J-EDA: A workbench for tuning similarity and diversity search parameters in content-based image retrieval

João V. O. Novaes[1], Lúcio F. D. Santos[2], Luiz Olmes Carvalho[3], Daniel de Oliveira[4], Marcos V. N. Bedo[5], Agma J. M. Traina[1], and Caetano Traina Jr.[1]

[1] Institute of Mathematics and Computer Science – University of São Paulo (ICMC/USP) São Carlos/SP – Brazil
novaes.jvo@usp.br, {agma,caetano}@icmc.usp.br

[2] Federal Institute of Technology of North of Minas Gerais (IFNMG) Montes Claros/MG – Brazil
lucio.santos@ifnmg.edu.br

[3] Institute of Computer and Mathematics – Federal University of Itajubá (UNIFEI) Itajubá/MG – Brazil
olmes@unifei.edu.br

[4] Institute of Computing – Fluminense Federal University (IC/UFF) Niterói/RJ – Brazil
danielcmo@ic.uff.br

[5] Fluminense Northwest Institute – Fluminense Federal University (INFES/UFF) St. A. Pádua/RJ – Brazil
marcosbedo@id.uff.br

Abstract. Similarity searches can be modeled by means of *distances* following the Metric Spaces Theory and constitute a fast and explainable query mechanism behind content-based image retrieval (CBIR) tasks. However, classical distance-based queries, *e.g.*, Range and k-Nearest Neighbors, may be unsuitable for exploring large datasets because the retrieved elements are often similar among themselves. Although similarity searching is enriched with the imposition of rules to foster result diversification, the fine-tuning of the diversity query is still an open issue, which is is usually carried out with and a non-optimal expensive computational inspection. This paper introduces J-EDA, a practical workbench implemented in Java that supports the tuning of similarity and diversity search parameters by enabling the automatic and parallel exploration of multiple search settings regarding a user-posed content-based image retrieval task. J-EDA implements a wide variety of classical and diversity-driven search queries, as well as many CBIR settings such as feature extractors for images, distance functions, and relevance feedback techniques. Accordingly, users can define multiple query settings and inspect their performances for spotting the most suitable parameterization for a content-based image retrieval problem at hand. The workbench reports the experimental performances with several internal and external evaluation metrics such as $P \times R$ and Mean Average Precision (mAP), which are calculated towards either incremental or batch procedures performed with or without human interaction.

Categories and Subject Descriptors: H.2 [**Database Management**]: Miscellaneous; H.3 [**Information Storage and Retrieval**]: Miscellaneous

Keywords: Content-based image retrieval, Result diversification, Similarity searching

1. INTRODUCTION

Content-Based Image Retrieval (CBIR) systems are first-class tools for recovering images that are most similar to a given image reference (or *query center*) regarding their visual content. They rely on a distance function to assess the similarity of the visual features extracted from the reference image and those stored in a database. Figure 1 describes the typical CBIR process execution that starts with a *description phase* representing every image as a feature vector, which summarizes visual content, *e.g.*, color, texture, or shape. Next, the CBIR search process compares the feature vectors by using a distance function and a query operator – See Figure 1 *retrieval phase*. Distance functions that comply

This research was supported by CAPES [grant 001], CNPq, FAPEMIG, FAPERJ and FAPESP [grants 2020/07200-9 2 and 2016/17078-0].

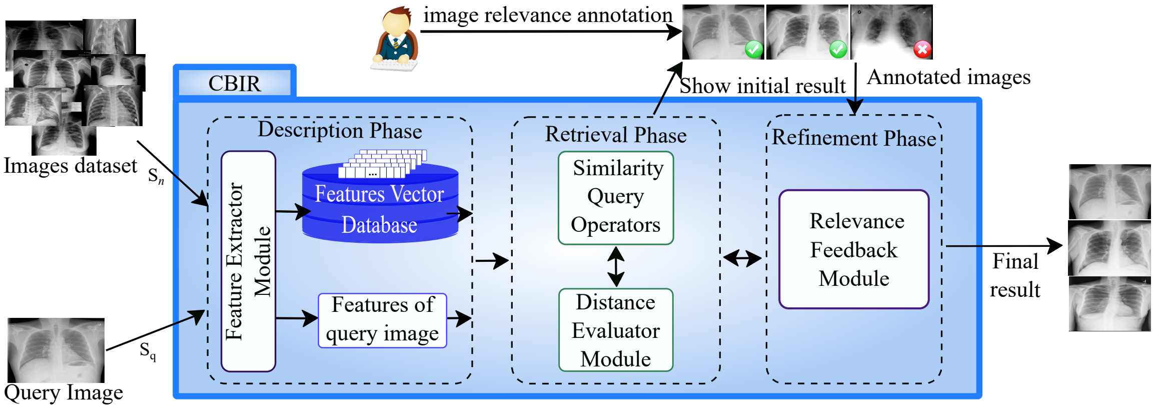


Fig. 1. A pipeline execution of a CBIR task.

with the Metric Space Theory axioms are called *metrics*. The combination of metrics and indexes enables the CBIR process to benefit from Metric Access Methods, which are computationally efficient routines that speeds up the search execution [Traina et al. 2019]. Finally, query operators are used for filtering and dividing the stored images according to their distances to the search reference.

The most employed operators are those of Range (Rng) and k-Nearest Neighbor (k-NN) queries. The former retrieves the images whose distances to the query reference are not larger than a maximum threshold, and the latter recovers the k closest images to the query reference [Hetland 2009]. CBIR also includes an optional *refinement phase* (Figure 1), in which the user can adjust the conditions and re-execute the query through relevance feedback techniques. Such refinement shifts the query perspective (or the reference image) following the user-provided annotations over the result set.

CBIR systems are commonly designed and implemented for specific domains, such as Medical [Bedo and et. al 2016; Santos et al. 2018], agricultural [Ruiz et al. 2018], remote sensing [Rosu et al. 2017], and social networks [Maigrot et al. 2017]. Such lack of a general solution is mainly due to the variety of parameters involved in the search process, *e.g.*, feature vector extractor, distance function, query operator, and relevance feedback technique, which can impact negatively in the quality of CBIR results [Bedo and et. al 2016]. Additionally, another threat to CBIR implementation is the volume and complexity of stored images since both of those factors affect the semantics and the computational efficiency of distance-based query operators [Santos and et. al 2014; Pestov 2013].

A major semantic problem related to data volume is that the query operators are unable to filter result images similar to each other [Drosou et al. 2017]. It leads to the so-called "CBIR gap" where result sets include nearly-duplicate data, which prevents the exploration of relevant portions of the search space and skew any decision-making process. The bridging of such gap is carried out with user-oriented relevance feedback cycles [Xioufis et al. 2015; Santos et al. 2018] or result diversification [Santos et al. 2018]. While relevance feedback cycles are a burdensome task that may drive the user to give up the search [Santos and et. al 2014], result diversification provides new query operators that avoid recovering images similar among themselves [Jasbick et al. 2020].

Therefore, the fine-tuning of CBIR systems requires a suitable choice for the *search parameters*, which depends on the data distribution. Several workbench prototypes were proposed to adjusted those parameters, such as (1) *Image-Hunter* [Tronci and et. al 2013], (2) *DivDB* [Vieira and et. al 2011a], (3) *Poikilo* [Drosou and Pitoura 2013], (4) *VikS* [Santos and et. al 2014], and (5) *Kundaha* [Santos et al. 2018]. Those tools target the parameter tuning problem for a domain, as follows:

—Feature extractor, which determines how images are represented {(4), and (5)};

—Distance function, which calculates the objects' dissimilarity {(2), (3), (4), and (5)};

—Query operators, which filters objects by a distance-based criterion {(2), (4), and (5)};
—Result diversification, which provides diversity to result sets {(2), (3), and (4)};
—Relevance feedback, which provides query refinement {(1), (4), and (5)}.

However, to the best of the authors' knowledge, the existing workbench prototypes for parameter tuning are focused on examining only a subset of parameters and lack an evaluation of the simultaneous combination for the five search parameters listed above. This manuscript addresses this open issue by extending the previous study entitled "*J-EDA: A diversified similarity workbench for content-based image retrieval*" [Novaes et al. 2019] through a practical workbench prototype for CBIR fine-tuning. Our prototype, coined `J-EDA`, enables the interactively setting of features extractors, distance functions, query criteria, result diversification, and relevance feedback in exploration sessions with and without an experimental interface. The parameter tuning process can be carried out with unsupervised and supervised sets and include one or more datasets in an evaluation. The results are returned in an analytics-driven interface with easy-to-follow charts and tables that include average elapsed query time, number of relevance feedback cycles, and quality metrics as Precision and Recall, Mean Average Precision, and Accuracy.

The remainder of this manuscript is organized as follows. Section 2 discusses background and preliminaries. Section 3 introduces the `J-EDA` workbench, a practical workbench for the analysis of diversified similarity queries regarding content-based image retrieval tasks, while Section 4 reports a case study performing diversified similarity queries over real image dataset of Covid-19 diagnosis. Section 5 provides a discussion on related work, and Section 6 concludes the paper.

2. PRELIMINARIES

When analyzing large image databases, Content-Based Image Retrieval (CBIR) systems provide tools for analysts to retrieve small data subsets according to conditions posed on visual features previously extracted from the image visual contents [Marques and Rangayyan 2013; Traina et al. 2010]. In general, data retrieval requires comparing the elements in the database by using comparison operators (θ), and specific θ-operators are applied for each data domain. For instance, θ-operators based on Identity Relation (IR: $\theta \in \{=, \neq\}$) or Order Relation (OR: $\theta \in \{<, \leq, \geq, >\}$) are universally applicable for the number and the string domains, once it is always possible to state about the equality of two elements or to establish precedence among them. However, in the CBIR context, checking if two images are byte-to-byte identical is useless, as well as trying to put several images in a specific order, and therefore IR/OR operators are not suitable for such queries. CBIR systems employ Similarity Relation (SR) [Silva et al. 2013; Traina et al. 2019], which provides a set of tailored operators to execute the retrieving task. The main concepts involving similarity searching are described following.

2.1 Similarity searching

The mathematical model that supports Similarity Search is the *Metric Space*. Formally, a Metric Space is a pair $\langle \mathbb{O}, \delta \rangle$, where $\mathbb{O}$ is a data domain and δ is a metric that quantifies how dissimilar two elements are [Zezula et al. 2010]. Also, for any objects $o_h, o_i, o_j \in \mathbb{O}$, the following properties hold:

(1) *Symmetry*: $\delta(o_i, o_j) = \delta(o_j, o_i)$,
(2) *Non-negativity*: $\delta(o_i, o_j) \geq 0$,
(3) *Identity of indiscernibles*: $\delta(o_i, o_j) = 0 \Leftrightarrow o_i = o_j$, and
(4) *Triangle inequality*: $\delta(o_i, o_j) \leq \delta(o_i, o_h) + \delta(o_h, o_j)$

One of the most employed metric for n-dimensional data, i.*e*., $\mathbb{O} \subseteq \mathbb{R}^n$, is the Minkowski family: L_p. Given two objects $o_i, o_j \in \mathbb{R}^n$ and a value $p \in \mathbb{N}^*$, the L_p function measures the dissimilarity value between o_i and o_j according to Eq. (1).

$$L_p\left(o_i, o_j\right) = L_p\left(\left[o_i^{(1)}, o_i^{(2)}, \ldots, o_i^{(n)}\right], \left[o_j^{(1)}, o_j^{(2)}, \ldots, o_j^{(n)}\right]\right) = \sqrt[p]{\sum_{t=1}^{n}\left|o_i^{(t)} - o_j^{(t)}\right|^p} \quad (1)$$

Considering a particular dataset $\mathcal{O} \subseteq \mathbb{O}$ and a query object $o_q \in \mathbb{O}$, two main queries can be performed to retrieve the most similar objects $o_i \in \mathcal{O}$ to o_q: the similarity Range query and the k-Nearest Neighbor query, expressed in Definitions 2.1 and 2.2, respectively.

Definition 2.1 *Range query - Rq.* Given a maximum threshold $\xi \in \mathbb{R}_+$, a Range query retrieves the elements in $\mathcal{O}$ that are at most the distance ξ from o_q. Formally, the result set of a Range query is: $\mathcal{R}' = Rq(o_q, \xi, \mathcal{O}, \delta) = \{o_i \in \mathcal{O} \mid \delta(o_q, o_i) \leq \xi\}$.

Definition 2.2 *k-Nearest Neighbor query - kNN.* Given an amount value $k \in \mathbb{N}^*$, a k-Nearest Neighbor query (or neighborhood query) retrieves the k most similar elements in $\mathcal{O}$ to o_q. Formally, the result set of a k-Nearest Neighbor query $kNN(o_q, k, \mathcal{O}, \delta)$ is $\mathcal{R}'_k$, such that:

$$kNN(o_q, k, \mathcal{O}, \delta) = \begin{cases} \mathcal{R}'_0 = \varnothing \\ \mathcal{R}'_k = \left\{o_i \in \mathcal{O} \mid \forall o_j \in \left\{\mathcal{O} \setminus \mathcal{R}'_{k-1}\right\} : \delta(o_q, o_i) \leq \delta(o_q, o_j)\right\} \end{cases}$$

Both similarity operators are suitable for searching data in Metric Spaces considered as sparse. Nowadays, considering the quick growth of the data volume mostly due to the popularity of social networks and mobile devices, more and more data such as images are gathered and stored in large repositories. The main problem with the similarity model is that, as the Metric Space is filled with data, its regions become denser, and elements within those regions become too similar. Therefore, both similarity operators may struggle in the search. For instance, when computing a Range query in a denser Metric Space, even a slight variation of the parameter ξ drastically increases the cardinality of the result set [Pestov 2013]. Also, the kNN query may become unstable because its result set tends to include elements too similar to each other, which does not bring valuable information for the CBIR context. Aiming at overcoming those problems and without losing the practical meaning and behavior of the similarity search, a *diversity degree* can be added into similarity queries [Drosou et al. 2017; Santos et al. 2018; Jasbick et al. 2020], as next explained.

2.2 Diversified similarity searching

The inclusion of a diversity factor in similarity queries has been mainly explored for kNN queries. In general, it can be seen as an optimization problem to retrieve a result set $\mathcal{R}$ of k objects in a similarity query, such that the retrieved elements are relevant to the similarity query *and* also diverse among them. A diversity metric div is employed as the objective function. Formally, the Result Diversification Problem is stated in Definition 2.3.

Definition 2.3 *Result Diversification Problem.* Let $\mathcal{R}'$ be the result set of a similarity query on a dataset $\mathcal{O} \subseteq \mathbb{O}$, $k = \{k \in \mathbb{N}^* \mid k \leq |\mathcal{O}|\}$ be an amount value, and div be a diversity metric. A Diversified Result Set $\mathcal{R}$ is a set of k elements such that $\mathcal{R} = \arg\max_{\mathcal{R} \subseteq \mathcal{R}', |\mathcal{R}|=k} div(\mathcal{R}')$.

The main approaches to solve the Result Diversification Problem (RDP) are classified into three taxonomy groups: distance-based, novelty-based, and coverage-based methods [Drosou et al. 2017; Zheng et al. 2017].

Distance-based methods are based on the *argmax* function for the optimization problem. For instance, methods *MaxMin* and *MaxSum* aim at maximizing *argmax* as the minimal and the maximal sum of *div* values, respectively. Those methods fit poorly on similarity queries because they process the entire dataset $\mathcal{O}$ regardless of the query object o_q [Zheng et al. 2017]. Therefore, the result set is the same for any query object, which expressively reduces the search semantics.

Novelty-based methods consider the RDP as a bi-criteria problem where *similarity* and *diversity* compete against each other ruled by a linear parameter $\lambda \in [0, 1]$. Setting $\lambda = 0$ turns *diversity* irrelevant, and the RDP becomes a kNN query. Increasing the λ value retrieves elements more distant from the query object. Finding optimal values to λ is a NP-hard problem [Drosou et al. 2017], which leads to solutions that employ (meta-)heuristics to produce suitable results [Vieira and et. al 2011b]. Examples of that group includes the Maximum Marginal Relevance (MMR) [Carbonell and Goldstein 1998], Greedy with Marginal Contribution (GMC) [Vieira and et. al 2011b] and the Swap [Yu et al. 2009] methods.

Given a query object $o_q \in \mathbb{O}$, the MMR assigns a score for each element $o_i \in \mathcal{O} \subseteq \mathbb{O}$ according to the objective function $\arg\max = MMR(o_i, o_q) = (1-\lambda)\cdot\delta_{sim}(o_i, o_q) + 2\cdot\lambda\cdot\sum_{o_j\in\mathcal{R}'}\delta_{div}(o_i, o_j)$, where *similarity* and *diversity* are computed using the metrics δ_{sim} and δ_{div}, respectively. The diversified result set $\mathcal{R}$ is greedy and incrementally constructed in k steps where each element $o_i \in \{\mathcal{O} \setminus \mathcal{R}\}$ with the highest $MMR(o_i, o_q)$ value is selected. In a similar way, the GMC method employs the same objective function of MMR, but adds to it the weighted contribution of elements o_h that are outside the result set by using the diversity metric δ_{div}. Such a term is $2\cdot\lambda\cdot\sum_{o_h\in\mathcal{R}'}^{\mathcal{R}'\subseteq\mathcal{O}\setminus\mathcal{R},|\mathcal{R}'|=h-|\mathcal{R}|}\delta_{div}(o_i, o_h)$. The Swap method executes element permutations in order to avoid a local maxima result set. From a baseline result set $\mathcal{R}'$ produced by a kNN query, Swap sorts the elements $o_i \in \{\mathcal{O}\setminus\mathcal{R}'\}$ by distance δ_{sim} and swaps them with the $\mathcal{R}'$ objects according to some objective function such as those aforestated.

Coverage-based methods create a distance-based separation among the elements in the Metric Space. Those methods consider the objects that are set apart by a threshold as diverse, following predefined criteria. Examples of that group are the Motley [Jain et al. 2004] and the Better Results with Influence Diversification (BRID) [Santos et al. 2013] techniques.

The Motley method employs a user-defined value $r \in \mathbb{R}_+$ to define a separation threshold among the elements in the result set. For a given query object $o_q \in \mathbb{O}$, Motley sorts the elements $o_i \in \mathcal{O}$ by their distance to o_q and includes the 1-NN into the partial result set $\mathcal{R}$. The next elements o_i are included in $\mathcal{R}$ only if their distance to any element in $\mathcal{R}$ is at least r. The procedure continues until obtaining $|\mathcal{R}| = k$ or the candidate list turns empty.

Applying the properties of the Metric Space model, the BRID technique creates dynamic separation thresholds and eliminates the requirement for an extra user-provided parameter. The idea in BRID is to balance *similarity* and *diversity* through the *Influence concept*. Given two objects $o_i, o_j \in \mathcal{O}$, the Influence function $I(o_i, o_j)$ measures the mutual impact of the distance between o_i and o_j according to Eq. (2). Using the Influence function, BRID creates a *ternary* relationship in a tuple of three elements $o_h, o_i, o_j \in \mathbb{O}$, where o_h is said to be *more influenced* by o_i than by o_j if $I(o_h, o_i) \geq I(o_h, o_j)$. Then, for an arbitrary $o_q \in \mathbb{O}$ and an object $o_i \in \mathcal{O} \subseteq \mathbb{O}$, BRID defines a region of exclusion denoted *strong influence set*: $\ddot{I}_{o_q,o_i}$. The strong influence set leverages the distance between the pair $\langle o_q, o_i\rangle$ and delimits its influence coverage set according to Eq. (3).

$$I(o_i, o_j) = 1/\delta(o_i, o_j) \tag{2}$$

$$\ddot{I}_{o_q,o_i} = \{o_j \in \mathcal{O} \mid (I(o_i, o_j) \geq I(o_i, o_q)) \wedge (I(o_j, o_i) \geq I(o_j, o_q))\} \tag{3}$$

By using the strong influence set, BRID can bind a query object o_q, a dataset $\mathcal{O}$, and a result set $\mathcal{R}$, to retrieve the k closest elements in $\mathcal{O}$ to o_q that also lie outside the strong influence sets of the objects in $\mathcal{R}$. As a result, combining BRID with the neighborhood queries leads to the Diversified Neighborhood Queries, expressed in Definition 2.4.

Definition 2.4 *Diversified Neighborhood Queries.* A Diversified Neighborhood Query retrieves $k \in \mathbb{N}^*$ non-influenced elements $o_h \in \mathcal{O}$ whose distances to the query center o_q are smallest. The result set is $\mathcal{R} = \Big\{ o_h \in \mathcal{O} \mid \Big(\forall o_j \in \mathcal{R} : o_h \notin \ddot{I}_{o_j,o_q} \Big) \wedge \Big(\forall o_i \in \{\mathcal{O} \setminus \mathcal{R}\} : \delta(o_h, o_q) \leq \delta(o_i, o_q) \vee \exists o_j \in \mathcal{R} : o_i \in \ddot{I}_{o_j,o_q} \Big) \Big\}$

2.3 Relevance feedback

A common issue related to similarity queries is the semantic gap, which depends on the feature extraction process and their understanding from the human perspective. Relevance Feedback (RF) techniques are employed in CBIR systems to alleviate such a problem. Their goal is to consider the user interaction with the system to adjust the query result to the user expectation. Among the several approaches for RF, the main are: Query Point Movement [Rocchio 1971], Support Vector Machines (SVM) [Tronci and et. al 2013] and Similarity Refinement [Su et al. 2015; Santos et al. 2018].

Query Point Movement is one the seminal RF methods. It aims at repositioning the query object o_q in Metric Space moving it away from the irrelevant elements and bringing it closer to those tagged as relevant. In each iteration, that method updates the query center according to the expression $o'_q = o_q + 1/r \cdot \sum_{t=1}^{i} o_i - 1/s \cdot \sum_{t=1}^{j} o_j$, for the r amount of relevant elements o_i and s non-relevant elements o_j.

Support Vector Machines are interesting to deal with data in high dimensional spaces. Once that method finds a decision boundary in the features space, it can be trained using the feedback given by the user. Thus, for each n dimensional object o_i, its relevance rel is combined into a relevance score given by $1/n \cdot \sum_{t=1}^{n} rel^t_{SVM}(o_i)$.

The Similarity Refinement method aims at assigning a weight for each dimension of the feature space. In that manner, the method changes the way the employed metric represents the distance among the elements by defining some dimensions as more or less important than others. For instance, in Eq. (1), the method includes a factor w_t multiplying each term t of the summation expression.

2.4 Evaluation metrics

Considering the number of combinations of parameters associated with each query, the quality of the answers must be measured in some aspect to evaluate how good or worse a result set is concerning others. The literature is rich in evaluation metrics, where the most employed ones are: Precision vs. Recall ($P \times R$), Mean Average Precision (mAP), and Average Recall (AR) [Baeza-Yates and Ribeiro-Neto 1999; Su et al. 2015].

In $P \times R$, Precision is defined as the number of relevant elements retrieved by the number of relevant elements in the dataset. The recall is defined as the number of relevant elements retrieved by the total number of elements retrieved in the query. In a $P \times R$ plot, the top the curve, the better the result.

The Average Precision is a metric employed to evaluate the precision of the query after each relevant object is retrieved. The Average Precision is given by Eq. (4), where Rel is the number of relevant elements. In an evaluation process, several queries are usually performed. The Mean Average Precision is employed to evaluate the precision of all queries according to Eq. (4), where Q is the set of all performed queries. Lastly, the Average Recall enables to measure the mean of the recall of all performed queries, that is, the number of relevant elements retrieved by the number of relevant elements stored in the database (RS). AR is defined according to Eq. (4).

$$AP = 1/|Rel| \cdot \sum_{t=1}^{|\mathcal{R}|} \text{Precision}(\mathcal{R}_i);\ mAP = 1/|Q| \cdot \sum_{q=1}^{|Q|} AP_q;\ AR = \frac{\sum_{q=1}^{|Q|} |Rel|}{|Q| \cdot |RS|} \tag{4}$$

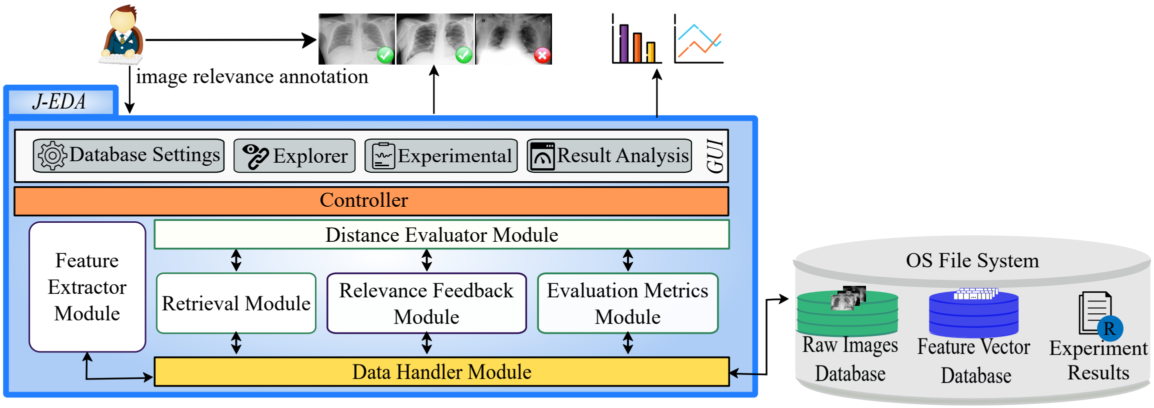


Fig. 2. J-EDA architecture and its components.

3. THE J-EDA WORKBENCH

J-EDA[1] workbench is a JAVA desktop application implemented according to the modular architecture of Figure 2. Roughly speaking, J-EDA consists of eight main components, namely: *(i)* Graphical User Interface (GUI), *(ii)* Controller, *(iii)* Feature Extractor, *(iv)* Distance Evaluator, *(v)* Data Handler, *(vi)* Retrieval Module, *(vii)* Relevance Feedback Module, and *(viii)* Evaluation Metrics Module. J-EDA GUI enables users to set input parameters as well as choose the experimental setup to be evaluated, whereas the Controller translates the user inputs into commands for the implementation modules. The Feature Extractor component provides an implementation of classical low-level algorithms for describing images regarding color, texture, and shape. The Distance Evaluator component implements the following set of weighted distance functions Bray-Curtis, Canberra, L_∞, L_2 and L_1.

In its turn, the Data Handler component provides a common data structure for pairing images and feature vectors for content-based retrieval. J-EDA Retrieval module provides both similarity query criteria k-*NN* and *Rng* that can be combined with the following result set diversification methods: *(i)* Better Result with Influence Diversification (BRID), *(ii)* First Match (FM), *(iii)* Greedy Marginal Contribution (GMC), *(iv)* Maximal Marginal Relevance (MMR), and *(v)* Relative Grouping based on Influence (ReGi). Analogously, the Relevance Feedback module includes the following user-driven query feedback approaches: *(i)* Query Point Movement, *(ii)* Support Vector Machines-based Relevance, and *(iii)* Similarity Refinement. The system's final module is the Evaluation Metrics that quantify the experimental trials and reports quality measures such as $P \times R$ and mAP.

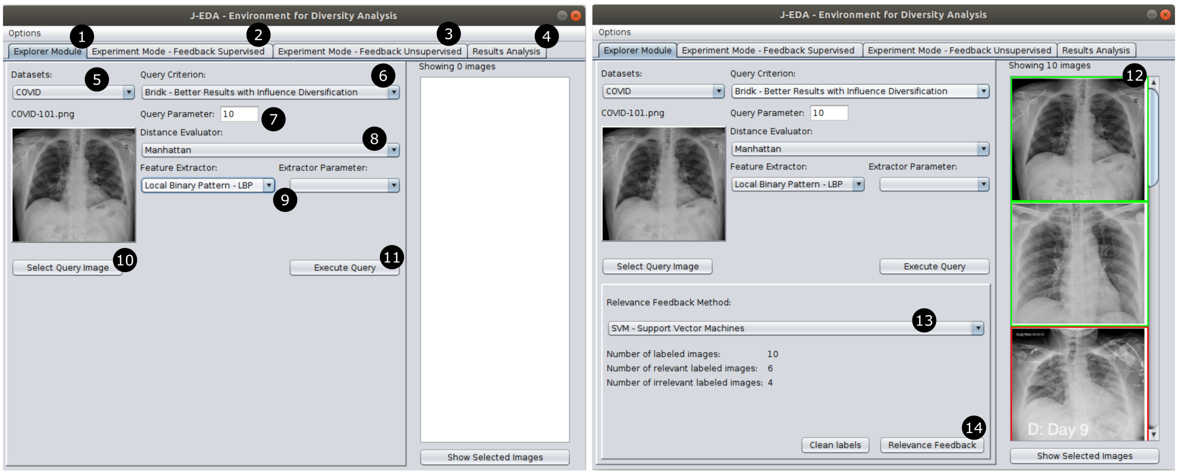
A theoretical background regarding the implemented result diversification and relevance feedback approaches can be found at J-EDA repository[1]. In our implementation, we used the OpenCV4 library[2] behind the Feature Extractor Module, which provides the low-level image descriptors Color Layout, Color Structure, Dominant Color, Edge Histogram, Scalable Color, Haralick, and Local Binary Pattern to J-EDA. Both Retrieval and Relevance Feedback Modules were implemented from scratch using the algorithms provided by [Rocchio 1971; Carbonell and Goldstein 1998; Porkaew and Chakrabarti 1999; Marakakis et al. 2009; Skopal et al. 2009; Vieira and et. al 2011b; Santos et al. 2013; Santos et al. 2014]. Finally, the Evaluation Metrics Module was implemented by using the JFreeCharts library[3], which generates the plots for the experimental evaluations. Accordingly, the Data Handler Module acts as a data bus to provide access to the files shared by J-EDA modules and the OS file system.

[1] Available at `github.com/NovaesJVO/J-EDA`
[2] Available at `opencv.org/opencv-4-0/`
[3] Available at `www.jfree.org/jfreechart/index.html`

4. A CASE STUDY FOR J-EDA WORKBENCH – THE COVID-19-R DATASET

Here, we present a case study for J-EDA and the main steps for identifying the most suitable search parameters to perform content-based image retrieval operations over the labeled dataset Covid-19-R[4] [Rahman et al. 2021; Chowdhury and et. al 2020]. The dataset includes 21,173 images from chest X-rays whose contents are classified into four classes of mutually exclusive labels, namely: *COVID*, *lung Opacity*, *Normal* and *Viral Pneumonia*.

Upon entering J-EDA, users are provided with the main workbench interface – Figure 3. It includes four upper tabs that summarize the types of experimental evaluations available on J-EDA, namely (1) Explorer, which enables the manual evaluation of a single query image over a dataset, (2) Experimental batch mode with supervised feedback, which enables the batch evaluation of image datasets with user interactions, (3) Experimental batch mode with unsupervised feedback, which automatically evaluates query parameters over labeled image datasets, and (4) Result set analysis, which reports the metrics calculated in the experiments from previous tabs.



Datasets: COVID
COVID-101.png
Query Criterion: Bridk - Better Results with Influence Diversification
Query Parameter: 10
Distance Evaluator: Manhattan
Feature Extractor: Local Binary Pattern - LBP
Extractor Parameter:
Showing 0 images

Datasets: COVID
COVID-101.png
Query Criterion: Bridk - Better Results with Influence Diversification
Query Parameter: 10
Distance Evaluator: Manhattan
Feature Extractor: Local Binary Pattern - LBP
Extractor Parameter:
Relevance Feedback Method: SVM - Support Vector Machines
Number of labeled images: 10
Number of relevant labeled images: 6
Number of irrelevant labeled images: 4
Showing 10 images

Fig. 3. J-EDA main interface with Explorer mode.

The input dataset is defined in the J-EDA main interface where users can either *(i)* set J-EDA to extract visual features regarding the original set of images and labels, or *(ii)* provide feature vectors produced by other tools. In the Explorer panel (Figure 3 (1)), users may also define the query settings, such as the query type, *e.g.*, Range and kNN, and its parameters (6), *e.g.*, number k of neighbors (7), the distance function (8) as well as the feature extractor name and its values (9). Finally, users must provide a query reference (10) and run the similarity search (11). The panel displayed on (12) shows the result set for the setup of Figure 3 over the case study of Covid-19-R dataset.

Users may label each result image as relevant or non-relevant by using right and left clicks, respectively. A relevance cycle is performed by using both the dynamic labels and a user-defined feedback method (13–14). Besides fast CBIR prototyping and investigation of particular cases, the Explorer interface enables the first assessment of viable search parameters for inspection in further interactive supervised and unsupervised sessions.

In the Covid-19-R dataset case study, we used 10 reference images labeled and validated by experts to experiment with every possible J-EDA query setting. Table I provides the top-2 query parameters for those 10 manually examined images, which reduce the number of settings to be investigated in

[4] Available at https://www.kaggle.com/tawsifurrahman/covid19-radiography-database/version/4

ID	Query Criterion	Parameter	Distance Function	Feature Extractor	RF Method
1	BRID	k = 10	L_1	LBP	SVM
2	k-NN	k = 10	L_1	LBP	Similarity Refinement

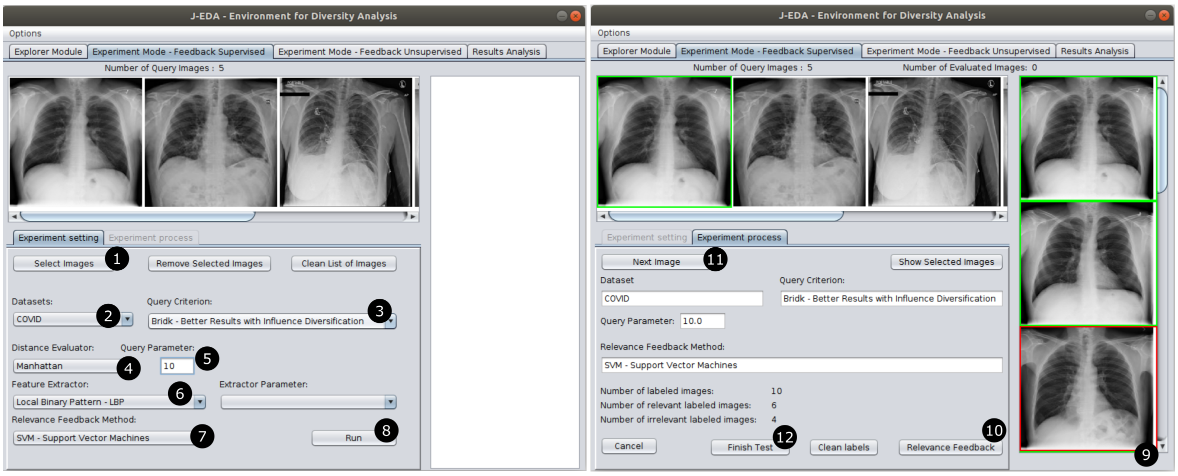
Table I. Delimited search parameters for the `Covid-19-R` case-study after the initial analysis with J-EDA Explorer.

the next tuning stages. Users can inspect such delimited search parameters in detail regarding several query centers and any fraction of the dataset and visualize the results and the evaluation metrics.

Figure 4(2–7) presents the setup for the *holdout* evaluation of the first search parameterization (Table I – ID= 1). In this case, 1% of the images were detached from `Covid-19-R` as a stratified sample, and stored into a separated directory (1) to be queried in batch (8). The holdout evaluation is set after the query parameters (1) and the number of feedback cycles (2) since users must provide the set of images as either *(i)* a random stratified sample of a repository of images (4), *(ii)* a script containing the set of query reference locations (5), or *(iii)* a manual choice of images in the file system (6). Two distinct versions of this Relevance Feedback interface are provided by J-EDA: supervised and unsupervised. In the supervised case, J-EDA shows the retrieved images for every reference into a separated panel, where users inspect the results (9) and label them as relevant (green background) or non-relevant (red background). Feedback cycles were executed upon request and as many as needed by the experts (10). Next, users can either proceed to the next image (11) or interrupt the batch assessment (12). The information obtained during the experiment is saved into a separate log file and is used to calculate the `J-EDA` metrics and describe the experimental evaluation.

In addition to the human-in-the-loop interactions in Figure 4(2–7), which are particularly suitable for *unsupervised* datasets, J-EDA provides a second possibility, which automatically selects relevant images based on the label associated with each result set element – a task applicable only to *supervised* datasets. Figure 5 presents the `Covid-19-R` evaluation with unsupervised feedback. In this case, relevance is inferred upon the classes of the query and result images, *i.e.*, equal labels imply relevance. Besides query parameters (1) and feedback cycles (2), users must provide the set of images as either *(i)* a random sample of an image repository (4), *(ii)* a script containing the query references' locations (5), or *(iii)* manually chosen images (6). As in the supervised case, the experimental log is stored into an external file for further analysis and comparison.

Finally, Figure 6 presents the Results Analysis GUI for the case study and the unsupervised relevance feedback evaluation regarding metrics Accuracy by iteration (Figure 6(a)) and mAP (Figure 6(b)). In



Number of Query Images : 5

Datasets: COVID

Query Criterion: Bridk - Better Results with Influence Diversification

Distance Evaluator: Manhattan

Query Parameter: 10

Feature Extractor: Local Binary Pattern - LBP

Relevance Feedback Method: SVM - Support Vector Machines

Number of Query Images : 5   Number of Evaluated Images: 0

Dataset: COVID

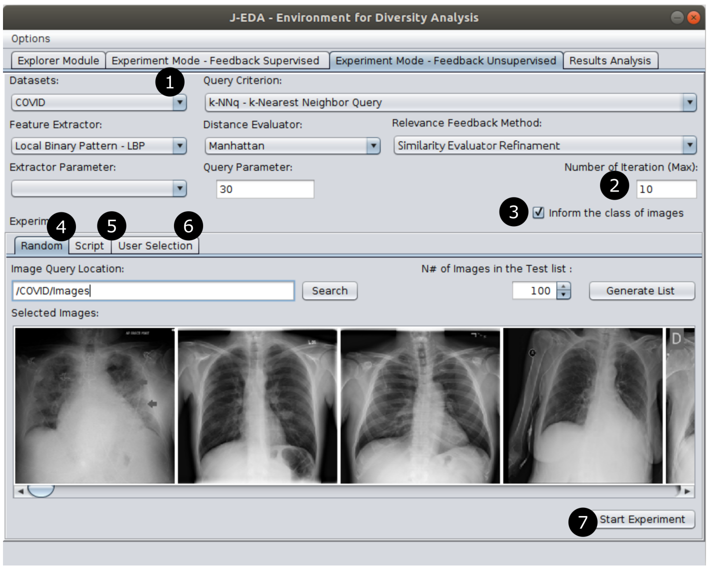
Query Criterion: Bridk - Better Results with Influence Diversification

Query Parameter: 10.0

Relevance Feedback Method: SVM - Support Vector Machines

Number of labeled images: 10
Number of relevant labeled images: 6
Number of irrelevant labeled images: 4

Fig. 4. Relevance feedback support on J-EDA.


J-EDA - Environment for Diversity Analysis
Options
Explorer Module
Experiment Mode - Feedback Supervised
Experiment Mode - Feedback Unsupervised
Results Analysis
1
2
3
4
5
6
7
Random
Script
User Selection
Search
Generate List
Start Experiment

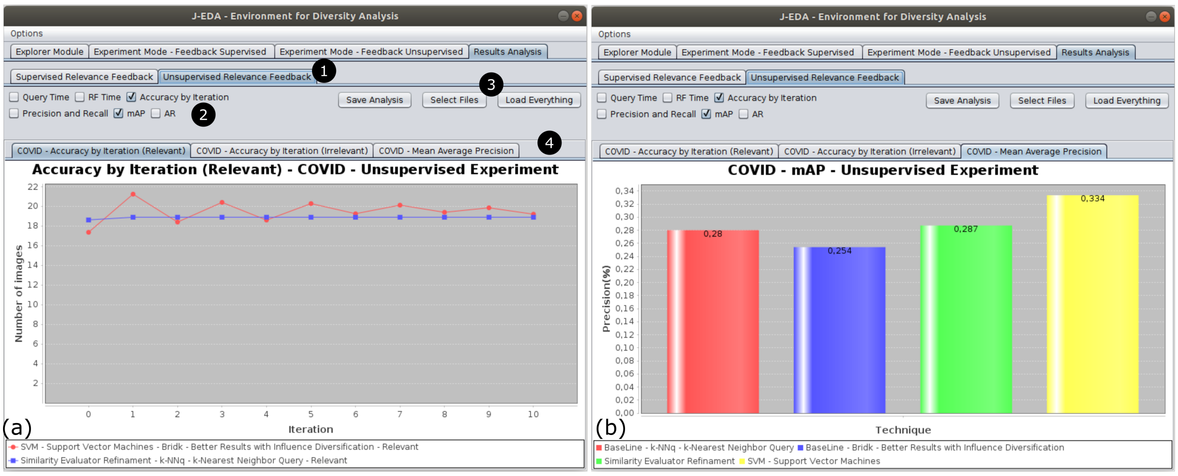
Datasets: COVID
Query Criterion: k-NNq - k-Nearest Neighbor Query
Feature Extractor: Local Binary Pattern - LBP
Distance Evaluator: Manhattan
Relevance Feedback Method: Similarity Evaluator Refinament
Extractor Parameter:
Query Parameter: 30
Number of Iteration (Max): 10
☑ Inform the class of images
Experim
Image Query Location: /COVID/Images
N# of Images in the Test list : 100
Selected Images:

Fig. 5. Experimental mode GUI with unsupervised relevance feedback over labels.

this J-EDA interface, users can choose several supervised and unsupervised metrics (2) as well as plot them in different types to analyze pieces of results (4) recovered from log files (3). The accuracy metrics (Figure 6(a)) regarding the compared query setups (Table I) show that, after one feedback cycle, the combination of BRID diversified similarity algorithm with SVM-based relevance feedback technique achieved a better result in comparison to traditional k-NN with similarity refinement technique after several relevance feedback cycles. Figure 6(b) also shows the result for mAP metric, BRID method and SVM-Based were better ranked, but both relevance feedback techniques have increased the precision of the baseline method.

5. RELATED WORK

Existing parameter-tuning prototypes aim at adjusting individual CBIR rather than examining a combination of the entire search specification. For instance, CBIR *ImageHunter* [Tronci and et. al


J-EDA - Environment for Diversity Analysis
Options
Explorer Module
Experiment Mode - Feedback Supervised
Experiment Mode - Feedback Unsupervised
Results Analysis
Supervised Relevance Feedback
Unsupervised Relevance Feedback
1
2
3
4
Save Analysis
Select Files
Load Everything
COVID - Accuracy by Iteration (Relevant)
COVID - Accuracy by Iteration (Irrelevant)
COVID - Mean Average Precision
Accuracy by Iteration (Relevant) - COVID - Unsupervised Experiment
Number of images
22
20
18
16
14
12
10
8
6
4
2
0
1
2
3
4
5
6
7
8
9
10
Iteration
(a)
SVM - Support Vector Machines - Bridk - Better Results with Influence Diversification - Relevant
Similarity Evaluator Refinament - k-NNq - k-Nearest Neighbor Query - Relevant
COVID - mAP - Unsupervised Experiment
Precision(%)
0,34
0,32
0,30
0,28
0,26
0,24
0,22
0,20
0,18
0,16
0,14
0,12
0,10
0,08
0,06
0,04
0,02
0,00
0,28
0,254
0,287
0,334
Technique
(b)
BaseLine - k-NNq - k-Nearest Neighbor Query
BaseLine - Bridk - Better Results with Influence Diversification
Similarity Evaluator Refinament
SVM - Support Vector Machines

□ Query Time □ RF Time ☑ Accuracy by Iteration
□ Precision and Recall ☑ mAP □ AR

Fig. 6. J-EDA Result Analysis GUI interface for unsupervised feedback.

2013] addresses relevance feedback cycles (k-NN-based and SVM-based) but lacks support for result diversification. Another workbench, *DivDB* [Vieira and et. al 2011a] enables comparing of several novelty-based diversification methods but does not provide a path for assessing how those algorithms fit into relevance feedback cycles. Analogously, *Poikilo* [Drosou and Pitoura 2013] enables the tuning of coverage and novelty-based diversification models, while relevance feedback, distance functions, and feature extractors are left unsupported. The prototype *VikS* [Santos and et. al 2014] provides support for distance functions and result diversification methods while measuring the impact of such parameters depends on user interpretation without relevance feedback cycles. Although *VikS*, `Poikilo` and `DivDB` visualization tools enhances the exploration of the search space, the feature extraction process must be carried out by a third-part system, such as *ImageHunter*.

Finally, related CBIR tools for specific domains are also unable to act as a parameter tuning workbench. For instance, *Kundaha* [Santos et al. 2018] provides a CBIR process for searching medical images through a set of feature extractors, distance functions, metric indexes, relevance feedback methods coupled with medical image visualization functions, *e.g.*, windowing. However, its diversification approach is based solely on the BRID algorithm [Santos et al. 2013], which prevents the use of other result diversification approaches for querying other domains.

6. CONCLUSIONS AND FUTURE WORK

This manuscript described the J-EDA workbench, an easy-to-use and powerful environment for determining suitable settings for querying generic datasets by similarity. Besides the Explorer, J-EDA provides two experimentation modules that benefit from both human-in-the-loop interactions as well as automatic assessments of labeled datasets. Lastly, J-EDA provides an analytic module that allows users to explore metrics and plots regarding distinct query parameters. As future work, we pursue the implementation of J-EDA plugins to support metric indexes and new diversity searching operators. Additionally, plugins to support new relevance feedback techniques are also in consideration to be included in the explorer module.

REFERENCES

Baeza-Yates, R. A. and Ribeiro-Neto, B. A. *Modern Information Retrieval.* ACM Press, Harlow, EN, 1999.

Bedo, M. and et. al. Endowing a content-based medical image retrieval system with perceptual similarity using ensemble strategy. *J. Digital Imaging* 29 (1): 22–37, 2016.

Carbonell, J. and Goldstein, J. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *ACM Conference on Research and Development in Information Retrieval.* ACM, New York, USA, pp. 335–336, 1998.

Chowdhury, M. E. H. and et. al. Can AI Help in Screening Viral and COVID-19 Pneumonia? *IEEE Access* vol. 8, pp. 132665–132676, 2020.

Drosou, M., Jagadish, H., Pitoura, E., and Stoyanovich, J. Diversity in big data: A review. *Big data* 5 (2): 73–84, 2017.

Drosou, M. and Pitoura, E. POIKILO: A tool for evaluating the results of diversification models and algorithms. *PVLDB* 6 (12): 1246–1249, 2013.

Hetland, M. L. The basic principles of metric indexing. In *Swarm intelligence for multi-objective problems in Data Mining.* Springer, New York, USA, pp. 199–232, 2009.

Jain, A., Sarda, P., and Haritsa, J. Providing diversity in k-nearest neighbor query results. In *Advances in Knowledge Discovery and Data Mining.* Springer, 2004.

Jasbick, D., Santos, L., de Oliveira, D., and Bedo, M. Some Branches May Bear Rotten Fruits: Diversity Browsing VP-Trees. In *SISAP.* Springer, pp. 140–154, 2020.

Maigrot, C., Kijak, E., Sicre, R., and Claveau, V. Tampering detection and localization in images from social networks: A CBIR approach. In *ICIAP.* Lecture Notes in Computer Science, vol. 10484. Springer, pp. 750–761, 2017.

Marakakis, A., Galatsanos, N. P., Likas, A., and Stafylopatis, A. Relevance Feedback for Content-Based Image Retrieval Using Support Vector Machines and Feature Selection. In *ICANN.* Springer, pp. 942–951, 2009.

Marques, P. M. d. A. and Rangayyan, R. M. *Content-based Retrieval of Medical Images: Landmarking, Indexing, and Relevance Feedback.* Synthesis Lectures on Biomedical Engineering. Morgan & Claypool Publishers, San Rafael, California, USA, 2013.

Novaes, J. V. O., Bedo, M., Oliveira, D., Traina, A. J. M., Traina Jr, C., and Santos, L. F. D. J-EDA: A diversified similarity workbench for content-based image retrieval. In *SBBD*. SBC, pp. 1–6, 2019.

Pestov, V. Lower bounds on performance of metric tree indexing schemes for exact similarity search in high dimensions. *Algorithmica* 66 (2): 310–328, 2013.

Porkaew, K. and Chakrabarti, K. Query refinement for multimedia similarity retrieval in MARS. In *Int. Conf. on Multimedia*. ACM, pp. 235–238, 1999.

Rahman, T., Khandakar, A., Qiblawey, Y., Tahir, A., Kiranyaz, S., Abul Kashem, S. B., Islam, M. T., Al Maadeed, S., Zughaier, S. M., Khan, M. S., and Chowdhury, M. E. Exploring the effect of image enhancement techniques on covid-19 detection using chest x-ray images. *Comp. in Biology and Medicine* vol. 132, pp. 104319, 2021.

Rocchio, J. J. Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing* 1 (1): 313–323, 1971.

Rosu, R., Donias, M., Bombrun, L., Said, S., Regniers, O., and Da Costa, J. Structure Tensor Riemannian Statistical Models for CBIR and Classification of Remote Sensing Images. *IEEE Trans. on Geoscience and Remote Sensing* 55 (1): 248–260, 2017.

Ruiz, R. R., Rodríguez-Mazahua, L., López Chau, A., Peláez-Camarena, S. G., Abud-Figueroa, M. A., and Machorro-Cano, I. A CBIR system for the recognition of agricultural machinery. *Res. Comput. Sci.* 147 (3): 9–16, 2018.

Santos, L. and et. al. Have you met VikS?: A novel framework for visual diversity search analysis. In *SBBD Demos*. SBC, Curitiba, Brazil, pp. 209–214, 2014.

Santos, L. F. D., Bedo, M. V. N., Ponciano-Silva, M., Traina, A. J. M., and Traina Jr., C. Being similar is not enough: How to bridge usability gap through diversity in medical images. In *CBMS*. IEEE, pp. 287–293, 2014.

Santos, L. F. D., Blanco, G., Oliveira, D. d., Traina, A. J. M., Traina, Caetano, J., and Bedo, M. V. N. Exploring diversified similarity with kundaha. In *CIKM*. ACM, Torino, Italy, pp. 1903–1906, 2018.

Santos, L. F. D., Oliveira, W. D., Ferreira, M. R. P., Traina, A. J. M., and Jr., C. T. Parameter-free and domain-independent similarity search with diversity. In *SSDBM*. ACM, New York, NY, USA, pp. 5:1–5:12, 2013.

Silva, Y. N., Aref, W. G., Larson, P.-A., Pearson, S. S., and Ali, M. H. Similarity queries: their conceptual evaluation, transformations, and processing. *The VLDB J.* 22 (3): 395–420, 2013.

Skopal, T., Dohnal, V., Batko, M., and Zezula, P. Distinct nearest neighbors queries for similarity search in very large multimedia databases. In *WIDM*. ACM, pp. 11–14, 2009.

Su, W., Yuan, Y., and Zhu, M. A Relationship between the Average Precision and the Area Under the ROC Curve. In *Int. Conf. on The Theory of Information Retrieval*. ACM, pp. 349–352, 2015.

Traina, A. J. M., Brinis, S., Pedrosa, G. V., Avalhais, L. P. S., and Traina Jr, C. Querying on large and complex databases by content: Challenges on variety and veracity regarding real applications. *Info. Sys.* vol. 86, pp. 10–27, 2019.

Traina, A. J. M., Traina Jr, C., Balan, A. G. R., Ribeiro, M. X., Bugatti, P. H., Watanabe, C. Y. V., and Marques, P. M. d. A. Feature extraction and selection for decision making over medical images. In *Biomedical Image Processing - Methods and Applications*. Springer, New York, NY, USA, pp. 197–223, 2010.

Tronci, R. and et. al. ImageHunter: A Novel Tool for Relevance Feedback in Content Based Image Retrieval. In *DART*. Springer, Palermo, Italy, pp. 53–70, 2013.

Vieira, M. and et. al. DivDB: A System for Diversifying Query Results. *PVLDB* 4 (12): 1395–1398, 2011a.

Vieira, M. and et. al. On query result diversification. In *ICDE*. IEEE, pp. 1163–1174, 2011b.

Xioufis, E. S., Papadopoulos, S., Gînsca, A.-L., Popescu, A., Kompatsiaris, Y., and Vlahavas, I. P. Improving diversity in image search via supervised relevance scoring. In *ICMR*. ACM, China, pp. 323–330, 2015.

Yu, C., Lakshmanan, L., and Amer-Yahia, S. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *ICDE*. IEEE, ACM, pp. 1299–1302, 2009.

Zezula, P., Amato, G., Dohnal, V., and Batko, M. *Similarity Search: The Metric Space Approach*. Vol. 2. Springer, New York, USA, 2010.

Zheng, K., Wang, H., Qi, Z., Li, J., and Gao, H. A survey of query result diversification. *Know. and Info. Sys.* vol. 51, pp. 1–36, 2017.