UGQE: Uncertainty Guided Query Expansion

Firat Oncel[1(✉)], Mehmet Aygün[1,2], Gulcin Baykal[1], and Gozde Unal[1]

[1] Istanbul Technical University, Istanbul, Turkey
oncelf@itu.edu.tr
[2] The University of Edinburgh, Edinburgh, Scotland

Abstract. Query expansion is a standard technique in image retrieval, which enriches the original query by capturing various features from relevant images and further aggregating these features to create an expanded query. In this work, we present a new framework, which is based on incorporating uncertainty estimation on top of a self attention mechanism during the expansion procedure. An uncertainty network provides added information on the images that are relevant to the query, in order to increase the expressiveness of the expanded query. Experimental results demonstrate that integrating uncertainty information into a transformer network can improve the performance in terms of mean Average Precision (mAP) on standard image retrieval datasets in comparison to existing methods. Moreover, our approach is the first one that incorporates uncertainty in aggregation of information in a query expansion procedure.

Keywords: Uncertainty · Image retrieval · Self attention

1 Introduction

Image retrieval methods rely mainly on a projection from a high dimensional input data to a relatively low-dimensional vector space.

Due to several sources of error such as occlusions or loss of information during projection to the vector space, image search is enriched with a query expansion idea that depends on constructing a latent model, which is based on aggregating a collection of responses from an initial query [7,29]. On the other hand, in the recent years, image retrieval methods are dominated by Convolutional Neural Networks (CNNs), which replace hand-crafted features in feature extraction phase of image retrieval systems [4,13,22,24].

As important as a role feature extraction plays in performance of image retrieval systems, a set of related tools such as database augmentation [31], query expansion [7], hashing [8] and so on play crucial roles in image retrieval systems. Particularly, Query expansion (QE) is regarded as one of the most powerful tools, as it increases the performance of an image retrieval system no matter how the features are extracted [7,14]. The basic idea of a QE method is to enhance the

M. El Yacoubi et al. (Eds.): ICPRAI 2022, LNCS 13363, pp. 109–120, 2022.
https://doi.org/10.1007/978-3-031-09037-0_10

quality of the search vector through an augmentation of the query vector space using some priors. The latter is provided by an initial search that results in a collection of vectors that lead to a richer latent feature representation of the query. One of the main limitations of QE algorithms [3,7,13,24] is that the assigned weights to neighbors of the query are monotonic. In the recent work of Gordo et al. [14], a self-attention mechanism was used via transformers [32] in order to predict weights which do not have to be monotonic so that irrelevant neighbors, which are not true neighbors, can be eliminated.

While neural networks have the flexibility to create and assign different weights in QE algorithms, they are famous for their overconfidently wrong predictions [15], which might hurt the quality of the expanded query feature vector. In recent years, there has been a lot of interest on how to incorporate uncertainty estimation into deep neural network models [12,18,21,28] to alleviate their problem in making overconfident predictions. These works enable integrating the ability of saying "I am not sure about this prediction". In this paper, we integrate a dedicated pairwise uncertainty estimation between a query and each of its neighbors in creation of an enriched image representation. The latter is in terms of image features relative to the original extracted features which in combination provide a tool in generating more powerful expanded queries. To that end, we design an uncertainty-guided transformer encoder, which relies on and expands on the self-attention-based Learnable Attention-based Query Expansion (LAttQE) model by [14] via incorporating an uncertainty that is estimated with the Evidential Deep Learning (EDL) framework [28].

Our proposed new module, which is called the Uncertainty Guided Query Expansion (UGQE), first takes a query and features of retrieved top-k neighbors and estimates the uncertainty in whether the neighbors are from the same landmark with the query or not, using the EDL setting within a transformer encoder architecture. Next, our model utilizes the obtained uncertainty information in order to generate a new feature representation via concatenating transformed features and original features. Finally, the LAttQE model is used to form the expanded query by aggregating these new generated features of the nearest neighbors.

Our main contributions in this paper can be summarized as follows: (i) a method to create uncertainty guided features to enrich the original image representations; (ii) a demonstration of how to use EDL for quantifying the pair-wise uncertainty. Our experiments demonstrate that the proposed UGQE method increases the performance of the traditional QE methods and outperforms the LAttQE model when the uncertainty is integrated, in most settings.

2 Related Work

Image Retrieval and Query Expansion: Image retrieval systems generally consist of two parts. The first part extracts a representation from an image, and the second part performs the search by utilizing a distance measure in the representation space. Before the revival of (deep) neural networks, the representation

extraction part was based on hand-crafted feature engineering methods such as SIFT [20], Fisher Vectors [23] and VLAD [16]. Further extensions of those methods are also introduced for improving whole retrieval pipelines [3]. Recently, the feature extraction phase is dominated by deep neural networks. Bottleneck features, or activations at the output of certain later layers are used as the representations for the input data. In an early work by Babenko et al. [4], pre-trained networks are used as feature extractors. However, in recent works, CNNs are fine-tuned in unsupervised [24] and supervised [2,13,22] settings for learning more efficient representations. Generally, the supervised methods use noisy data for fine-tuning. To improve the feature representation of query images, Chum et al. [7] introduced the idea of QE for image retrieval, which first appeared in text retrieval systems [6,26]. The QE idea is very simple: after the first search, highly ranked images are filtered with a spatial verification step to retain high quality results. Next, using the original query description along with the descriptions of retrieved presumably high quality results, a new expanded query description is generated. A number of methods are suggested for generating the newly expanded vector, however the most commonly used one just averages the initial query and the high quality ranked ones for generating the expanded vector.

While the QE methods were first introduced for hand-crafted features in image retrieval systems, deep learning based image retrieval systems also used QE in their pipelines for improving their performance since QE methods do not rely on how features are extracted. Moreover, lately, state of the art results are obtained with QE methods that are combined with deep representations in the feature extraction phase [13,24,30], or more recently, with learning the QE weights [14] using the attention mechanism [32].

Transformers - Visual Transformers: The transformer [32] architecture has become the default choice for most of the natural language processing tasks [5,9,19]. Transformer architectures are capable of grasping the relationship between sequential inputs using a self-attention mechanism. Transformers are also lately used in computer vision tasks [10], via representing images as sequence of patches. El-Nouby et al. [11] extends visual transformers to a metric learning setting to perform image retrieval. While this work focuses on the initial representation of images, in our work we focus on how to expand the initial representations using transformers in the QE framework.

Graph Attention Networks [33] are used in the metric learning setting to weigh the feature vectors in a batch [27]. Most recent and relevant work in query expansion that includes the attention setting is by Gordo et al. [14]. They utilized a self attention mechanism to weigh the nearest neighbors of the query image for integrating their information. Our work expands on the transformer idea, strengthening it with the uncertainty information. The uncertainty that we introduce is captured from the neighborhood relationships between feature vectors, and integrated into generation of an expanded query feature vector.

Uncertainty in Deep Learning: Predicting model uncertainty is an emerging field in deep learning. MC Dropout [12] is one of the earliest and widely

utilized works in the related literature. In this method, a dropout mechanism is applied not only in training time but also in test time, and a multitude of predictions are averaged to get the prediction, and the variance among predictions is used to calculate the uncertainty. This approach presents high computation time requirements since multiple forward passes are needed. Deep Ensembles [18] and their variants are also related to MC Dropout, bearing a similar approach in incorporating an indirect uncertainty estimation into deep neural networks. In contrast to deep ensemble methods, a line of recent research work, known as Evidential Deep Learning directly estimates both data and model uncertainty. The latter, which is also known as epistemic uncertainty, deals with how certain a neural network can be when making predictions. Sensoy et al. [28] estimate the classification uncertainty collecting evidence for each class by placing a Dirichlet distribution on the class probabilities instead of a softmax to allow the neural network models to directly quantify the uncertainty in their outputs. Amini et al. [1] extends this idea into continuous outputs by placing an Inverse Gamma distribution at the end of the regression task. In our work, we integrate the uncertainty estimation between a query image and its top-k neighbors to generate new features. With these features, the image representations are enhanced while the estimated uncertainty is integrated into the query expansion procedure. To our knowledge, our work is the first to introduce an uncertainty guidance through gauging the reliability of neighboring feature vectors into the QE framework.

3 Method

As we deploy an uncertainty quantification through the Evidential Deep Learning approach into the Attention-Based QE setting, we give a brief overview of both frameworks.

Attention-Based Query Expansion Learning: LAttQE [14] utilizes the self-attention mechanism to form the expanded query based on the original query and its k-nearest neighbors by predicting the weights for each of the respective feature vectors. They formulate the problem as a metric learning problem. Let ϕ be a CNN feature extractor, which takes an input image, and transforms it into a D-dimensional feature vector. The query image is denoted by q, and its feature vector is denoted by $\mathbf{q} = \phi(q)$. The positive pair, the i-th negative pair and the k-th nearest neighbor of the query image are denoted as $\mathbf{p}$, $\mathbf{n_i}$ and $\mathbf{d_k}$, respectively. $\hat{\mathbf{q}}$ can be formulated as: $\hat{\mathbf{q}} = \mathbf{w_q q} + \sum_{\mathbf{n=1}}^{\mathbf{k}} \mathbf{w_{d_k} d_k}$, where w_q and w_{d_k} are assigned weights to the query and its neighbors. Additionally, the transformer outputs whether the retrieved neighbors are from the same landmark as the query or not. To account for that, an additional variable y is defined such that y is set to 1 if we have a positive pair, and set to 0 otherwise. The corresponding contrastive loss can be formulated as follows:

$$L_{qe}(\hat{\mathbf{q}}, \mathbf{f}, y) = y \left\| \hat{\mathbf{q}} - \mathbf{f} \right\| + (1 - y) \max(0, m - \left\| \hat{\mathbf{q}} - \mathbf{f} \right\|), \tag{1}$$

where $\mathbf{f}$ is either $\mathbf{p}$ or $\mathbf{n_i}$, and m is a selected margin.

Evidential Deep Learning (EDL): For estimating the uncertainty or inversely the reliability of a neighbor in contributing to the feature description of a query, we utilize the objective proposed in [28]. Let us denote the evidence collected from the k-th class of a multi-class (e.g. K-class) classification problem as e_k, and Dirichlet distribution parameters as $\alpha_k = e_k + 1$. Then the Dirichlet strength is given by $S = \sum_{k=1}^{K} \alpha_k$. Belief masses are defined as $b_k = \frac{e_k}{S}$, and the expected probability for the k-th class is $\hat{p_k} = \frac{\alpha_k}{S}$. Finally, the uncertainty in the prediction can be calculated as the residual belief remaining when we subtract the sum of our beliefs from unity: $u = 1 - \sum_{k=1}^{k} b_k$. Given N input samples and labels, the classification uncertainty can be quantified by minimizing the following objective function:

$$\mathcal{L}(\Theta) = \sum_{i=1}^{N} \mathcal{L}_i(\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{K} (y_{ij} - \hat{p}_{ij})^2 + \frac{\hat{p}_{ij}\,(1 - \hat{p}_{ij})}{(S_i + 1)} \tag{2}$$

where Θ denotes neural network parameters, y_i is a one-hot vector encoding of ground-truth of sample x_i with $y_{ik} = 0$ and $y_{ij} = 1$ for all $k \neq j$, and i denotes the index of a data sample for $i = 1, ..., N$. Here, the objective entails minimization of the sum of the squared prediction errors and an estimate of the variance in the second term to obtain the evidential distribution parameters. Although this is a non-Bayesian neural network approach, by placing evidential priors over the classification output, this framework outputs the uncertainty u in the form of what is left beyond our beliefs and collected evidence that are based on the assumed evidential distribution.

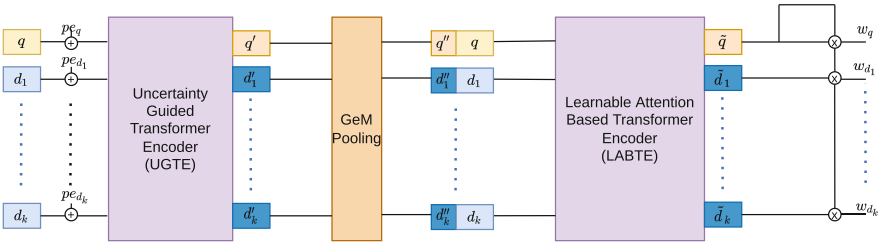

Fig. 1. Uncertainty Guided Query Expansion: UGTE takes input features, calculates the uncertainty between the query and each neighbor, and outputs the uncertainty guided features. Then, LABTE takes a combination of new features and original features, and outputs the weights for the query and its nearest neighbors to generate the expanded query.

Uncertainty Guided Query Expansion Learning: Leveraging on the evidential distribution idea, we propose the Uncertainty Guided Query Expansion (UGQE) model, which adapts and fuses ideas from both the EDL and LAttQE to create an improved attention-based architecture that enables and exploits

uncertainty learning in query expansion. Our complete UGQE model is depicted in Fig. 1. UGQE involves two transformer encoders, the first one generates the uncertainty guided features. The first transformer encoder integrates a pairwise uncertainty quantification, by employing a $K = 2$-class evidential classification, which outputs a target label y that is 1 if the i-th neighbor $\mathbf{d_i}$ of the query $\mathbf{q}$ is relevant, and 0 otherwise. The first transformer is trained end-to-end with the second transformer encoder, i.e. the LAttQE architecture, to produce the final weights that are used in the query expansion.

The details of the UGQE algorithm is described in Algorithm 1. Input feature vectors, which are the query ($\hat{\mathbf{q}}$) and its top-k nearest neighbors ($\mathbf{n_1}, \mathbf{n_2}, ..., \mathbf{n_k}$), are 2048-Dimensional. The output vectors of this model is also 2048-D, same as the input size. Our model takes these features (line 1 in Algorithm 1), as inputs and then sends the output vectors to the GeM Pooling layer [24], which reduces the dimensions of vectors from 2048-Dimensions (2048D) to 512D (line 2 in Algorithm 1). As the point of this model is to learn informative features, we observe that the reduced features were able to retain the useful information in the original features. These 512D features, which are later concatenated to the original features, are fed to a multilayer perceptron (MLP). Then the EDL Loss (Eq. 2) is minimized, and an uncertainty estimate is produced for each neighbor, which is a quantity that signifies a kind of confidence in the neighbor status of each neighbor. Then, the obtained uncertainty feature vectors are concatenated to the original features to be input to the second transformer, i.e. the LAttQE model (line 3 in Algorithm 1). LAttQE is trained end-to-end with the first transformer encoder, and the output of the overall model are the weights of the query and its nearest neighbors that provide construction of the final expanded query ($\hat{\mathbf{q}}$) (lines 4–10 in Algorithm 1).

Algorithm 1: Uncertainty Guided Query Expansion (UGQE)

input : Learnable Attention Based Transformer Encoder ($LABTE$)
Uncertainty Guided Transformer Encoder ($UGTE$)
features of query and k-nearest neighbors: $F = \{\mathbf{q}, \mathbf{d_1}, \mathbf{d_2}, ..., \mathbf{d_k}\}$
positional encodings: $PE = \{\mathbf{pe_q}, \mathbf{pe_{d_1}}, \mathbf{pe_{d_2}}, ..., \mathbf{pe_{d_k}}\}$
output : expanded query: $\hat{\mathbf{q}}$

1 $F' \leftarrow UGTE([F; PE])$
2 $F' \leftarrow GeMPooling(F')$ ▷ Uncertainty guided features
3 $\{\tilde{\mathbf{q}}, \tilde{\mathbf{d_1}}, \tilde{\mathbf{d_2}}, ..., \tilde{\mathbf{d_k}}\} \leftarrow LABTE([F'; F])$
4 **for** $i \leftarrow 1$ **to** K **do**
5 $\quad w_i \leftarrow normalizeddotproduct(\tilde{q}, \tilde{d}_i)$
6 **end**
7 $\hat{\mathbf{q}} \leftarrow \mathbf{q}$
8 **for** $i \leftarrow 1$ **to** K **do**
9 $\quad \hat{\mathbf{q}} = \hat{\mathbf{q}} + w_i \mathbf{d_i}$
10 **end**
11 **return** $\hat{\mathbf{q}}$

Implementation Details: UGQE model has 2 attention layers (each one with 64 heads) and created features are 512D. Only EDL Loss is used to train this model. We use Adam optimizer [17] with initial learning rate of $1e^{-4}$, a weight decay of $1e^{-6}$, and an exponential learning rate scheduler with a decay of 0.99. Then we concatenate the original features with the created ones to train the second self-attention model, which is the LAttQE, which has 3 attention layers (each one with 64 heads). We use the contrastive loss in Eq. (1), with a margin parameter of 2.1, and the binary-cross entropy loss to train this model. In the implementation of the EDL, we use the exponential activation layer at the end of our uncertainty MLP block. We use 2048 queries per epoch and 15 negative samples, which is chosen empirically, for each positive sample, selected from a pool of 40000 samples. Choice of the number of negative samples for a positive sample and the margin is essential as the loss is usually 0 with a low number, i.e. 5 or 10, of negative samples, and a low margin, i.e. 0.1. Queries and the pool of samples are updated at every epoch. We also use Adam optimizer with initial learning rate of $1e^{-4}$, a weight decay of $1e^{-6}$, and an exponential learning rate scheduler with decay of 0.99 for the second transformer. We train our setting for 300 epochs with a batch size of 64, and 127 neighbors. We do not use neighborhood dropping, which is proposed in [14], as it does not improve the scores in our experiments. Both models have learnable positional encodings to integrate the positional information. As there is no publicly available official implementation of [14][1], we implemented the LAttQE architecture, which constitutes our baseline. Using the original hyperparameters reported in the paper[2], slightly lower performance values are obtained than the reported results. We report both results in our paper, however, we take our implementation as the baseline to compare it with the proposed UGQE.

4 Experiments

Training Dataset: We use rSfM120k dataset [24] for training the models in our experiments. The training part of the dataset has 91642 images and 551 landmarks, while validation part has 6403 images and 162 landmarks, distinct from the training ones. To extract features, we use ResNet101, which is trained on the Google Landmarks dataset [22], with GeM pooling and a whitening layer. Resulting feature vectors are 2048-D vectors. Single scale vectors are extracted at the original scale, with the input image size of 1024, while multi scale vectors are extracted at 3 scales; $1024 * (1, \sqrt{2}$ and $1/\sqrt{2})$, and then mean-aggregated. Both single scale and multi scale vectors are l_2 normalized. We use Python Image Retrieval Toolbox[3] to extract all features. We conduct our experiments

[1] https://github.com/filipradenovic/cnnimageretrieval-pytorch/issues/68.

[2] Unfortunately it was not possible to reproduce the performance results reported in the paper. This is also reported by researchers via issues opened in the github repo of the paper.

[3] https://github.com/filipradenovic/cnnimageretrieval-pytorch.

with both single scale and multi scale vectors, whereas training with single scale vectors gives slightly better results so that we report this setting.

Test Datasets: We conduct our tests over three publicly available datasets using the suggested test protocols as described in [25]. The training dataset and the test dataset do not contain overlapping images. We provide test results on the Medium (M) and Hard (H) protocols.

- Revisited Oxford ($\mathcal{R}$Oxford ($\mathcal{R}$Oxf)): 4993 images with 70 query images.
- Revisited Paris ($\mathcal{R}$Paris ($\mathcal{R}$Par)): 6322 images with 70 query images.
- Distractor set (1 million distractors ($\mathcal{R}$1M)): 1 million distractors, which are irrelevant from ($\mathcal{R}$Oxf) and ($\mathcal{R}$Par) datasets, are added to the two datasets to evaluate the performance in the harder setup.

All test feature vectors are extracted at the multiscale setting, as described in the training dataset part to make a fair comparison with the existing methods. We use the commonly used mAP (mean Average Precision) to evaluate the performance.

Comparison to State-of-the-Art: All experimental results are presented in Table 1. Results of AQE [7], AQEwD [13], DQE [3], αQE [24] and LAttQE (depicted in gray) are given as reported in [14]. Our implementation of the LAttQE is given by LAttQE*. In both settings, the UGQE improves the baseline method [14] in terms of the reported performance measure. On the ($\mathcal{R}$Oxf) dataset, the UGQE performs better than other existing methods, and gives similar performance on the ($\mathcal{R}$Par). Furthermore, compared to the traditional methods and the LAttQE, the UGQE gives more balanced results as the mean mAP outperform the traditional methods. There are some mixed performance outputs where the αQE outperformed the transformer based techniques in some scenarios. A possible explanation is that after some point as the number of neighbors taken into account in the expansion increases, the performance in ($\mathcal{R}$Oxford) decreases while the performance in ($\mathcal{R}$Paris) increases. This indicates that there is a trade-off in the performance of the two test datasets using traditional methods. However, even then it can be observed that the addition of the uncertainty guidance in the self-attention transformer framework helped UGQE surpass the performance results of the baseline in all settings. Although the results of LAttQE reported in [14] are not reproducable[4], the proposed UGQE outperforms the reproducable baseline method. Some sample visual results are given in Fig. 2, where the UGQE tends to retrieve irrelevant images (red framed) in later ranks than the others.

Database Side Augmentation: We employ a Database-side Augmentation (DBA) approach that is similar to that of [14], which entails dividing the weights by the temperature parameter T that regularizes the softmax inputs. Hence,

[4] We will release our codes and pretrained models for the reproducable baseline (LAttQE) as well as the UGQE at the time of publication.

the softmax function is applied to the regularized logits. After calculating the expanded query with the tempered weights, the resulting vector is l_2 normalized. For a given query, the weights and the expanded query are calculated as follows: $\mathbf{w} = \text{Softmax}\left(\text{normalizeddotproduct}\left(\tilde{\mathbf{q}}, \left[\tilde{\mathbf{d}}_0, \tilde{\mathbf{d}}_1, \ldots, \tilde{\mathbf{d}}_k\right]\right)/T\right)$, and $\hat{\mathbf{q}} = \mathbf{w_q q} + \sum_{\mathbf{n=1}}^{\mathbf{k}} \mathbf{w_{d_k} d_k}$, and finally $\hat{\mathbf{q}} = \dfrac{\hat{\mathbf{q}}}{||\hat{\mathbf{q}}||_2}$.

All database vectors are augmented beforehand *offline* as described above. This strategy gives the best results for database side augmentation. In Table 1, DBA+QE section shows the performance results when the described DBA procedure was applied before all QE models. The DBA-added UGQE method again outperforms its baseline in all scenarios. Again, there are mixed results where for $\mathcal{R}$Paris ($\mathcal{R}$Paris), the DBA-added αQE outperforms the others.

Table 1. Mean average precision (mAP) of $\mathcal{R}$Oxford ($\mathcal{R}$Oxf) and $\mathcal{R}$Paris ($\mathcal{R}$Par) with and without 1 million distractors ($\mathcal{R}$1M). Our uncertainty guided query expansion method outperforms both traditional and learning based expansion methods on most of the settings. (* with our implementation, gray lines with † as reported in [14], and all other scores also taken from [14]).

		$\mathcal{R}$Oxf		$\mathcal{R}$Oxf + $\mathcal{R}$1M		$\mathcal{R}$Par		$\mathcal{R}$Par + $\mathcal{R}$1M		
		M	H	M	H	M	H	M	H	Mean
No QE										
—		67.3	44.3	49.5	25.7	80.6	61.5	57.3	29.8	52.0
QE										
[7]	AQE	72.3	49.0	57.3	30.5	82.7	65.1	62.3	36.5	56.9
[13]	AQEwD	72.0	48.7	56.9	30.0	83.3	65.9	63.0	37.1	57.1
[3]	DQE	72.7	48.8	54.5	26.3	83.7	66.5	64.2	38.0	56.8
[24]	αQE	69.3	44.5	52.5	26.1	**86.9**	**71.7**	**66.5**	**41.6**	57.4
[14]	LAttQE†	73.4	49.6	58.3	31.0	86.3	70.6	67.3	42.4	59.8
[14]	LAttQE*	73.2	49.7	57.1	30.0	84.3	67.2	63.9	37.8	57.9
	UGQE	**73.3**	**50.1**	**58.3**	**31.0**	86.2	70.8	65.0	39.3	**59.2**
DBA + QE										
[7]	ADBA + AQE	71.9	53.6	55.3	32.8	83.9	68.0	65.0	39.6	58.8
[13]	ADBAwD + AQEwD	73.2	53.2	57.9	34.0	84.3	68.7	65.6	40.8	59.7
[3]	DDBA + DQE	72.0	50.7	56.9	32.9	83.2	66.7	65.4	39.1	58.4
[24]	αDBA + αQE	71.7	50.7	56.0	31.5	**87.5**	**73.5**	**70.6**	**48.5**	61.3
[14]	LAttQE + LAttDBA†	74.0	54.1	60.0	36.3	87.8	74.1	70.5	48.3	63.1
	LAttQE + LAttDBA*	73.8	54.4	57.8	33.0	85.8	70.6	67.2	42.7	60.7
	UGQE + UGQEDBA	**75.5**	**56.3**	**58.0**	**31.6**	87.3	73.3	67.7	43.7	**61.7**

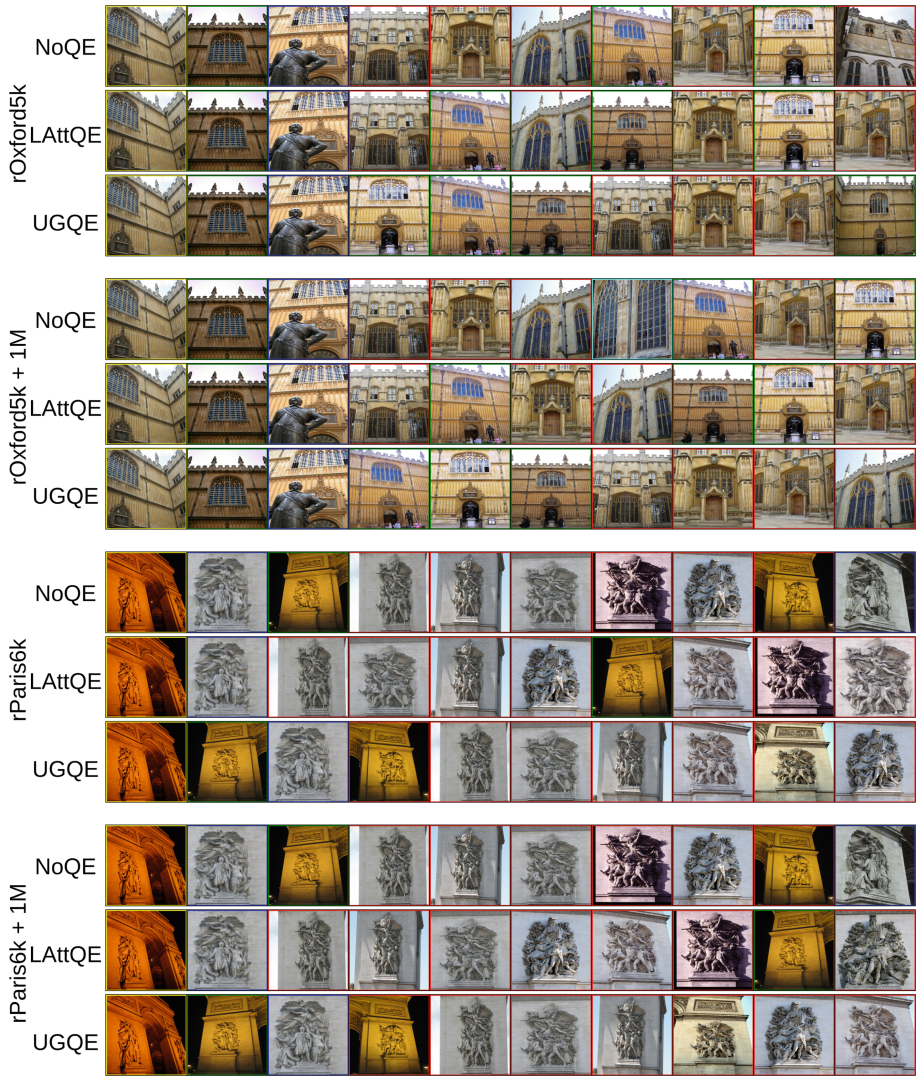

Fig. 2. Sample Visual Results: Color codes are as follows: Yellow Frame: Query Image, Blue or Green Frame: Relevant Image, Red Frame: Irrelevant Image, Cyan Frame: Distractor Image. As can be seen from examples, uncertainty information helps to remove some of the irrelevant retrievals. (Color figure online)

5 Conclusion

In this work, we proposed an uncertainty guided self-attention mechanism to learn query expansion in an end-to-end fashion. We built a novel feature generation method that is compatible with the traditional methods such as the QE, AQEwD and the learning based LAttQE method. Our work provides evidence to

our hypothesis that the integration of uncertainty information on the neighborhood relationships in image retrieval methods can lead to the creation of more robust retrieval systems.

References

1. Amini, A., Schwarting, W., Soleimany, A., Rus, D.: Deep evidential regression. In: Advances in Neural Information Processing Systems, vol. 33 (2020)
2. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016 (2016)
3. Arandjelovic, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: CVPR (2012)
4. Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 584–599. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_38
5. Brown, T., et al.: Language models are few-shot learners. In: Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901 (2020)
6. Buckley, C.: Automatic query expansion using smart: TREC 3. In: Proceedings of the 3rd Text REtrieval Conference (TREC-3), pp. 69–80 (1994)
7. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: automatic query expansion with a generative feature model for object retrieval. In: 2007 IEEE 11th ICCV (2007)
8. Datar, M., Indyk, P.: Locality-sensitive hashing scheme based on p-stable distributions. In: Proceedings of the 20th Annual Symposium on Computational Geometry, SCG 2004, pp. 253–262. ACM Press (2004)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)
10. Dosovitskiy, A., et al.: An image is worth 16 × 16 words: transformers for image recognition at scale. In: 9th International Conference on Learning Representations, ICLR (2021)
11. El-Nouby, A., Neverova, N., Laptev, I., Jégou, H.: Training vision transformers for image retrieval. CoRR abs/2102.05644 (2021)
12. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In: ICML, pp. 1050–1059. PMLR (2016)
13. Gordo, A., Almazán, J., Revaud, J., Larlus, D.: End-to-end learning of deep visual representations for image retrieval. Int. J. Comput. Vis. **124**(2), 237–254 (2017)
14. Gordo, A., Radenovic, F., Berg, T.: Attention-based query expansion learning. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12373, pp. 172–188. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58604-1_11
15. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: International Conference on Machine Learning, pp. 1321–1330 (2017)
16. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3304–3311 (2010)
17. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv:1412.6980 (2014)

18. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
19. Liu, X., Duh, K., Liu, L., Gao, J.: Very deep transformers for neural machine translation. arXiv preprint arXiv:2008.07772 (2020)
20. Lowe, D.: Object recognition from local scale-invariant features. In: Proceedings of the 7th IEEE ICCV (1999)
21. Malinin, A., Gales, M.: Predictive uncertainty estimation via prior networks. arXiv preprint arXiv:1802.10501 (2018)
22. Noh, H., Araujo, A., Sim, J., Weyand, T., Han, B.: Large-scale image retrieval with attentive deep local features. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), October 2017 (2017)
23. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_11
24. Radenovic, F., Tolias, G., Chum, O.: Fine-tuning CNN image retrieval with no human annotation. IEEE Trans. Pattern Anal. Mach. Intell. **41**(7), 1655–1668 (2019)
25. Radenović, F., Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Revisiting Oxford and Paris: large-scale image retrieval benchmarking. In: CVPR (2018)
26. Salton, G., Buckley, C.: Improving retrieval performance by relevance feedback. J. Am. Soc. Inf. Sci. **41**(4), 288–297 (1990)
27. Seidenschwarz, J., Elezi, I., Leal-Taixé, L.: Learning intra-batch connections for deep metric learning. In: 38th International Conference on Machine Learning (ICML) (2021)
28. Sensoy, M., Kaplan, L., Kandemir, M.: Evidential deep learning to quantify classification uncertainty. In: Advances in Neural Information Processing Systems (2018)
29. Tolias, G., Jégou, H.: Visual query expansion with or without geometry: refining local descriptors by feature aggregation. Pattern Recogn. **47**(10), 3466–3476 (2014)
30. Tolias, G., Sicre, R., Jégou, H.: Particular object retrieval with integral max-pooling of CNN activations. In: Bengio, Y., LeCun, Y. (eds.) 4th International Conference on Learning Representations, ICLR (2016)
31. Turcot, T., Lowe, D.G.: Better matching with fewer features: the selection of useful features in large database recognition problems. In: ICCV Workshop (2009)
32. Vaswani, A., et al.: Attention is all you need. In: NeurIPS (2017)
33. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018)