0.8, dropout factor of 50%, and LReLU factor of 0.01 are observed as leading to the highest classification performance for roadway crack images.

		Algorithm = SGD		
Adedigba et al. (2021) [39]	Resnet152 DenseNet169	Learning Rates ( $\alpha_{min}$ and $\alpha_{max}$ ) (cyclical learning rate policy) = (1e-3 to 1e-1 for ResNet and 1e-2 & 1e-4 for DenseNet); Momentum = (0.8 to 0.99 for ResNet and 0.79 to 0.9 for DenseNet); Optimizer = Adam ; Weight-Decay = 0.01; Batch Size = (64 and higher); Epochs = 20	Chest X-ray (CXR) datasets	Presents techniques for diagnosing COVID-19 from chest X-ray (CXR) and address problems associated with training deep models with less voluminous datasets. An automatic method for optimal hyper-parameter selection & dynamic tuning of hyper-parameter is performed and cyclical learning rate policy used for setting the values for learning rate ensured rapid convergence of the loss function. ResNet and DenseNet have achieved accuracy of 95.43% and 96.83%, respectively, on the validation set, 97% accuracy on the test set. No conclusion is given on optimal values for the selected hyper-parameters.
Proposed Work	ResNet18 GoogLeNet AlexNet	Initial learning rate (ILR), mini-batch size (MBS), max epochs (ME), & Optimizer Algorithm (OA) (table 4, table 5, table 6 of section 3)	Natural scene images of Wang database(20 classes)	Thirty plus experimentations are conducted on three pretrained CNNs to fine-tune the hyper-parameters to yield optimal values of hyper-parameters and also an efficient transfer-net for CBIR. (The details of optimal values of hyper-parameters recommendation are given in section 4). The ResNet18 consumes less time during transfer learning and achieved image retrieval accuracy of 99.09%.

The state-of-the-art in Table 7 details the various values tested during fine tuning of the hyper-parameters, various pretrained CNNs and datasets used in different applications (retrieval of images, classification of images, & medical diagnosing from medical images etc). The proposed work

#### 4. Results and Recommendations

In this section, the validation accuracy, retrieval accuracy of the transferred networks of chosen pre-trained CNNs and observations made on values of hyper parameters and distance metrics while validating the proposed image descriptors & transfer-networks in retrieval of images are reported.

detailed the pretrained CNNs and hyper-parameters, dataset used for CBIR. And the optimal values required for the hyper-parameters of transfer-nets and the efficient pre-trained CNN for CBIR are recommended and are discussed in section 4.

*RETRIEVAL RESULTS OF CNNs.* To verify the effectiveness of the proposed method, the effects of the metrics on classification scores are examined. Accordingly, a comparison is made among the six distance metrics and 3 pre-trained CNNs with the small datasets. Using the transfer networks of pre-trained CNNs the classification scores are obtained for 20 classes and image retrieval is performed by choosing random query images from each class. The recall results produced are shown in Table 8 and Table 9 respectively.

**Table 8.** The mean recall accuracies of Resnet18, GoogLeNet, & AlexNet on wang20 dataset

Image classes available in wang20 dataset	Distance Metrics & The Pre-trained CNNs																	
	Euclidean			Seuclidean			Cityblock			Cosine			Mahalanobis			Chebychev		
	Resnet18	GoogLeNet	AlexNet	Resnet18	GoogLeNet	AlexNet	Resnet18	GoogLeNet	AlexNet	Resnet18	GoogLeNet	AlexNet	Resnet18	GoogLeNet	AlexNet	Resnet18	GoogLeNet	AlexNet
Africans	100	99	98	100	99	98	98	99	98	98	99	98	100	98	99	100	99	98
Beach	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Monument	99	100	100	99	99	100	99	99	100	99	100	100	100	99	100	90	100	100
Bus	100	99	100	100	99	100	100	99	100	100	99	100	100	99	100	100	99	100
Butterfly	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Car	100	100	99	100	100	99	100	100	99	100	100	99	100	100	100	100	100	99
Dinosaur	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Elephant	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Food	99	95	100	99	96	100	100	99	100	99	99	100	99	96	100	100	99	100
Forest	100	100	99	100	100	99	100	100	99	100	100	99	100	100	98	100	100	99
Horse	99	100	100	99	100	100	99	100	100	99	100	100	100	100	100	100	100	100
Map	100	96	99	100	97	99	100	97	99	100	97	99	100	97	96	100	97	99
Mountain	99	100	97	99	100	97	99	100	96	99	100	97	100	100	99	100	100	96
Planet	99	98	100	99	99	99	99	99	100	98	97	100	100	98	98	99	99	100
Rose	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Space	100	97	97	100	98	96	100	98	98	100	98	96	98	97	95	100	98	96
Sports	98	96	93	97	98	93	98	97	96	98	97	96	99	98	98	99	94	96
Sunrise	100	100	99	100	100	99	100	100	99	100	100	99	100	100	94	100	100	99
Text	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Yacht	99	100	98	99	100	96	99	100	100	99	100	98	100	100	98	100	100	99

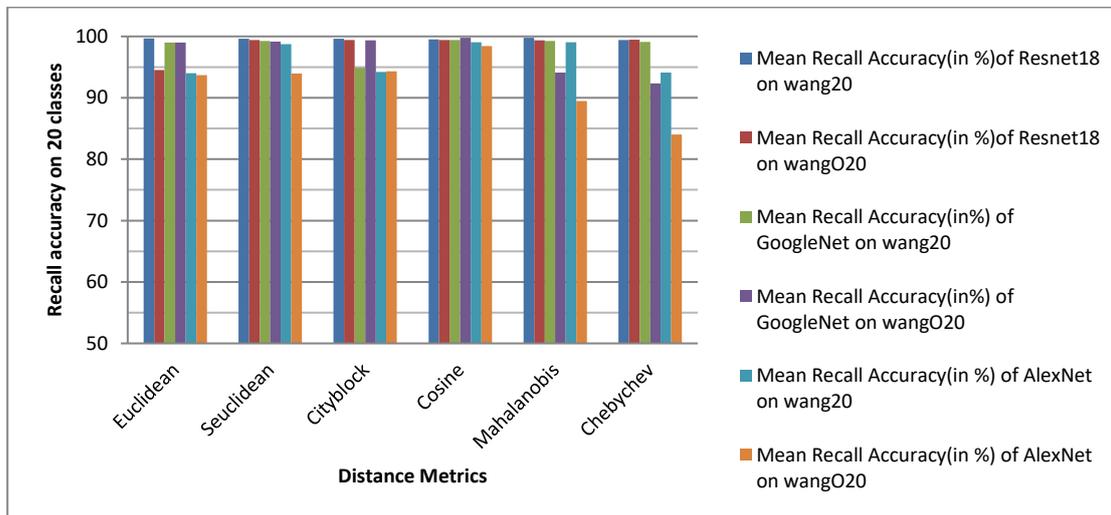
mean Recall Accuracy (in %) of the proposed system	99.65	99	94	99.6	99.25	98.75	99.6	94.87	94.2	99.5	99.4	99.05	99.8	99.1	98.75	99.4	99.25	99.05
----------------------------------------------------	-------	----	----	------	-------	-------	------	-------	------	------	------	-------	------	------	-------	------	-------	-------

**Table 9.** The mean recall accuracies of Resnet18, GoogLeNet, & AlexNet on wangO20 dataset

For the image classes available in wangO20 dataset	Distance Metrics & the Pre-trained CNNs																	
	Euclidean			Seuclidean			Cityblock			Cosine			Mahalanobis			Chebychev		
	Resnet18	GoogLeNet	AlexNet	Resnet18	GoogLeNet	AlexNet	Resnet18	GoogLeNet	AlexNet	Resnet18	GoogLeNet	AlexNet	Resnet18	GoogLeNet	AlexNet	Resnet18	GoogLeNet	AlexNet
mean Recall Accuracy (in %) on wangO20	94.55	99	93.7	99.4	99.15	93.95	99.4	99.35	94.3	99.4	99.8	98.4	99.35	94.1	89.5	99.45	92.35	84

The comparison of recall accuracies of the Resnet18, GoogLeNet & Alexnet are shown in the graph of Figure 2,

which also exhibits the impact of distance metrics in retrieval of images using the proposed image descriptor on these networks.



**Fig. 2.** Comparison of mean recall accuracies obtained on 20 classes' datasets using various distance metrics

The snapshots of retrieval results obtained for the Query-Image 'Dinosaur' by employing Euclidean distance metric is shown in Figure 3.

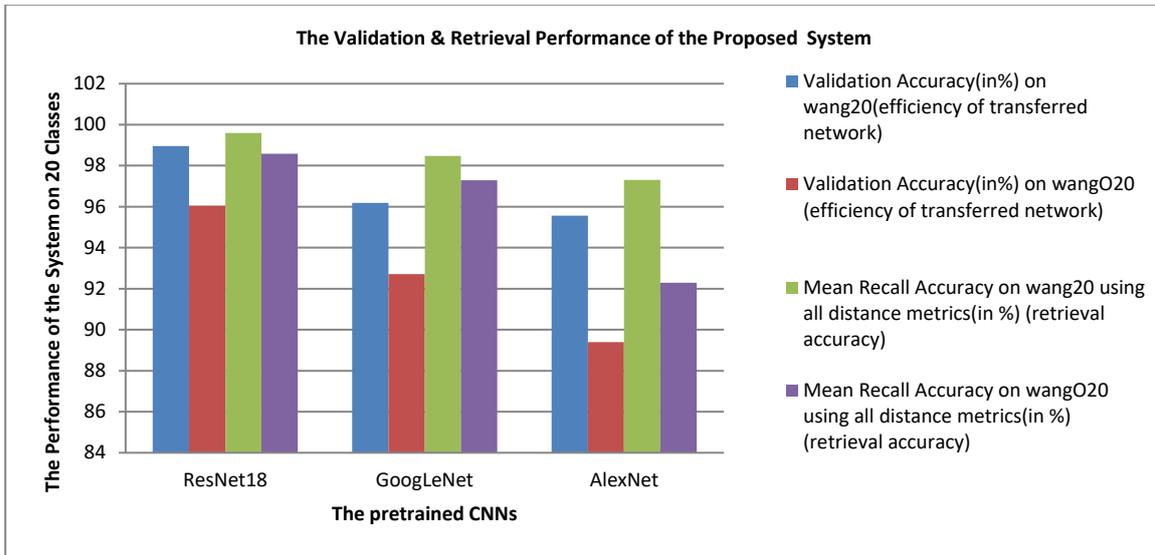


**Fig. 3.** Retrieval results of the Query-Image ‘Dinosaur’ after employing Euclidean distance metric

The performance of the proposed system in terms of validation and retrieval accuracy is given in Table 10 and Figure 4 respectively.

**Table 10.** The performance of the best experiments, the transferred-nets and distance metrics on wang20 and wangO20 datasets

The Pre-trained CNNs	The best experiments chosen	Validation Accuracy (in%) on wang20 (efficiency of transferred network)	Validation Accuracy (in%) on wangO20 (efficiency of transferred network)	Mean Recall Accuracy on wang20 using all distance metrics (in %) (retrieval accuracy)	Mean Recall Accuracy on wangO20 using all distance metrics (in %) (retrieval accuracy)
ResNet18	E6	98.95	96.04	99.59	98.59
GoogLeNet	E22	96.19	92.72	98.48	97.29
AlexNet	E22	95.56	89.40	97.3	92.30



**Fig. 4.** The performance of the proposed system

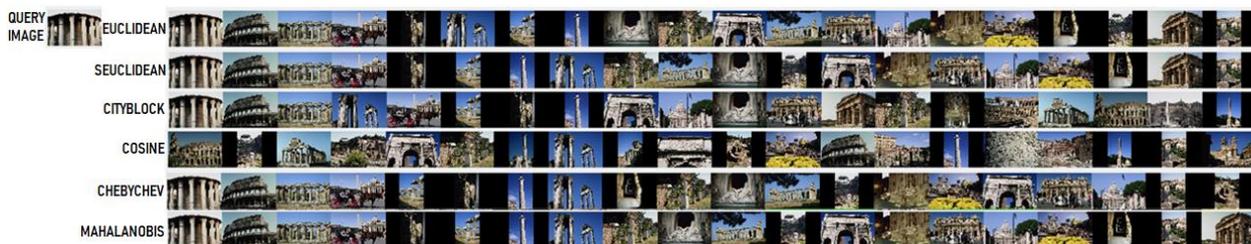
The Table 10 and Figure 4 illustrates the performance of the best experiments, the transferred-networks and the 6 distance metrics on wang20 and wangO20 datasets respectively. The classification scores are efficient for image retrieval. The performance of the best experiments in yielding efficient transfer networks is presented in terms of validation accuracy. The performance of the metrics and transferred networks using proposed classification scores is reported in terms of recall accuracy. It is observed that

among the chosen CNNs, the ResNet18 is less time consuming and yields best results. The experimental values of E22 (OA-sgdm, MBS-10, ME-10, ILR-1e-4) have obtained good results on all the 3 networks.

The instances of retrieval results of all the six distance metrics are shown in Figure 5 and Figure 6 respectively.



**Fig.5.** Retrieval results of all the six distance metrics used in retrieval of rose images



**Fig.6.** Retrieval results of all the six distance metrics used in retrieval of monument images

The distance metrics used in similarity matching are suitable to deal with the vector of classification scores. The pair-wise distance computed between the classification scores of query image to database images generates the vector of distances. The smallest difference between two sets of vectors will yield the larger similarity between the images. The images displayed in the first row are similar as shown in Figure 5 & Figure 6 for the distances generated

from Euclidean, Seuclidean, Cityblock and Mahalanobis. For Chebychev it is observed that color features are emphasized and most similar color images are in first places. The requirement for usage of Mahalanobis distance metric is that the feature matrix must be square with the same number of columns as query-feature matrix and it must be symmetric and positive definite. Despite the not in place arrangements of similar characteristic images, the

best recall results are obtained by the cosine distance metric for all the image classes. The mean recalls computed on 20 classes of the Wang dataset using the three transfer networks and all the 6 distance metrics is

shown in Table 11. This shows that the proposed image descriptor is best for retrieval of images using mentioned distance metrics.

**Table 11.** The summarized values from table 8 and table 9

Distance Metrics	ResNet18	GoogLeNet	AlexNet	Mean Recall
Euclidean	97.1	99	93.85	96.65
Seuclidean	99.5	99.2	96.35	98.35
Cityblock	99.5	97.1	94.25	96.95
Cosine	99.45	99.6	98.72	99.25
Mahalano bis	99.575	96.6	94.125	96.76
Chebychev	99.425	95.8	91.525	95.58
Mean Recall	99.091	97.88	94.80	

The data in Table 11 indicates ResNet18 (mean recall of 99.091) is better than the other two transfer networks namely, GoogLeNet (mean recall of 97.88) and AlexNet (mean recall of 94.80). Further for image retrieval the table indicates Cosine similarity measure (mean recall of 99.25) yields better results.

*RECOMMENDATIONS.* The most suitable metrics and optimal values of the hyper-parameters are recommended on the 3 CNNs for an efficient CBIR and the details are provided in Table 12.

**Table 12.** The recommendations for an efficient CBIR system

Sl. No.	The Best Pre-trained CNN for IR (in descending order)	Suitable Distance Metric	Optimal Values of the Hyper-parameters (OA, MBS, ME, ILR)	Remarks
1	ResNet18	Cosine	{OA=Adam, MBS=32, ME=10, ILR=1e-4}	ResNet18 is less time consuming and the Adam optimizer is suitable to use with transfer-learning on Resnet18.
2	GoogLeNet	Cosine	{OA=sgdm, MBS=10, ME=10, ILR=1e-4}	GoogLeNet yields best results when used with sgdm optimizer for transfer-learning.
3	AlexNet	Cosine	{OA=sgdm, MBS=10, ME=10, ILR=1e-4}	AlexNet is more time consuming compared to ResNet18 &

				GoogLeNet and yields best results with sgd optimizer.
--	--	--	--	-------------------------------------------------------

Table 12 shows the parametric values optimal for efficient transfer-learning on each of the three pre-trained CNNs and distance metrics for retrieval of images. And the

Cosine similarity measure is observed as being the most suitable metric to use with the classification scores of all three transfer networks. The following section describes the details of the recommended framework of CBIR.

### 5. Recommended Framework for CBIR

The recommended framework of CBIR is shown in Figure 7 and the details are described below.

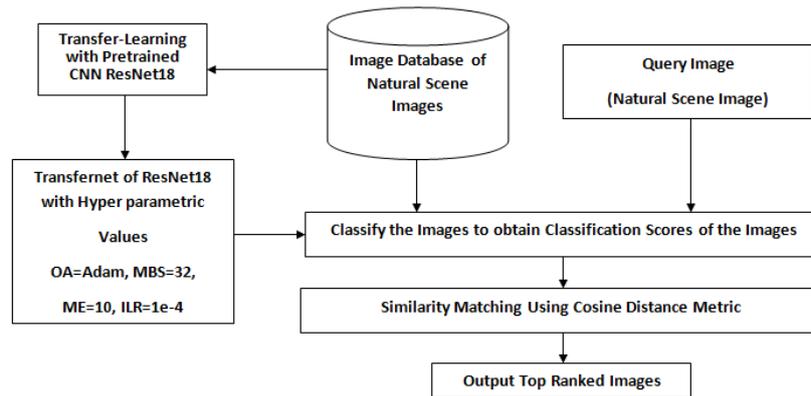


Fig.7. The recommended framework for CBIR

Figure 7 depicts the framework recommended for CBIR with the transfer-learning proposed on ResNet18 using the hyper-parametric values and similarity measure mentioned as in Table 12. The experiments conducted on pre-trained CNNs (ResNet18, GoogLeNet, AlexNet) using natural scene images as described in this paper have led us to the following conclusions. The ResNet18 is less time consuming while transfer-learning on natural scene images and the optimal values of the hyper parameters, the Adam optimizer, MBS of 32, ME of 10, and 1e-4 of ILR

yields efficient transfer-net. The Cosine similarity measure is the most efficient (Table 11) to use with classification scores for matching and retrieval of images. And the recommended CBIR system achieves the retrieval accuracy of 99.45% on natural scene images.

Comparison made between proposed framework of CBIR to the current state-of-the-art CBIR systems are tabulated in Table 13.

Table 13. Comparison of proposed framework of CBIR to current state-of-the-art of CBIR systems

Paper	Method	Pretrained CNN used	Dataset	Image features	Distance metrics	Remarks
Öztürk et al. (2021) [40]	Hash codes are generated by dictionary learning (DL) approach using CNN. The Convolution layers between 92 <sup>nd</sup> and 157 <sup>th</sup> layers of the ResNet-50 architecture are used to create the dictionaries.	ResNet-50 (MBS=20, ILR= 0.0005, ME=15,	Natural scene images of COREL dataset (26 classes are selected)	Binary and Hash Table	Euclidean distance & Hamming distance	The performance of Hamming distance is lower than Euclidean distance using DL method in retrieving the natural scene images. Retrieval results are presented in terms of mean average retrieval time.

<p>Trappey et al. (2021) [41]</p>	<p>VGG19 is used as LogoSimNet. The transfer learning methodology uses embedded learning with triplet loss to fine-tune a pre-trained convolutional neural network model.</p>	<p>VGG16, VGG19 (combines a pre-trained ImageNet CNN with fine-tuned layers using 7625 Logo-2K+ images in 140 categories), ResNet50, InceptionV3, MobileNetV2 and EfficientNet B0. (MBS=256, ME=20, ILR=0.001, OA=ADAM)</p>	<p>Logo-2K+ dataset</p>	<p>Transfer learning and low level features (color, size, background, texture, logo contour, logo feature position, partial similarity, and shape concept).</p>	<p>L2 distance</p>	<p>VGG19 , VGG16 and Resnet50 are observed to be top performing models. The performance is measured by (mAP and Recall@K) Recall@10 and obtained the accuracy of 95% in logo image retrieval .</p>
<p>Pathak et al. (2021) [42]</p>	<p>Feature-fusion of CNN (Improved and refined version of darknet-53, named GN-Inception-Darknet-53) based features and handcraft features (A modified version of dot-diffused block truncation coding (DDBTC) is used to extract handcraft features)</p>	<p>GN-Inception-Darknet-53</p>	<p>Corel-1K, Corel-5K, Corel-10K, VisTex, Stex &amp; Color Brodatz</p>	<p>High-level features (from the output layer of the average pooling layer i.e. 'Avg_1') and low level features (from RGB color space)</p>	<p>-</p>	<p>5 different measures(Average Precision Rate (APR), Average Recall Rate (ARR), F-Measure, Average Normalized Modified Retrieval Rank (ANMRR), and Total Minimum Retrieval Epoch (TMRE)) are used to measure the performance of the GN-Inception-Darknet-53 &amp; DDBTC with feature fusion in CBIR and shows 7.02% of improvement in retrieval of natural scene &amp; texture images compared to the performance of other related works.</p>

Zhang et al. (2022) [43]	Transfer learning and hand-crafted features	Inception-v3 (ILR=0.0001, MBS=64, epoch=224)	Woolen fabric images (No public or standard dataset is available & 82,073 wool fabrics of different categories were collected from a woolen fabric factory)	Transfer learning and hand-crafted features (The feature was extracted by the ORB descriptor and aggregated by VLAD, and the feature dimension was reduced by the principle component analysis (PCA))	Ball tree with KNN search method	Woolen fabric image (texture image) retrieval is performed. The performance of the system is measured by Mean Average Precision (mAP) and is 0.824 and average elapsed time is 0.408s.
Proposed work	Transfer learning for CBIR	ResNet18 (ILR=1e-4, MBS=32, ME=10, OA=adam)	Natural scene images of Wang Database (20 classes)	Classification scores from the Softmax layer of Transfer-Net of ResNet18	Cosine distance	Content-based image retrieval is performed. The performance of the system is measured by recall rate. The recall accuracy of the proposed system is 99.45%.

The Table 13 shows the comparative analysis made between proposed system to the other systems in [40-43], with respect to methodologies, Pretrained CNNs, datasets, image features, distance metrics used in retrieval of images. And the proposed work recommends an efficient framework for CBIR system.

## 6. Conclusion

An efficient framework for CBIR using dimensionally efficient CNN classification scores is proposed in this paper. The dimensionally efficient vector of classification scores is generated by utilizing the deep features at the Softmax layer of the pre-trained CNN. Extensive experimentations for optimization of values of the hyper-parameters are conducted and using these optimal values of hyper-parameters an effective transfer-learning is performed on the three pre-trained CNNs namely, the ResNet18, GoogLeNet, and AlexNet on natural scene images and the Wang database is used for experimentation. For the retrieval task performed with deep features, the similarity measure influences the retrieval performances, and hence the distance metrics namely Euclidean,

euclidean, Cosine, Correlation, Mahalanobis & Chebychev are applied to evaluate the efficacy of these metrics in similarity measurement on classification scores in CBIR. The choice of distance metric has a bearing on the performance of the system. And the Cosine similarity measure is observed as being the most suitable to use with the CNN classification scores for retrieval of images. And unlike the image descriptors such as hash codes and binary codes in IR these CNN classification scores offers a storage and retrieval efficient representation of images and the experimentation conducted on pre-trained CNNs and distance metrics demonstrate the suitability of classification scores in CBIR. After extensive experimentations on pre-trained CNNs and the 6 distance metrics, an efficient framework for CBIR is recommended which consists of a transfer network of ResNet18 with the optimal values of hyper parameters having Adam optimizer, MBS of 32, ME of 10, and ILR of 1e-4 and the Cosine similarity for matching and retrieval of images. The recommended framework of CBIR achieves the retrieval accuracy of 99.45% on natural scene images of twenty

classes of the Wang dataset, and hence is a suitable framework for CBIR.

The CNN classification scores have reduced the problem of “Curse of dimensionality” but still there is a requirement to reduce the size of classification scores which increases linearly with the increase in the number of classes in the database. In phase-3 similarity matching and retrieval of images are performed with the Cosine similarity measure, showing the suitability of the metric using classification scores and suitability of classification scores in the retrieval of images in CBIR. And Cosine similarity measure is found to be the best one to use with the classification scores of the transfer networks. In future, an automatic tuning of values of hyper-parameters during transfer-learning and reduction in the size of classification score vector without compromising the efficiency of retrieval of images in CBIR will be explored.

### Acknowledgments

This work is supported by department of CSE, VTU-Belagavi and partially supported by all other departments of VTU Belagavi

### Author contributions

SA and HP conceived of the presented idea. HP developed the theory and performed the computations and wrote the manuscript under the supervision of SA.

### Availability of data and materials

The datasets analyzed during current study are publicly available in the Wang Database.

### Competing interests

The authors report there are no competing interests to declare.

### Funding

None

### References

[1] Prasad, B. E., Gupta, A., Toong, H.-M. D., & Madnick, S. E. (1987), “A Microcomputer-Based Image Database Management System”, *IEEE Transactions on Industrial Electronics*, IE-34(1), 83–88. <https://doi.org/10.1109/tie.1987.350929>

[2] Kato, Toshikazu; Jamberdino, Albert A.; Niblack, Carlton W. (1992), “Image Storage and Retrieval Systems – Database architecture for content-based image retrieval”, *SPIE Proceedings* [SPIE SPIE/IS&T 1992 Symposium on Electronic Imaging: Science and Technology - San Jose, CA (Sunday 9 February 1992)], 1662, 112–123. doi:10.1117/12.58497

[3] Zachary, J.M.; Iyengar, S.S. (1999), “Content based image retrieval systems”, [IEEE Comput. Soc 1999 IEEE Symposium on Application-Specific Systems and Software Engineering and Technology. ASSET'99 - Richardson, TX, USA (24-27 March 1999)] *Proceedings 1999 IEEE Symposium on Application-Specific Systems and Software Engineering and Technology. ASSET'99* (Cat. No.PR00122), 136–143. doi:10.1109/asset.1999.756762

[4] Veltkamp, R. C., & Tanase, M. (2002), “A Survey of Content-Based Image Retrieval Systems”, *Content-Based Image and Video Retrieval*, 47–101. [https://doi.org/10.1007/978-1-4615-0987-5\\_5](https://doi.org/10.1007/978-1-4615-0987-5_5)

[5] Liu, Y., Zhang, D., Lu, G., & Ma, W.-Y. (2007), “A survey of content-based image retrieval with high-level semantics”, *Pattern Recognition*, 40(1), 262–282. <https://doi.org/10.1016/j.patcog.2006.04.045>

[6] Rafiee, G. & Dlay, S.s & Woo, Wai Lok. (2010), “A review of content-based image retrieval”, *2010 7th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2010*. 775-779. 10.1109/CSNDSP16145.2010.5580313.

[7] Yasmin, M., Mohsin, S., & Sharif, M. (2014), “Intelligent Image Retrieval Techniques: A Survey”, *Journal of Applied Research and Technology*, 12(1), 87–103. [https://doi.org/10.1016/s1665-6423\(14\)71609-8](https://doi.org/10.1016/s1665-6423(14)71609-8)

[8] Rodrigues, Josiane & Cristo, Marco & Colonna, Juan. (2020), “ Deep hashing for multi-label image retrieval: a survey”, *Artificial Intelligence Review*. 53. 10.1007/s10462-020-09820-x.

[9] Dubey, S. R. (2021), “A Decade Survey of Content Based Image Retrieval using Deep Learning”, *IEEE Transactions on Circuits and Systems for Video Technology*, 1–1. <https://doi.org/10.1109/tcsvt.2021.3080920>

[10] Varga, Domonkos; Sziranyi, Tamas (2016), “Fast content-based image retrieval using convolutional neural network and hash function”, [IEEE 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC) - Budapest, Hungary (2016.10.9-2016.10.12)] *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 002636–002640. doi:10.1109/SMC.2016.7844637

[11] Alzubi, A., Amira, A., & Ramzan, N. (2017), “Content-based image retrieval with compact deep convolutional features”, *Neurocomputing*, 249, 95–105. <https://doi.org/10.1016/j.neucom.2017.03.072>

[12] Tzelepi, M., & Tefas, A. (2018), “Deep convolutional

- learning for Content Based Image Retrieval”, *Neurocomputing*, 275, 2467–2478. <https://doi.org/10.1016/j.neucom.2017.11.022>
- [13] Tzelepi, Maria; Tefas, Anastasios (2018), “Fully Unsupervised Convolutional Learning for Fast Image Retrieval”, [ACM Press the 10th Hellenic Conference - Patras, Greece (2018.07.09-2018.07.12)] *Proceedings of the 10th Hellenic Conference on Artificial Intelligence - SETN '18*, pp. 1–6. doi:10.1145/3200947.3201007
- [14] Chitrakar, Pradip; Zhang, Chengcui; Warner, Gary; Liao, Xinpeng (2016), “Social Media Image Retrieval Using Distilled Convolutional Neural Network for Suspicious e-Crime and Terrorist Account Detection”, [IEEE 2016 IEEE International Symposium on Multimedia (ISM) - San Jose, CA, USA (2016.12.11-2016.12.13)] *2016 IEEE International Symposium on Multimedia (ISM)*, pp.493–498. doi:10.1109/ISM.2016.0110
- [15] Zhuang, Fuzhen; Qi, Zhiyuan; Duan, Keyu; Xi, Dongbo; Zhu, Yongchun; Zhu, Hengshu; Xiong, Hui; He, Qing (2020), “A Comprehensive Survey on Transfer Learning”, *Proceedings of the IEEE*, pp. 1–34. doi:10.1109/JPROC.2020.3004555
- [16] Day, O., & Khoshgoftaar, T. M. (2017), “A survey on heterogeneous transfer learning”, *Journal of Big Data*, 4(1). <https://doi.org/10.1186/s40537-017-0089-0>
- [17] Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018), “A Survey on Deep Transfer Learning”, *Lecture Notes in Computer Science*, 270–279. [https://doi.org/10.1007/978-3-030-01424-7\\_27](https://doi.org/10.1007/978-3-030-01424-7_27)
- [18] Shorten, C., & Khoshgoftaar, T. M. (2019), “A survey on Image Data Augmentation for Deep Learning”, *Journal of Big Data*, 6(1). 60–. <https://doi.org/10.1186/s40537-019-0197-0>
- [19] He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian (2016), “Deep Residual Learning for Image Recognition”, [IEEE 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - Las Vegas, NV, USA (2016.6.27-2016.6.30)] *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.770–778. doi:10.1109/CVPR.2016.90
- [20] Szegedy, Christian; Wei Liu, ; Yangqing Jia, ; Sermanet, Pierre; Reed, Scott; Anguelov, Dragomir; Erhan, Dumitru; Vanhoucke, Vincent; Rabinovich, Andrew (2015), “Going deeper with convolutions”, [IEEE 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - Boston, MA, USA (2015.6.7-2015.6.12)] *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9. doi:10.1109/CVPR.2015.7298594
- [21] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017), “ImageNet classification with deep convolutional neural networks”, *Communications of the ACM*, 60(6), 84–90. <https://doi.org/10.1145/3065386>
- [22] Lu, Xin; Kang, Xin; Nishide, Shun; Ren, Fuji (2019), “Object detection based on SSD-ResNet”, [IEEE 2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS) - Singapore (2019.12.19-2019.12.21)] *2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, 89–92. doi:10.1109/CCIS48116.2019.9073753
- [23] X. Liu, Y. Meng and M. Fu, "Classification Research Based on Residual Network for Hyperspectral Image," *2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)*, 2019, pp. 911-915, doi: 10.1109/SIPROCESS.2019.8868838.
- [24] Zhou, Yitao; Ren, Fuji; Nishide, Shun; Kang, Xin (2019), “Facial Sentiment Classification Based on Resnet-18 Model”, [IEEE 2019 International Conference on Electronic Engineering and Informatics (EEI) - Nanjing, China (2019.11.8-2019.11.10)] *2019 International Conference on Electronic Engineering and Informatics (EEI)*, 463–466. doi:10.1109/eei48997.2019.00106
- [25] Kumar, Vidit; Tripathi, Vikas; Pant, Bhaskar (2020), “Content based Fine-Grained Image Retrieval using Convolutional Neural Network”, [IEEE 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN) - Noida, India (2020.2.27-2020.2.28)] *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*, 1120–1125. doi:10.1109/SPIN48934.2020.9071334
- [26] Y. Tan, Y. Li, H. Liu, W. Lu and X. Xiao, (2020), “Performance Comparison of Data Classification based on Modern Convolutional Neural Network Architectures”, *2020 39th Chinese Control Conference (CCC)*, 2020, pp. 815-818, doi: 10.23919/CCC50068.2020.9189237.
- [27] Paidi, V., Fleyeh, H., & Nyberg, R. G. (2020), “Deep learning-based vehicle occupancy detection in an open parking lot using thermal camera”, *IET Intelligent Transport Systems*, 14(10), 1295–1302. <https://doi.org/10.1049/iet-its.2019.0468>
- [28] Malik, F., & Baharudin, B. (2013), “Analysis of distance metrics in content-based image retrieval using statistical quantized histogram texture features in the DCT domain”, *Journal of King Saud*

University - Computer and Information Sciences, 25(2), 207–218. <https://doi.org/10.1016/j.jksuci.2012.11.004>

- [29] B. Patel, k. Yadav and D. Ghosh, (2020), “State-of-Art: Similarity Assessment for Content Based Image Retrieval System”, *2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC)*, 2020, pp. 1-6, doi: 10.1109/iSSSC50941.2020.9358899.
- [30] Ó Searcóid, Mícheál (2006), “2.7 Distances from Sets to Sets. Metric Spaces”, *Springer Undergraduate Mathematics Series, Springer*, pp. 29–30, ISBN 978-1-84628-627-8
- [31] Bellot, F., & Krause, E. E. (1988), “Taxicab Geometry: An Adventure in Non-Euclidean Geometry”, *The Mathematical Gazette*, 72(461), 255. <https://doi.org/10.2307/3618288>
- [32] Dengsheng Zhang, & Guojun Lu. (2003), “Evaluation of similarity measurement for image retrieval”, *International Conference on Neural Networks and Signal Processing, 2003. Proceedings of the 2003*, Vol.2, 928 - 931. <https://doi.org/10.1109/icnns.2003.1280752>
- [33] Wang, J. Z., Jia Li, & Wiederhold, G. (2001), “SIMPLicity: semantics-sensitive integrated matching for picture libraries”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9), 947–963. <https://doi.org/10.1109/34.955109> <https://sites.google.com/site/dctresearch/Home/content-based-image-retrieval>.
- [34] Jia Li, & Wang, J. Z. (2003), “Automatic linguistic indexing of pictures by a statistical modeling approach”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9), 1075–1088. <https://doi.org/10.1109/tpami.2003.1227984>
- [35] Kingma, D., and Jimmy, B. (2014), “Adam: A Method for Stochastic Optimization”, arXiv preprint arXiv:1412.6980, 2014.
- [36] Robert, C. (2014), “Machine Learning, a Probabilistic Perspective”, *CHANCE*, 27(2), 62–63. <https://doi.org/10.1080/09332480.2014.914768>.
- [37] Li, H., Chaudhari, P., Yang, H., Lam, M., Ravichandran, A., Bhotika, R., Soatto, S., (2020), “Rethinking the Hyper-parameters for Fine-Tuning”, *ICLR 2020*, arXiv:2002.11770v1. <https://doi.org/10.48550/arXiv.2002.11770>
- [38] Zhou, S., & Song, W. (2020), “Deep learning-based roadway crack classification using laser-scanned range images: A comparative study on hyper-parameter selection”, *Automation in Construction*, 114, 103171. doi:10.1016/j.autcon.2020.103171
- [39] Adedigba, A. P., Adeshina, S. A., Aina, O. E., & Aibinu, A. M. (2021), “Optimal hyper-parameter selection of deep learning models for COVID-19 chest X-ray classification”, *Intelligence-Based Medicine*, 5, 100034. doi:10.1016/j.ibmed.2021.100034
- [40] Öztürk, Ş. (2021), “Convolutional neural network based dictionary learning to create hash codes for content-based image retrieval”, *Procedia Computer Science*, 183, 624–629. <https://doi.org/10.1016/j.procs.2021.02.106>
- [41] Trappey, A. J. C., Trappey, C. V., & Shih, S. (2021), “An intelligent content-based image retrieval methodology using transfer learning for digital IP protection”, *Advanced Engineering Informatics*, 48, 101291. <https://doi.org/10.1016/j.aei.2021.101291>
- [42] Pathak, D., & Raju, U. S. N. (2021), “Content-based image retrieval using feature-fusion of GroupNormalized-Inception-Darknet-53 features and handcraft features”, *Optik*, 246, 167754. <https://doi.org/10.1016/j.ijleo.2021.167754>
- [43] Zhang, N., Shamey, R., Xiang, J., Pan, R., Gao, W. (2022), “A novel image retrieval strategy based on transfer learning and hand-crafted features for wool fabric”, *Expert Systems with Applications*, 191, 116229. <https://doi.org/10.1016/j.eswa.2021.116229>