Egy. J. Pure & Appl. Sci. 2024; 62(3):64-78

Egyptian Journal of Pure and Applied Science

Harnessing Deep Features for Improved Multi-Query Texture Retrieval

Hewayda M.S. Lotfy

Mathematics Department, Faculty of Science, Ain Shams University, Cairo, Egypt

ARTICLE INFO

Received 21 September 2024
Accepted 24 December 2024

Keywords
CNN features,
Transfer Learning,
CBIR,
Multi Queries Image retrieval

Correspondence
Hewayda M.S. Lotfy

E-mail*
(Corresponding Author)
Hewayda_lotfy@sci.asu.edu.eg

ABSTRACT

Developing an efficient classifier-based image retrieval system is vital for accurately and swiftly retrieving relevant images in computer vision applications. Hand-crafted features usually require extensive tuning and may fail to generalize across different types of images, making the retrieval process labor-intensive and less adaptable. Despite the advancements in deep learning for image retrieval, there is limited research on integrating Multi-Query (MQ) techniques with deep features for image retrieval. The novel MQ Deep Image Retrieval (MQDIR) system exploits this approach to extract deep features from an Image Set (IS) and handle MQ simultaneously. The methodology enhances the retrieval process by capturing more nuanced image characteristics through using MQs that traditional methods might overlook. A new precision-based metric is introduced in this study to offer a comprehensive average performance evaluation. The metric considers the precision of retrieval results across multiple ISs and Convolutional Neural Networks CNNs and allows a finer assessment of system performance compared to conventional measures. The experiments are conducted on popular benchmark ISs, including texture images, and demonstrate that MQDIR consistently outperforms existing methods in terms of retrieval accuracy and efficiency.

1. Introduction

Content-Based Image Retrieval CBIR is a well-studied problem. The retrieval problem typically falls into two types: category-level retrieval and instance-level retrieval. For example, given an Egyptian Pyramid query, a category-level retrieval seeks to locate any pyramids to be considered as a match. Whereas an instance-level retrieval locates any Egyptian Pyramids to be considered a match.

Regular CBIR techniques may utilize machine learning methods, including both unsupervised learning (such as Kmeans, Fcmeans, and GMM) and supervised learning (such as SVM) as well as advanced deep learning approaches. In classifier-based CBIR systems, the classifier first identifies and then categorizes the image. In [1], the author uses a bag of LBP features and then applies classification at a Fully Connected Layer (FCL).

The region of interest is first identified using SURF points then LBP features are extracted and clustered using a proposed method called Clustering with Fixed Centers to create a bag of LBP features. In [2] a deep learning IR method based on hashing distance variance for MQ having multiple labels, which computes the minimum variation of Hamming hashing distances between queries and IS. The variance produces the nearest images to the Pareto space center and the hash codes of the images are generated by Resnet50. The MLMQ-IR method uses a new method depending on binary cross-entropy loss and bit-balance loss functions. It is competed with six methods depending on multi-label and MQ through the benchmark ISs using metrics such as NDCG, ACG and wMAP. It is claimed that the method has the best average mean rank extracting the most relevant images among the other IR methods. CNNs are among the most widely used algorithms for image classification. Their hierarchical structure and extensive parameterization make them highly effective for object detection and classification.

Usually, CNNs have two main components: the Feature Extraction (FE) network and the classifier network. The FE network processes the input image, performing automatic feature extraction, which is its primary benefit. The research in [3] proposes a strategy that relies on a weight-learner based on the MQ linear weighted combination of the distances between each query image and a given image in an IS. The weight of each feature descriptor is learned through unsupervised learning, allowing discrimination by assigning the largest weight to the most relevant feature. The methodology in [4], uses hash codes generated by a deep multilabel image hashing algorithm for MQ with CNN. Retrieval is based on the Pareto front method, and reranking is performed on the retrieved images using non-binary deep features, which can increase retrieval accuracy. In [5], texture features are taken from various pre-trained models like DenseNet201, ResNet50, ResNet101, and AlexNet, and then SVM is used as a classifier. The performance is investigated using the KTH-TIPS, CURET, and Flowers texture ISs.

In [6], a methodology for predicting tropical storms is presented, using a CNN trained on real-time images to extract features. This method is assumed to decrease delivery time for testing images while increasing prediction accuracy. The proposed cloud image classification shows 94% prediction accuracy. The research in [7-10] presents recent trends and reviews on CBIR. This paper is organized as follows: Section 2 introduces the methodology, Section 3 presents experimental results and discussion, and Section 4 offers the conclusion and future work.

2. MQDIR Methodology

In Transfer Learning, a pre-trained CNNw receives an IS as input and predicts its class labels. Cross-validation splits the IS into two subsets: a training set and a test set. Adjusting the network on the training set involves using a subset called the validation set. Applying metrics such as accuracy on the training images shows training progress, while applying these metrics to the test images measures the performance on the unseen data. Classifying the test images using a fine-tuned pre-trained model is critical, as performance is generally influenced by the IS. MQDIR follows an ablation procedure for learning which aims to determine the contribution of a component to a deep learning system by replacing a component, and then analyzing the resultant performance of the system. The system consists of two phases illustrated in Fig. 1 and listings (1 and 2). The first phase involves preparing two main elements for the FE process, the CNNw model and the ISi. The CNNw are Fine-tuned and trained for each IS. The second phase uses the MQ approach for the retrieval process for each specific model and IS. Consequently, a CNNw is first selected and experimented separately with an IS which is then replaced with another IS. The ISs that contain texture images are selected. The selected CNNw provide suitable choice with respect to the computational resources and the benchmark ISs. The second phase prepares for a main element which is the Query Set Size (QSS) and Query Set (QS) members selection. The QS selection is based on semantically related queries and QSS is changed from size 1 up to size 5. The precision is calculated for each experiment using MQ retrieval results to measure the system performance.

2.1. Feature extraction

CNNs learn features directly from image data, however, they can be combined with handcrafted features [12]. The FE network consists of numerous Convolutional layers (CL) and FCL Layers. The initial layers identify low-level features in an image, such as edges, corners, and colors. As the layers deepen, they focus on more complex, high-level features like identifying wheels in a car, making the features less sensitive to image variations. The deep features can be extracted from either whole images or image patches.

These features are essentially of two types: Convolutional Layer features, FCL features, or a fusion of both. In [13], it is argued that features obtained from the last FCL are effective and high-dimensional global feature descriptors that reflect semantic information. In such a case, it may require dimensionality reduction methods to obtain low-dimensional features. The performance of different layers varies throughout the retrieval process, with FCL features often achieving better results on standard retrieval ISs. In [14], FCL features are used to generate hash codes for medical CBIR. Research in [15] demonstrated high performance in high-resolution remote sensing image retrieval using FCL features from a fine-tuned, pre-trained CNN. Deeper models are computationally expensive and beneficial for learning higher-level abstract features, which helps mitigate the semantic gap. Images often contain multiple semantics, making it challenging to search for images with multiple meanings. Using semantically related MQ techniques, rarely exploited, can assist this process by highlighting the relevant semantics. More detail about the FE process from CNNs according to the model selected are found in section 3.

2.2 The MQ Approach

This approach increases the amount of information gathered from the queries, which is crucial for the retrieval process. MQDIR uses a QS of images with arbitrary QSSs for retrieval. For example, if QS = {Q1, Q2}, then QSS = 2. Given QS= {Q1, …, Q5}, the QSS =1, …, 5 selected from QS. The QS should include semantically related images containing some object to strongly declare its properties. The abundance of semantically similar objects in the IS should boost the retrieval of similar objects. The literature on MQ retrieval mainly presents two different approaches: Early Fusion and Late Fusion.

Early Fusion is a feature-level fusion approach that combines queries in the feature space. It involves accumulating the features of multiple queries into a single feature vector (MQ_FV) before the image search process, then using MQ_FV for similarity inspection in the retrieval process. Early Fusion combination methods include MQ-maximum, MQ-average, and MQ-sum functions [3]. Preprocessing steps like normalization may be required to ensure that the features are on the same scale [15]. Late Fusion is a decision-level fusion method, it processes the queries individually and then combines their retrieval results into a single list. Examples of Late Fusion methods include Max Similarity of accumulated retrieval results and weighted similarity where each retrieved image is ranked according to the weight of the query [16]. The selection between early and late fusion depends on factors, including efficiency, performance, and the specific application. Listing 1 provides the algorithm of the MQDIR FE phase for any ISs. Listing 2 provides a detailed description of MQDIR retrieval process for a specific IS. We have chosen early fusion since it is usually more computationally efficient, while late fusion can provide higher flexibility and accuracy at the cost of additional computation, due to multiple retrieval operations, and memory usage. To accurately characterize performance, the retrieved images are often assessed using well-known retrieval performance measures, such as Precision-based metrics.

3. MQDIR Experimental Results and Discussion

To perform CNN learning, a collection of labeled ISs should be available, if not they need to be labeled manually, which can be labor-intensive. Several popular image sets are used in computer vision research, such as those from the ImageNet project, COCO, and Google Open Images. For example, ImageNet provides labels for entire images or bounding boxes for objects within images. Other widely used benchmarks include CIFAR-100, Oxford 102, and Oxford-IIIT Pets. Selecting popular benchmark image sets and CNN models provides ways for comparing results with other approaches. There is little research on CNN with MQ texture retrieval. MQDIR is experimented with widely used five benchmark ISs (Table 1) that contain texture images as follows. Describable Texture Dataset (DTD): a challenging texture IS where each category with images containing at least 90% of the category's semantic content [17].

KTH-TIPS: A texture IS containing same-sized categories, each with 81 images [5]. Flowers IS: A part of the DPhi Data Sprint#25: Flower Recognition, purely flowers texture IS [18]. Caltech 101: A general IS with categories that have large object variations and messy backgrounds. Some classes include textures such as airplanes, bonsai, sky, flowers, grass textures, and dotted and striped animals [19]. GHIM-10K: A natural general IS containing textures [20]. All experiments in this study are conducted using MATLAB PC with the following configuration: Intel(R) Core (TM) i7-5500U CPU @ 2.40GHz, 16 GB RAM, x64-based processor, and a single Nvidia GTX950 GPU.

3.1. CNN Parameter Selection and Tuning

Two popular pre-trained CNNs were utilized in the experiments. The first CNN_1 is AlexNet, consisting of 25 layers, (referred to as A) [21] was initially trained on over one million images and is capable of classifying images into one thousand categories, including various objects and animals. A requires input images of size 227-by-227-by-3, the image data augmenter preprocesses images before training and validation to automatically resize the training images to the required size, randomly flip the training images along the vertical axis, and randomly translate them up to 30 pixels horizontally and vertically. Also color preprocessing (gray to color image transform) to ensure that all output images have the number of channels required by the input layer in case the image datastore contains a mixture of grayscale and RGB images. The features of A are extracted from the 23rd (FCL_1) layer, producing one thousand features while the 24th layer further processes these values into binary outputs.

The second CNN_2 is ResNet18, consists of 71 layers, (referred to as R) but requires input images sized 224x224x3 then a similar preprocessing mentioned for A is used. R incorporates additional CLs to extract more abstract features and employs skip connections between CLs to address the vanishing gradient problem during training. For this experiment, only the final three layers of R were replaced with a new FCL_2, a softmax layer, and a classification output layer to adapt the network for different classification tasks. This customization allows R to be fine-tuned for the new ISs and FCL_2 is set to match the number of classes in the IS. MQDIR uses features extracted from the 69th layer of R, where the number of features extracted from this FCL_2 is one thousand.

The input ISs are divided into 70% for training and 30% for testing. In [22] a deep learning approach to classify blood smears images, total of 18,365 images, to its morphological classes of leukocytes. The results show classification accuracy of 93.30% and 93.85% for AlexNet and ResNet18 respectively. In [23] a deep learning approach for Face recognition using FERET IS consisting of 14,126 images, the classification accuracy for Resnet18 and AlexNet are 96.3% and 93.3%, respectively.

Model Overfitting can occur if the model memorizes the details of the training images rather than learning to generalize. To ensure the prevention of overfitting in MQDIR's models. First, image pre-processing techniques such as data augmentation are used to help prevent both overfitting and remembering the precise specifics of the training images and additionally regularizing the model in order to improve testing accuracy on several ISs. The shuffling of the training images before training and between epochs, as well as the shuffling of the validation images before each validation, helps prevent model overfitting by ensuring that batches represent the entire training set and are independent of ordering.

The initial fine-tuning experiments used batch sizes of 32 and 128 which resulted in very low validation accuracy and faster convergence times. It is notable in practice that using larger batch sizes, especially for small IS can degrade model quality due to reduced generalization capability. An attempt to help large batches perform better is discussed in [11]. Second, the careful selection of hyperparameters for fine-tuning such as, the piecewise learning rate for the layers is set to 0.0001, with a learning rate drop schedule set to 0.2 every 5 epochs. The minimum batch size for all input sizes is set to 20, with 10 epochs for R and 15 epochs for A. The momentum value for the stochastic gradient descent, representing the contribution of the previous iteration's parameter update to the current iteration, is set to 0.95. The selected models are trained for a few epochs and validated every few iterations, known as Validation Frequency (VF). If the number of training images is large and VF is small, then it is not learning enough. Conversely, if VF is too large, then it will spend an extended time training. During the experiment, when VF is set to 3 (Table 2), the training time is substantial. However, when VF reaches 15, the training time is reduced by more than a third. To accelerate learning in the new layers, the Weight LearnRateFactor and Bias LearnRateFactor for FCL_2 are both increased to 10.

For evaluating CNN performance, various metrics are derived from the confusion matrix, such as validation accuracy, and these results are summarized in Table 2. Fig. 2 shows that the training and validation loss both decrease and stabilize at a specific point which indicates that the model does not overfit or underfit. Loss and accuracy are measures of loss and accuracy on the training set, and also on the validation set. As a result, this model has an accuracy of ~95% on the training set and ~92% on the validation set. This figure is produced and examined for each IS_i and CNN_w.

The proposed methodology aims to enhance retrieval accuracy, ensuring that more images from the same or similar classes are retrieved. The MQ formulation is designed adequately reflecting the user's interest. A QSS with one query is termed as 1-QSS, with two queries as 2-QSS, and two queries and more as 2+QSS. Initial experiments comparing the early fusion with the MQ-maximum approach showed it returned fewer relevant images than the MQ-average approach [16]. Table 2 indicates the selected investigation trials, for instance, trial (b) for A and trial (d) for R are chosen for evaluation with any IS with small changes that depend on IS. Classification-based CBIR retrieval results are evaluated from two perspectives: similarity without refereeing to classes (non-class-based) and similarity refereeing to predefined classes (class-based). In non-class-based retrieval, similar images are acceptable from any class, while in class-based retrieval, they must come from the same Query class. A hybrid approach may combine both retrieval types. Classifying DTD textures is particularly challenging. The training accuracy on the DTD using either A or R is relatively low, with a maximum of 66%. Other datasets show higher training accuracies: Caltech101 reaches 92%, GHIM-10K 97.40%, Flowers 91.16%, and KTH-TIPS 96.88%. The classification accuracy, derived from the confusion matrix for R, is 98.15% for GHIM-10K, 98.12% for KTH-TIPS, 92.11% for Flowers, 91.74% for Caltech101, and 62.77% for DTD.

3.2. Retrieval Results and Ranking

MQDIR is tested with QSs selected from a maximum number of 47 classes for any IS, which is optionally can be chosen otherwise. Relevance is estimated based on the similarity found in the entire or the parts of image content. If an image contains water, vegetation, and boats, its relevant images are those that contain one or more of these elements. Images that do not contain any of these three elements are considered irrelevant.

For DTD retrieval using A as shown in Fig. 3-(i), 1-QSS retrieval results achieve maximum precision for 13 classes, and 2+QSS for up to 21 classes. Fig. 3-(iii) and Fig. 3-(iv) show the maximum precision profile for all classes using 1-QSS and 2+QSS. In Fig. 3-(vii), the number of relevant images for 1-QSS is less than for any 2+QSS. For DTD using R as shown in Fig. 3-(ii), 1-QSS achieves maximum precision for 16 classes, while 2+QSS covers up to 21 classes. Fig. 3-(v) and Fig. 3-(vi) show more maximum precision values closer to 1 compared to Fig. 3-(iii) and Fig. 3-(iv). Fig. 3-(vii) and Fig. 3-(viii) indicate that 1-QSS retrieves fewer relevant images compared to 2+QSS. For Caltech101 using A, as shown in Fig. 3-(i), 1-QSS achieves maximum precision for 17 classes, while 2+QSS reaches up to 30 classes. As shown in Fig. 3-(iii) and Fig. 3-(iv) maximum precision tends to improve with 2+QSS. In Fig. 3-(vii), the number of relevant images retrieved by 1-QSS is lower than by any 2+QSS. For Caltech101 using R, in Fig. 4-(ii), the 1-QSS retrieval results achieve maximum precision for 22 classes, while 2+QSS gains maximum precision for up to 34 classes. The maximum precision curve for 2+QSS is significantly better than that shown in Fig. 4-(iv), as illustrated in Fig. 4-(vi). In Figs. 4-(vii) and 4-(viii), the number of relevant images for 1-QSS is lower than for any 2+QSS. Fig. 7 demonstrates increasing precision with the MQ approach with GHIM-10K IS, where the single query approach usually has the lowest precision. Overall, Figs. 3, 4, and 7 indicate that R outperforms A, with more classes achieving 100% precision and that MQ approach generally surpasses the single query approach.

For the challenging textures DTD, which contains up to 47 classes such as banded, blotchy, braided, bubbly, bumpy, wrinkled, zigzagged, dotted, polka_dotted, …etc. Fig. 5 demonstrates the superiority of 2+QSS over 1-QSS. The 'polka_dotted' and the ‘dotted’ images are retrieved in all cases. The images in the ‘polka_dotted’ class are visually relevant to those in the ‘dotted’ class. Both 'Sprinkled' and 'Perforated' images were retrieved when using used 1-QSS. In non-class-based retrieval, the retrieval performance for these queries in Fig. 5 is assumed to be 100% since all retrieved images have the dotted texture, while in class-based retrieval, the performance is lower. It is also noted that most of the queries in 2+QSS retrieval reappeared in the retrieved images.

In Table 3, fourteen classes retrieval results are examined, the non-class-based retrieval yields an average increase in precision ranging from 7% to 50% over class-based retrieval. Some DTD textures, such as 'marbled' and 'pitted', are particularly difficult to recognize. QSs selected from the same class typically may retrieve images from the same class or other classes. For example, using Caltech101 IS, retrieving results for the 'faces' class also retrieves similar images from 'faces_easy', and queries from 'crayfish' class retrieves similar images from 'lobster' and 'crab'. Another example in Fig 6, the 1-QSS result for the 'crab' class (the query images include the first image that include a 'crab' object), more images from other classes similar to 'crab' appear, the irrelevant images like 'pizza' and 'background_google' also appear. Whereas in 2-QSS retrieval, such irrelevant images typically disappear. Generally, precision improves significantly with 2+QSS, implying better MQ retrieval results.

3.3. MQ_Precision Metric

Regarding the system results ranking performance, the $Precision@NR$ matrix records the precision for each class for each QSS and is defined as the precision calculated at a cutoff NR representing the number of top relevant images considered. In the experiments, NR is set to 15. A newly defined measure called MQ_Precision (CNNs, IDBs) is used for MQDIR performance evaluation given a collection of labeled ISs and CNNs as a measure of the effect of QSS. It is the percentage of the average precision when comparing and evaluating specific combinations of CNNs with ISs using different QSS.

It is computed from $Precision@NR$ *matrix*:

$$\mathrm{MQ}_{\mathrm{Precision(CNNs,IDBs)}} = Percentage_{\forall QSS}\left(\sum\left(Average_{\forall QSS}(MaxPer)_{\forall CNN\ \forall IDB}\right)\right),$$

where $MaxPer = \frac{COUNT_c\{c=max(Precision@NR)\}}{Number\_of\_Classes}$

$MaxPer$ represents the percentage of classes that achieved maximum precision using either 1-QSS or 2+QSS where 'c' represents a class. It first determines the classes for each QSS that achieves maximum precision, counting these classes, and then calculating the percentage of these classes out of the total number of classes. After computing $MaxPer$ for each QSS across all ISs and CNNs under interest, the average is taken based on QSSs. These averages are summed, and the percentages of these averages from their sum are depicted in the pie chart in Fig. 8 and Fig. 9.

This measure is applicable to both class-based and non-class-based retrieval. The difference between precision and $\mathrm{MQ}_{\mathrm{Precision}}$ is that precision is used for a single IS while $\mathrm{MQ}_{\mathrm{Precision}}$ is used for measuring the average performance of MQDIR while applying multiple ISs and CNNs in the experiments. Table 4 presents the percentage of instances where each QSS achieves maximum precision for each IS and CNN. The pie charts in Fig. 8 show that the performance generally increases as the number of QSS increases and Fig. 9 shows the superiority of the MQ approach over single query approach (the yellow curve) for the ten experiments. From the previous outcomes, it is assumed that the more relevant images are obtained using R and some/all 2+QSS.

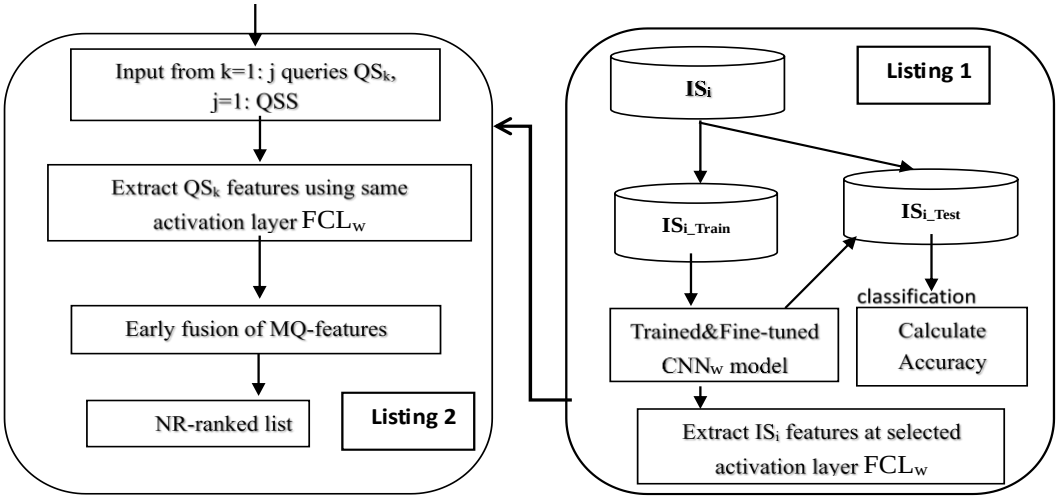


Fig. 1 MQDIR system End to End Process.

Listing 1 MQDIR FE Algorithm	**Listing 2** MQDIR Retrieval Algorithm
Inputs: IS= {IS_i} set of labeled ISs, i=1,…, 5, set of CNN_w, w=1,2 Outputs: Global Feature Vector FV at FCL_w Steps: For each IS_i in IS For each CNN_w a. Fine tune an input $CNN_w(IS_i)$ b. Split IS_i into IS_{i_Train} and IS_{i_Test} c. Train $CNN_w(IS_i)$ using IS_{i_Train} d. Calculate classification accuracy and save information & output e. Extract the features for IS_i using activations at FCL_w f. Construct and save FV_{ISi} g. Index images using generated FV_{ISi}	Inputs: Given IS_i, N is the number of classes in IS_i, FV_{Isi}: feature vector of IS_i, QS, QSS, NR: Number of top relevant retrieved images, Outputs: NR-relevant images, MQ_Precision for IS_i Steps: 1. Load $CNN_w(IS_i)$ information and FV_{Isi}. 2. Load QS for each class i (i=1,…,N). 3. for i=1: N for j=1: QSS a. For Each Query Image QS_k (k = 1 to j) ○ Retrieve QS_k index of from the augmented data store. ○ Compute FV_{QSk} by extracting features using FCL_w b. Construct and normalize MQ_FV using FV_{QSk} c. Apply MQ-Average on MQ_FV d. Compare MQ_FV with FV_{ISi} using Euclidean distance. e. Sort distances to identify the most relevant images. f. Retrieve NR-images based on the closest to MQ_FV. g. Construct matrix Precision@NR(i, j) of size N x QSS. 4. Calculate the overall precision measure MQ_Precision using Precision@NR (i, j).

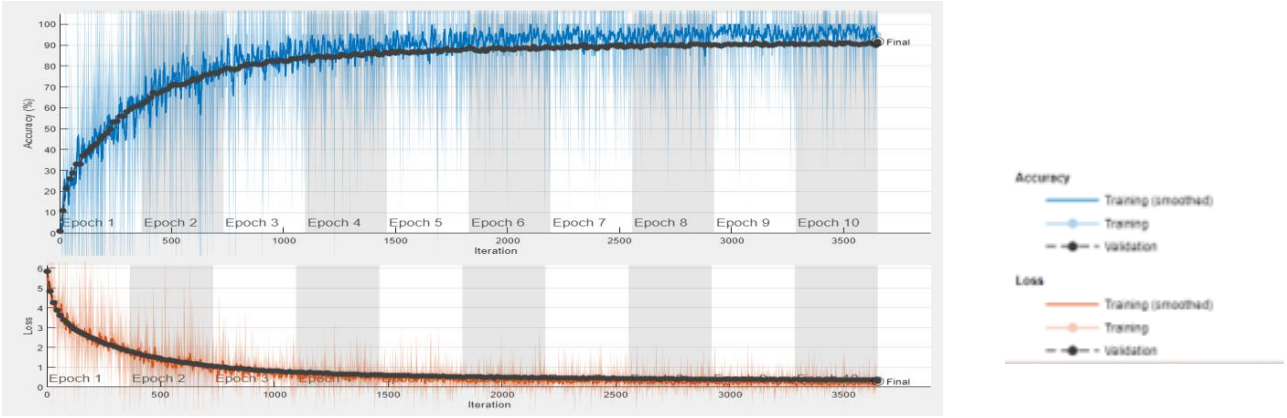


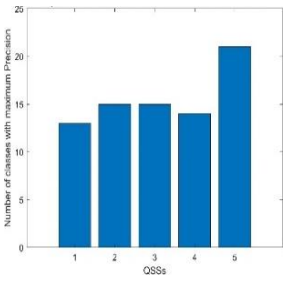
Fig. 2 Training progress for ResNet18 (Caltech101 IS) (validation accuracy 92.24%, average run time 208 minutes, VF=15, piecewise learning rate).

Table 1. Characteristics of ISs.

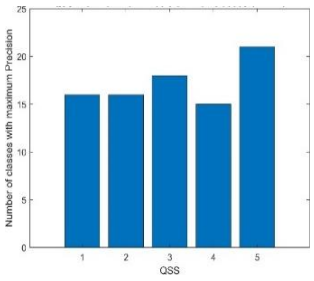
IS	DTD	KTH- TIP	Caltech101	Flowers.	GHIM
Resolution	300x300 up to 640x640	200x200	80x300 upto 3481x 2955	159x240 up to 500x441	400 × 300 and 300 × 400
Class	47	10	101	5	20
count	5640	810	9144	3670	10000
Scene Examples	Texture Images	Texture Images: aluminium_foil, cotton, linen	Texture & others airplanes, bonsai, sky,grass ,flowers,dotted, striped	Textures: daisy, rose, sunflower, tulip, dandelion	Texture & others buildings, festivals
type	color	gray	color	color	color

Table 2. Fine tuning Training Trials.

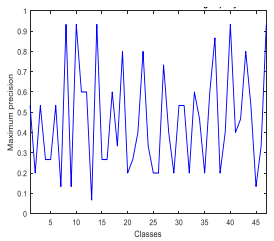
IS	Trial	epoch	VF	validation accuracy	Elapsed Time minutes	Final learn rate
DTD	b) A	15	15	57.71	146	4e-6
	d) R	10	15	63.74	120	2e-5
Caltech 101	b) A	15	15	90.00	225	4e-6
	d) R	10	15	92.24	208	2e-5
GHIM-10K	b) A	15	15	97.40	481	4e-6
	d) R	10	15	98.30	321	2e-5
Flowers	b) A	15	15	87.48	68	4e-6
	d) R	15	15	91.16	82	4e-6
KTH_ TIPS	b) A	15	15	96.88	11	4e-6
	d) R	15	15	96.88	12	4e-6



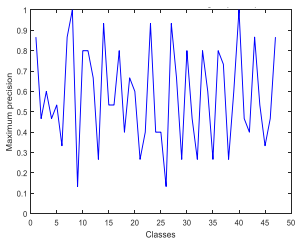
i. A



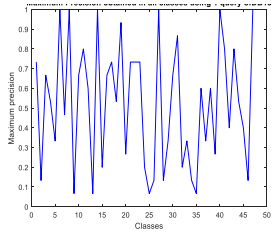
ii. R



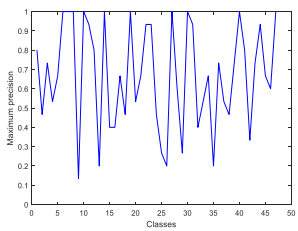
iii. A: 1-QSS



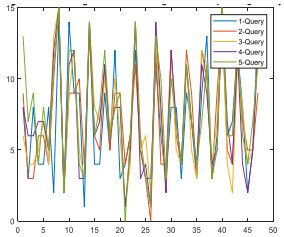
iv. A: 2+ QSS



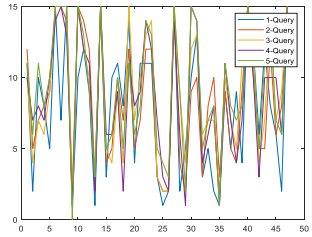
v. R: 1-QSS



vi. R: 2+ QSS.

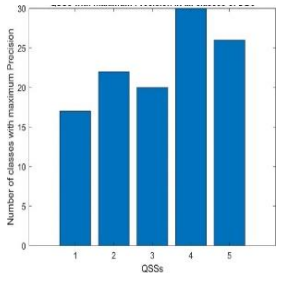


vii. A

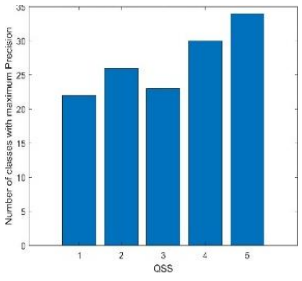


viii. R

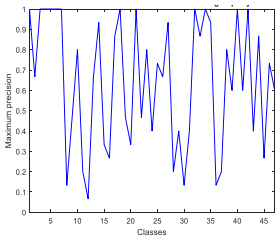
Fig. 3 DTD Precision using different QSS.



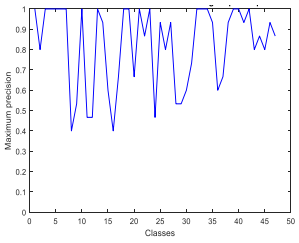
i. A



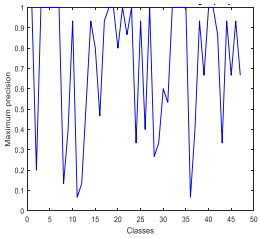
ii. R



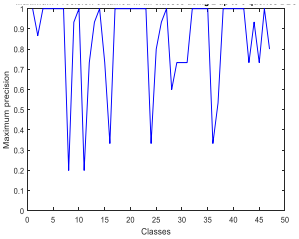
iii. A: 1-QSS



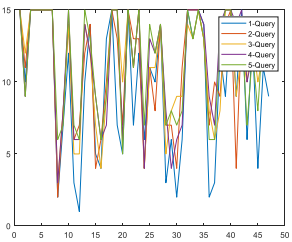
iv. A: 2+ QSS



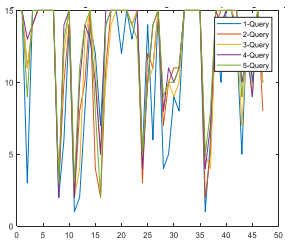
v. R: 1-QSS



vi. R: 2+ QSS



vii. A



viii. R

Fig. 4 Caltech101 Precision using different QSS.

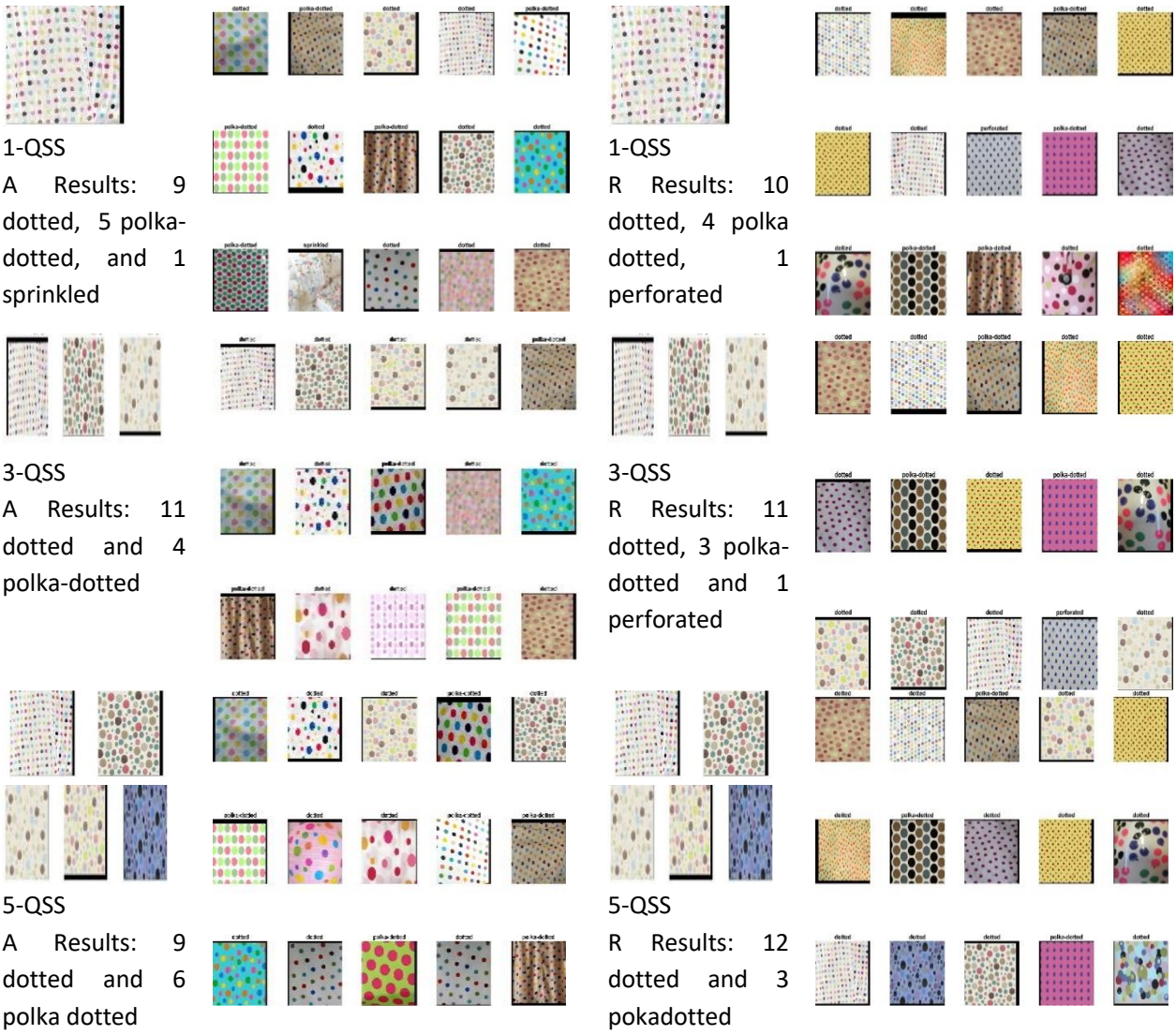
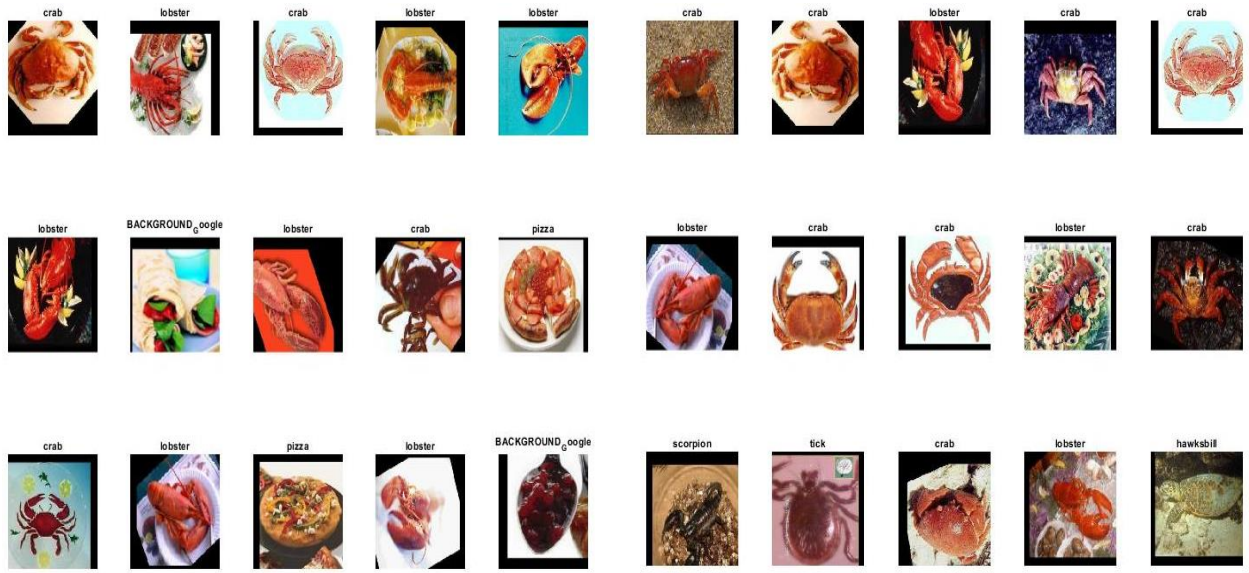


Fig. 5 Dotted Class-based retrieval (DTD IS).

Table 3. R-Average Precision showing Class- and Non-Class-based QSS Retrieval for some.

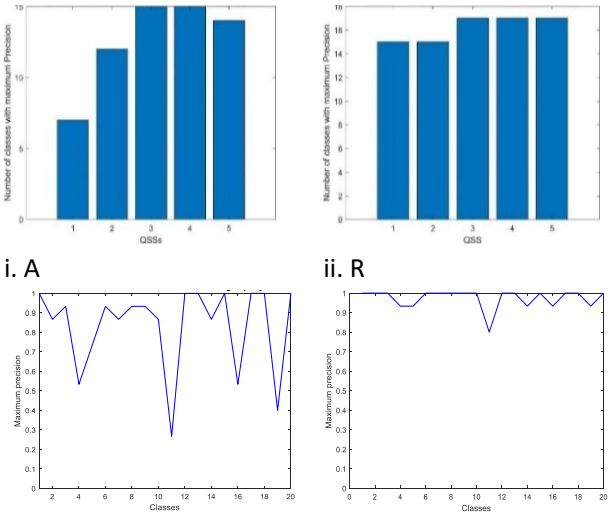
class	Non-class-based retrieval		Class-based retrieval	
	1-QSS	*2+ QSS*	*1 QSS*	*2+ QSS*
banded	4	13	2	5
braided	14	15	9	12
bubbly	13	13	13	13
bumpy	10	13	6	12
checkered	15	9	15	7
cobwebbed	10	15	10	15
cracked	13	15	12	15

class	Non-class-based retrieval		Class-based retrieval	
	1-QSS	*2+ QSS*	*1 QSS*	*2+ QSS*
dotted	15	15	9	11
fibrous	13	13	9	9
freckled	15	15	15	15
honeycombed	15	15	14	15
knitted	13	15	7	10
Paisley	11	11	9	11
pitted	3	8	1	5



1-QSS 2-QSS

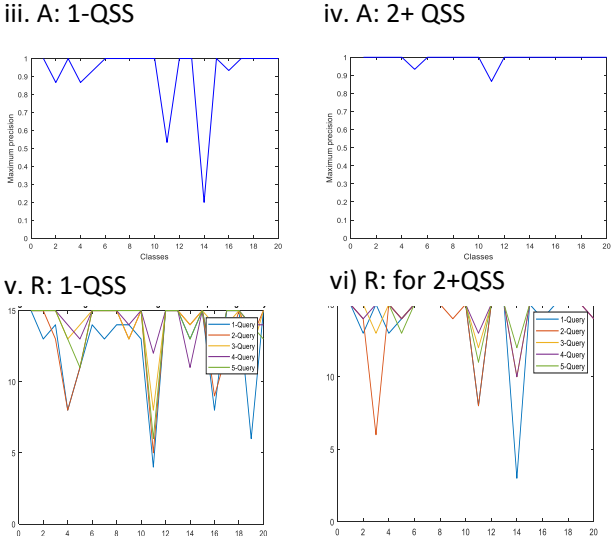
Fig. 6 MQDIR Retrieval results for class crab using R.



i. A ii. R

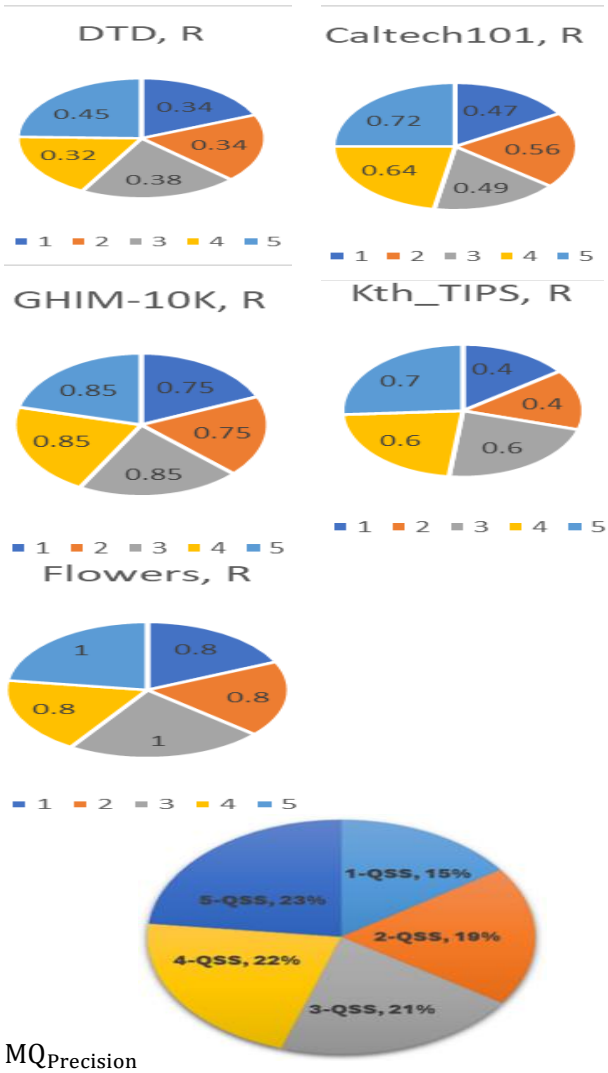
iii. A: 1-QSS iv. A: 2+ QSS

v. R: 1-QSS vi) R: for 2+QSS



vii. A viii. R

Fig. 7 GHIM-10K Precision for different QSS.



$MQ_{Precision}$

Fig. 8 MaxPer and for $MQ_{Precision}$ for all ISs.

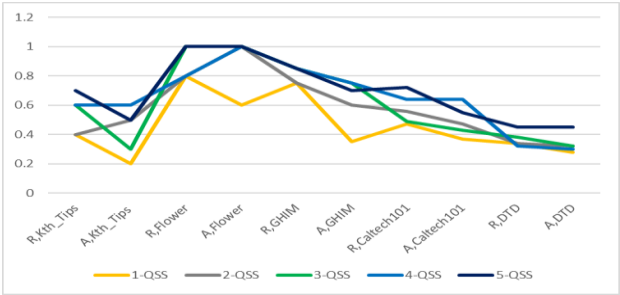


Fig. 9 MQDIR $\mathrm{MQ}_{\mathrm{Precision}}$

Table 4. (Class-Based) $MaxPer$ for NR=15.

	DTD		Caltech101		GHIM-10K		Flowers		Kth_TIPS	
	A	R	A	R	A	R	A	R	A	R
1-QSS	.28	.34	.37	.47	.35	.5	.6	.8	.2	.4
2-QSS	.32	.34	.47	.56	.6	.75	1	.8	.5	.4
3-QSS	.32	.38	.43	.49	.75	.85	1	1	.3	.6
4-QSS	.3	.32	.64	.64	.75	.85	1	.8	.6	.6
5-QSS	.45	.45	.55	.72	.7	.85	1	1	.5	.7

3.4. Discussion

Although there is no exact match for MQDIR, a detailed thorough comparison that highlights the main performance differences between MQDIR and current similar approaches are presented for context. In [23] a deep learning approach using FERET benchmark IS, the classification accuracy for ResNet18 and AlexNet are 96.3% and 93.3%, respectively while using our approach reached 100% for 1-QSS and 2-QSS in R. In the case of 2-QSS more images of class Faces than 1-QSS are retrieved where the query images include the first object in Fig. 10.

In [24], with Caltech101 IS, the authors achieved an accuracy of 0.88 for the top ten retrieved images using VGG-16 and a similarity score. Considerable time was spent before training to construct a gravitational field model to add the similarity score label for each image. MQDIR with Caltech101 achieved $\mathrm{MQ}_{\mathrm{Precision}}$ 72% of classes gaining the maximum class percentage for the top fifteen class-based retrieved images, with many of these classes reaching 100% precision, as shown in Fig. 4-(vi) and Table 4.

In [25], the models ResNet18 and SqueezeNet CNNs were used, with GHIM-10K reporting a 93% average precision using ResNet18. MQDIR achieved 85% of the GHIM-10K classes reach maximum precision in class-based retrieval, which exceed 90% using non-class-based retrieval. As shown in Fig. 7-(vi) and Table 4, 85% of the classes reached 100% precision in 2+QSS retrieval. In [26], features were extracted using block-level DCT and GLCM, with other features computed by taking the difference between the original image plane and the DC coefficients based on the reconstructed image plane. The average precision for the top ten images was 77.50% for GHIM-10K. MQDIR using R reached a maximum precision of 85% for 2+QSS, with many classes reaching 100% precision. Regarding texture retrieval in [5], CNN features were extracted from diverse pre-trained models including ResNet50, ResNet101, and AlexNet, with SVM used as a classifier for texture classification. Performance was investigated using the KTH-TIPS and Flowers texture ISs, with claimed accuracy ranging from 85% to 95%. MQDIR (Table 4) achieved 100% precision for the Flowers IS and 85% for GHIM-10K using class-based retrieval, which increased with non-class-based retrieval.

For texture recognition, description, and segmentation, the research in [27] obtained classification accuracy of 58.8% using a deep CNN called VGG-M and 62.9% using VGG-VD. MQDIR achieved a classification accuracy of 62.77% for DTD. In [28], SIFT features were fused with AlexNet features, accompanied by the MQ approach. They experimented with the Oxford (89%) and Paris (96.64%) ISs using 5-QSS. For GHIM-10K, 85% of classes reached maximum precision, and 100% for the Flowers IS. Assessing these results, MQDIR based on MQ approach demonstrates reliable retrieval performance compared to other methods.

MQDIR approach can be applied to any large ISs, however MQDIR will face computational difficulties. This can be accomplished utilizing more complex CNN with more layers and consequently will require larger computational resources so that the elapsed time in Table 2 may decrease while using larger patches, VF, and epochs. MQDIR is expected to perform better than the single query approach since it provides more information about the object through MQ, this would be helpful particularly in the context of noisy IS.

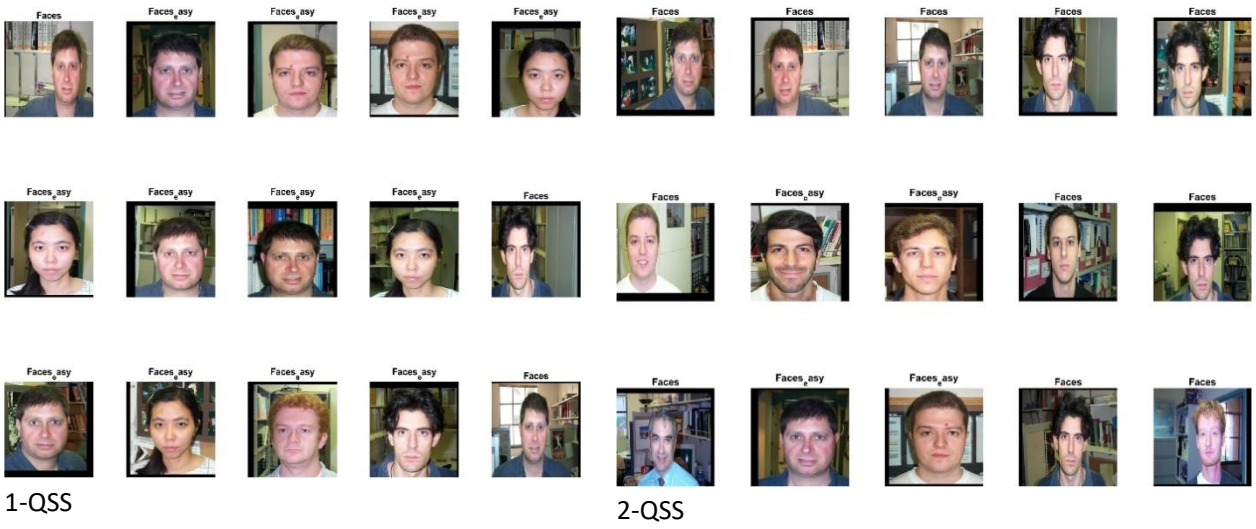


Fig. 10 Caltech101 IS class Faces 1-QSS and 2-QSS using R.

4. Conclusion

There are limited studies investigating the use of multi-query approach for deep learning in CBIR. This paper explores this approach introducing the MQDIR methodology which is experimented with various standard ISs of varying complexity and pretrained CNNs. The study demonstrates that the MQDIR approach is effective when using selected semantically related queries and class-based and non-class-based retrieval. It proved that the MQ approach outperforms the single-query approach using deep features. Additionally, this study introduces a new precision-based measure for evaluating MQ retrieval performance with different ISs and CNNs. The findings suggest that future research could further explore the performance of more complex CNN architectures on different ISs (possibly heterogeneous datasets). For instance, R showed better classification accuracy and retrieval performance compared to A, highlighting its potential for improved CBIR systems.

5. References

1. **Srivastava, D., Bakthula, R., and Agarwal, S. (2019).** Image classification using SURF and bag of LBP features constructed by clustering with fixed centers. Multimed Tools Appl **78:14129–14153.**
2. **Enver A. Abdurrahim T., and Ugur E.c (2024).** MLMQ-IR: Multi-label multi-query image retrieval based on the variance of Hamming distance**.**
3. **Abeer, A., Ouiem, B., Mohamed, and M. (2020).** Multiple Query Content-Based Image Retrieval Using Relevance Feature Weight Learning. J Imaging **6(1):2.**
4. **Cabir, V., and Enver, A. (2020).** Deep multi query image retrieval. Signal Process. Image Commun. **88:115970.**

5. **Philomina, S., and Uma, V. (2019).** Deep Learning based Feature Extraction for Texture Classification. International Conference on Computing and Network Communications (CoCoNet'19).
6. **Sung-Wook, H., Taekyeong, L., Hyunbin, K., Hyunwoo, C., Jong, G., and Hwanmyeong, Y. (2022).** Classification of wood knots using artificial neural networks with texture and local feature-based image descriptors. Holzforschung; **76(1): 1–13.**
7. **Ibtihaal, M., Sadiq, H., and Basheera, M. (2021).** Content-based image retrieval: A review of recent trends. Coge-nt Engineering, **8:1, 1927469.**
8. **Shiv, R. (2021).** A Decade Survey of Content Based Image Retrieval using Deep Learning. IEEE Transactions on Circuits and Systems for Video Technology.
9. **Muhammad, A., Norhalina, S., Fazli Whid, Muhammad, A., Ali, S., and Mukhtaj Khan (2022).** Comparative Analysis of Recent Architecture of Convolutional Neural Network. Mathematical Problems in Engineering, Volume **2022: 9** pages**,** Article ID **7313612.**
10. **Li, L., Jie, C., Paul, F., Guoying, Z., Rama, C., and Matti, P. (2019).** From BoW to CNN: Two Decades of Texture Representation for Texture Classification. International Journal of Computer Vision, **127:74–109.**
11. **Zheng, Q., Tian, X., Yang, M., and Wang, H. (2019)** Differential learning: A powerful tool for interactive content-based image retrieval. Engineering Letters **27(1):202–215.**
12. **Babenko, A., and et al. (2014).** Neural codes for image retrieval. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds) Computer Vision – ECCV. Lecture Notes in Computer Science, Vol **8689**. Springer, Cham.
13. **Şaban, Ö. (2020).** Stacked auto-encoder based tagging with deep features for content-based medical image retrieval, Expert Systems with Applications **161.**
14. **Zheng, Z., and Zhong, Zho. (2020).** Low Dimensional Discriminative Representation of FCL Features Using Extended LargeVis Method for High-Resolution Remote Sensing Image Retrieval. Sensors **20(17), 4718.**
15. **Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., and Fuad, E. A. (2017).** A survey of deep neural network architectures and their apps. Neurocomputing, **234:11–26.**
16. **Elena, S., Katarina, T., Ivica, D., and Suzana, L. (2013).** Multi-Query Content-based Medical Image Retrieval. The 10th Conference for Informatics and Information Technology.
17. **Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. (2014).** Describing textures in the wild. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, **3606-3613.**
18. **Timothy, T. Y., Ma, D., Cole, J., Ju, M. J., Beg. M. F., and Sarunic. M. V. (2021).** Spectral bandwidth recovery of optical coherence tomography images using deep learning. 12th International Symposium on Image and Signal Processing and Analysis **67-71.**
19. **Li, F., Fergus, R., and Perona, P. (2007).** Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. Computer Vision and Image Understanding **106(1):59–70.**
20. **Liu, G.-H., Yang, and J.-Y., Li, Z. (2015).** Content-based image retrieval using computational visual attention model. pattern recognition **48(8):2554-2566.**
21. **Kaiming, H., Xiangyu, Z., Shaoqing, R., and Jian, S. (2016).** Deep Residual Learning for Image Recognition. IEEE Conf. Proceedings on Computer Vision and Pattern Recognition **770-778.**
22. **Sumair A., Muhammad B., Muhammad U. K., and Fatima A. (2020).** Deep Learning-based Automatic Morphological Classification of Leukocytes using Blood Smears. Proc. of the 2nd International IEEE Conference on Electrical, Communication and Computer Engineering (ICECCE).
23. **Kadhim T. A., Zghal N. S., Hariri W. and Ben Aissa D (2022).** Face Recognition in Multiple Variations Using Deep Learning and Convolutional Neural Networks. IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), **pp. 305-311, doi: 10.1109/SETIT54465.2022.9875530.**

24. **Nitish, S. K., Dheevatsa, M., Jorge, N., Mikhail, S., Ping, T., and Peter, T. (2017).** on Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. ICLR.
25. **Ali, A. (2021).** Pre-trained CNNs Models for Content-based Image Retrieval. International Journal of Advanced Computer Science and Applications **12(7).**
26. **Varish, N., and Pal, A.K. (2018).** A novel image retrieval scheme using gray level co-occurrence matrix descriptors of discrete cosine transform-based residual image. Applied Intelligence **48: 2930–2953.**
27. **Cimpoi, M., Maji, S., Kokkinos, I., and Vedaldi, A. (2015).** Deep filter banks for texture recognition and segmentation. IEEE Conference on Computer Vision and Pattern Recognition, **3828-3836.**
28. **Huang, S., Hang, and H.-M. (2017).** Multi-query image retrieval using CNN and SIFT features. IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conf. **1026-1034.**