A Large-Scale Image Retrieval System for Everyday Scenes

Arun Zachariah
azachariah@mail.missouri.edu
University of Missouri-Columbia

Mohamed Gharibi
mggvf@mail.umkc.edu
University of Missouri-Kansas City

Praveen Rao
praveen.rao@missouri.edu
University of Missouri-Columbia

ABSTRACT

We present a system for large-scale image retrieval on everyday scenes with common objects. Our system leverages advances in deep learning and natural language processing (NLP) for improved understanding of images by capturing the relationships between the objects within an image. As a result, a user can retrieve highly relevant images and obtain suggestions for similar image queries to further explore the repository. Each image in the repository is processed (using deep learning) to obtain the most probable captions and objects in it. The captions are parsed into tree structures using NLP techniques, and stored and indexed in a database system. When a query image is posed, an optimized tree-pattern query is executed by the database system to obtain candidate matches, which are then ranked using tree-edit distance of the tree structures to output the top-k matches. Word embeddings and Bloom filters are used to obtain similar image queries. By clicking the suggested similar image queries, a user can intuitively explore the repository.

CCS CONCEPTS

• **Information systems** → **Image search**.

KEYWORDS

Image retrieval; deep learning; natural language processing; XML

ACM Reference Format:
Arun Zachariah, Mohamed Gharibi, and Praveen Rao. 2021. A Large-Scale Image Retrieval System for Everyday Scenes. In *ACM Multimedia Asia (MMAsia '20), March 7–9, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3444685.3446253

1 INTRODUCTION

Content-based image retrieval aims to find images in a database that are similar to a query image. Typically, there are two stages during image retrieval, namely, the filtering stage to identify a set of candidate images and a re-ranking stage, where a small number of similar candidates are re-ranked based on specific criteria. Recently, several techniques have explored the use of convolutional neural networks (CNNs) for large-scale image retrieval. These rely on CNN features for global image representations enabling fast filtering [6, 7, 9, 12, 20, 25, 27]. For re-ranking, local image representations from CNNs have been employed through spatial matching and geometric verification [15–17]. A recent technique extracted deep local features from CNNs for indexing and ranked based on geometric verification [17, 24]. Another instance retrieval technique [23] leveraged both local and global features from a Faster R-CNN [21].

Techniques relying on CNN-based features were tested on datasets containing buildings, scenic views, and landmarks along with other distractor images [2, 10, 18, 19]. The evaluated queries contained only buildings, landmarks, etc. We argue that images containing everyday scenes with common objects are quite different from the aforementioned datasets as they contain objects in the foreground and background. In such an image, certain objects become the main focus when a human observes it. We observed that prior techniques [9, 12, 17, 23] failed to precisely capture the main aspect of an image (depicting an everyday scene) leading to false positives [28]. For example, people in a query image were absent in the retrieved images. We remark that human cognition can capture the key essence of an image and describe it aptly via a caption; it can ignore objects (or regions) in an image that do not really matter to describe the main context of the image. Thus, accurate image captioning can aptly describe everyday scenes.

Motivated by the aforementioned reasons, we propose a system called QIK (**Q**uerying **I**mages Using Contextual **K**nowledge) to achieve superior image retrieval performance on everyday scenes with common objects. Rather than constructing local/global descriptors of images using CNN-based features, QIK uses the predictions made by deep networks for image understanding tasks, namely, image captioning and object detection. We refer to these predictions (made on an image) collectively as the *probabilistic image understanding* (PIU) of the image. QIK employs modern NLP techniques for efficient and accurate image retrieval on everyday scenes.

2 ARCHITECTURE OF QIK

The architecture of QIK is illustrated in Figure 1 and contains two major components: the *Indexer* and the *Query Processor*. The Indexer generates the PIU of each image in the repository using state-of-the-art models for image captioning [26] and object detection [22]. A PIU consists of most probable captions and detected objects in an image. This enables us to capture the context of everyday scenes and learn the relationships between objects in them. On each caption, the Indexer constructs a parse tree and a dependency tree [11]. A parse tree captures the syntactic structure of a caption by identifying noun phrases, nouns, verb phrases, verbs, adjectives, determiners, prepositions, etc., using parts-of-speech (POS) tagging. A dependency tree provides a representation of how words in a caption are connected by syntactic dependencies. The collection of these ordered trees are represented as XML (Extensible Markup Language) documents, which are stored and indexed using an XML database system. The other contents of a PIU (i.e., detected objects) are also indexed. The Indexer maintains Bloom filters for POS tags such as VBG, NN, JJ, and others to quickly test which words appear under these tags for all the image captions in the repository. It also

MMAsia '20, March 7–9, 2021, Virtual Event, Singapore

ACM ISBN 978-1-4503-8308-0/21/03.
https://doi.org/10.1145/3444685.3446253

constructs word embeddings by training on the image captions using word2vec [14]. These are required for fast generation of similar image queries, which will be discussed later. By design, the Indexer can index new images in real-time.

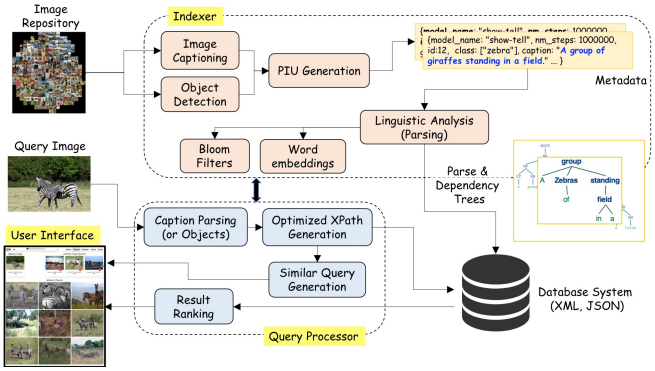

Figure 1: QIK's architecture

The Query Processor can either use image captions or detected objects for image retrieval. When using captions for retrieval, the filtering step begins with the generation of the most probable caption of a query image and the associated parse and dependency trees. The parse tree is processed to generate an optimized XPath query [8] containing only essential keywords in the caption while preserving the ordering between these keywords and their relationships. Essentially, the Query Processor ignores prepositions, determiners, conjunctions, etc., in the caption. After executing the optimized XPath query on the XML documents, the candidate images are fetched. For the ranking step, the Query Processor relies on the tree edit distance between the parse tree (or dependency tree) of a candidate image's caption and the parse tree (or dependency tree) of the query image's caption. The candidate images are ranked in increasing order of the computed tree edit distance, and the top-k matches are returned to the user. When using objects for retrieval, the filtering step begins with the detection of objects in an image using an object detection model with probability greater than a user-specified threshold. The set of candidate images that contain all of the detected objects are retrieved. For ranking, the probabilities of the detected objects in the query image and in a candidate image are combined to compute a score for the candidate image. The candidate images are ranked in decreasing order of the score, and the top-k matches are returned to the user.

To suggest similar image queries for a query image, the Query Processor does the following: Using the optimized XPath query, it generates different XPath queries by replacing the XPath text nodes with similar words. To obtain words similar to a given word, its nearest neighbors are computed using word embeddings constructed on image captions. This can lead to an exponential number of possibilities. Further, to ensure that by replacing a word with a similar word will yield an XPath query that produces a non-empty result, Bloom filters are checked based on the XML element name enclosing the word. Only new XPath queries that produce non-empty results are executed. Only one candidate image for a similar query is shown. (This process requires only the filtering step but not the ranking step.) For instance, if a user posed a query depicting "*a group of zebras standing on a field*," a similar image query depicting "*a group of giraffes standing on a field*" is suggested to the user. Thus, the user can click on these queries to explore the repository.

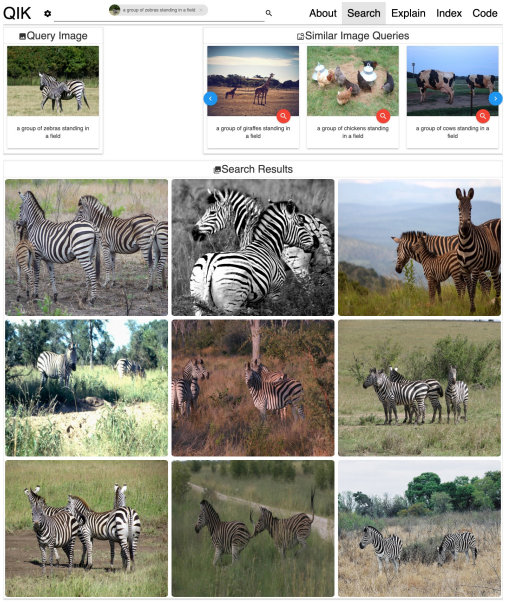

Figure 2: A screenshot of QIK

3 IMPLEMENTATION

QIK was developed in Java and compiled using Java 1.8. It uses Show and Tell [26] for generating image captions and the Stanford Parser package (version 3.9.2) for generating the parse/dependency trees of the captions. BaseX [1] is used to store and index the XML data generated from images and executing XPath queries. The user interface for querying and navigating over the results is built using Django [3]. A screenshot of QIK is shown in Figure 2. The QIK software and errata are available on GitHub [4].

4 DEMONSTRATION SCENARIOS

A user can interact with QIK using a web browser. Two datasets will be used: MS COCO [13] (with 123K images) and everyday scenes from Unsplash [5], a website hosting free high-resolution images. (The indexes on these datasets will be built ahead of time.) The user can select a query image. QIK will output the most relevant top-k matches ranked based on tree-edit distance. It will also return a set of similar image queries w.r.t. the context of the query image. The user can select the retrieval approach (i.e., using captions or detected objects), the ranking scheme (i.e., parse tree vs. dependency tree), and the value k for top-k matches. The user can click and execute any of the similar image queries. The relevant matches along with similar image queries for the executed query will be displayed. This way, the user can intuitively explore a large-scale image repository. The user can update the repository by adding a new image and indexing its PIU. Finally, the user can examine the execution plan of a query and observe how similar image queries were generated for the query image.

Acknowledgments: This work was supported by the National Science Foundation under Grant No. 1747751.

REFERENCES

[1] 2019. BaseX: A robust, high-performance XML database engine. http://basex.org/.

[2] 2019. Google Landmarks Dataset v2. https://github.com/cvdfoundation/google-landmark.

[3] 2020. Django. https://www.djangoproject.com/.

[4] 2020. QIK. https://github.com/MU-Data-Science/QIK.

[5] 2020. Unsplash. https://unsplash.com.

[6] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[7] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. 2014. Neural Codes for Image Retrieval. In *Computer Vision – ECCV 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 584–599.

[8] Anders Berglund, Scott Boag, Don Chamberlin, Mary F. Fernandez, Michael Kay, Jonathan Robie, and Jerome Simeon. 2002. *XML Path Language (XPath) 2.0 W3C Working Draft 16.* Technical Report WD-xpath20-20020816. World Wide Web Consortium. http://www.w3.org/TR/xpath20/

[9] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. 2016. Deep Image Retrieval: Learning Global Representations for Image Search. In *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Springer International Publishing, Cham, 241–257.

[10] Herve Jegou, Matthijs Douze, and Cordelia Schmid. 2008. Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. In *Proceedings of the 10th European Conference on Computer Vision: Part I* (Marseille, France) *(ECCV '08)*. 304–317.

[11] Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (2nd ed.). Prentice Hall, USA.

[12] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. 2016. Cross-Dimensional Weighting for Aggregated Deep Convolutional Features. In *Computer Vision – ECCV 2016 Workshops*, Gang Hua and Hervé Jégou (Eds.). 685–701.

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*. Springer International Publishing, Cham, 740–755.

[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems* (Lake Tahoe, Nevada). 3111–3119.

[15] Anastasiya Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiři Matas. 2017. Working Hard to Know Your Neighbor's Margins: Local Descriptor Learning Loss. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, CA) *(NIPS'17)*. 4829–4840.

[16] Dmytro Mishkin, Filip Radenović, and Jiři Matas. 2018. Repeatability Is Not Enough: Learning Affine Regions via Discriminability. In *Computer Vision – ECCV 2018*. Springer International Publishing, Cham, 287–304.

[17] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. 2017. Large-Scale Image Retrieval with Attentive Deep Local Features. In *Proc. of 2017 IEEE International Conference on Computer Vision (ICCV)*. 1–10.

[18] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2007. Object retrieval with large vocabularies and fast spatial matching.. In *Proc. of CVPR 2007*.

[19] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. 2008. Lost in quantization: Improving particular object retrieval in large scale image databases. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*. 1–8.

[20] F. Radenovic, G. Tolias, and O. Chum. 2019. Fine-Tuning CNN Image Retrieval with No Human Annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 7 (July 2019), 1655–1668.

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1* (Montreal, Canada) *(NIPS'15)*. 91–99.

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 6 (June 2017), 1137–1149.

[23] A. Salvador, X. Giro-i-Nieto, F. Marques, and S. Satoh. 2016. Faster R-CNN Features for Instance Search. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 394–401.

[24] Marvin Teichmann, Andre Araujo, Menglong Zhu, and Jack Sim. 2019. Detect-To-Retrieve: Efficient Regional Aggregation for Image Search. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[25] Giorgos Tolias, Ronan Sicre, and Herve Jegou. 2016. Particular object retrieval with integral max-pooling of CNN activations. In *International Conference on Learning Representations*. 1–12.

[26] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2017. Show and Tell: Lessons Learned from the 2015 MS COCO Image Captioning Challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 4 (April 2017), 652–663.

[27] A. B. Yandex and V. Lempitsky. 2015. Aggregating Local Deep Features for Image Retrieval. In *2015 IEEE International Conference on Computer Vision (ICCV)*. 1269–1277.

[28] Arun Zachariah, Mohamed Gharibi, and Praveen Rao. 2020. QIK: A System for Large-Scale Image Retrieval on Everyday Scenes With Common Objects. In *Proceedings of the 2020 ACM International Conference on Multimedia Retrieval (ICMR)*. Dublin, Ireland, 126–135.