A DBMS-Based Framework for Content-Based Retrieval and Analysis of Skin Ulcer Images in Medical Practice

Mirela T. Cazzolato[1], Lucas S. Rodrigues[1], Lucas C. Scabora[1], Guilherme F. Zabot[1], Guilherme Q. Vasconcelos[1], Daniel Y. T. Chino[1], Ana E. S. Jorge[2], Robson L. F. Cordeiro[1], Caetano Traina-Jr.[1], Agma J. M. Traina[1]

[1] Institute of Mathematics and Computer Sciences
University of São Paulo (USP) - São Carlos, Brazil

[2]Department of Physical Therapy
Federal University of São Carlos (UFSCar), São Carlos, Brazil

{mirelac, lucas_rodrigues, lucascsb, zabot, gui.queirozv}@usp.br
{chinodyt, robson, caetano, agma}@icmc.usp.br, anajorge@alumni.usp.br

Abstract. *Bedridden patients with skin lesions (ulcers) often do not have access to specialized clinic equipment. It is important to allow healthcare practitioners to use their smartphones to leverage information regarding the proper treatment to be carried. Existing applications require special equipment, such as heat sensors, or focus only on general information. To fulfill this gap, we propose* `ULEARn`*, a DBMS-based framework for the processing of ulcer images, providing tools to store and retrieve similar images of past cases. The proposed mobile application* `ULEARn-App` *allows healthcare practitioners to send a photo from a patient to* `ULEARn`*, and obtain a timely feedback that allows the improvement of procedures on therapeutic interventions. Experimental results of* `ULEARn` *and* `ULEARn-App` *using a real-world dataset showed that our tool can quickly respond to the required analysis and retrieval tasks, being up to 4.6 times faster than the specialist' expected execution time.*

1. Introduction

Medical facilities, such as hospitals and healthcare centers, generate large amounts of data daily, which is generally stored in a centralized system, allowing physicians and specialists to query for historical data. Exams such as Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) provide different support to the specialist during the diagnosis [García-Zapirain et al. 2018]. However, not every medical practice have access to such specialized equipment, such as home bedridden wounded patients, which may have limited movement and should be treated at home, or wounded patients in remote regions, where a shortage of specialists exist [Gamus et al. 2019]. Taking advantage of the increasing access to high-quality cameras, health care personnel can take photos of patients' wounds without depending on specialized equipment, and provide decentralized care with enough quality.

Nowadays, mobile devices can acquire high-quality images and act as a feasible, convenient, and cheap alternative for image acquisition [Dorileo et al. 2010]. One scenario where these images are especially useful is in the analysis of chronic skin wounds, often referred to as ulcers [García-Zapirain et al. 2018]. Chronic ulcers often

coincide with other morbidities, adding physical and psychological strain, further limiting a patient's ability to make their way to a specialist, usually located in urban centers [Gamus et al. 2019]. Skin ulcers occur due to reduced blood circulation in lower extremities, injuries, infections, tumors, and other skin conditions [Dorileo et al. 2010]. The visual appearance of these wounds provides clinical signs that may help healthcare practitioners in the diagnosis and treatment. The analysis focusing on the images' content can benefit specialists, providing clues based on past cases, helping them during diagnosis and follow-up procedures. Evaluating whether ulcers contain granulation, fibrin or necrosis allows the healthcare practitioner to analyze the evolution of the patient's condition.

Mobile devices lack on storage and processing power. Having the patients' data and images stored in a database allows not only to follow the patients' condition, but also to provide further knowledge to improve their treatment and the organization of the health system as a whole. Traditional data, such as text and numbers, can be queried using the traditional Database Management Systems (DBMS). On the other hand, considering the content of complex data such as images, videos, and audios require special treatment to store, represent and query them [Traina-Jr. et al. 2000]. DBMS support the analysis of complex data by providing a fast, self-organized, and scalable operations to store and retrieve complex objects and associated information. The most employed query type performed over complex data in DBMS are the similarity queries. They consider the object content, and retrieves the most similar objects in the database, given a similarity threshold or a number of similar objects to retrieve [Lu et al. 2017]. Similarity queries require representing each complex object as low-level feature vectors, retrieved by executing Feature Extraction Methods (FEM). A similarity measure computes the similarity between pairs of objects, often done by a Distance Function (DF) that gives the dissimilarity between pairs of (feature vectors from the) objects.

We are particularly interested in skin ulcer images. The literature has explored representing ulcer using color and texture features for classification purposes [Chino et al. 2018]. Other studies have proposed specific measures to improve the content-based image retrieval and classification of ulcers [Goyal et al. 2017]. However, to the best of our knowledge, every study has focused on the analysis of images using specific applications, without maintaining historical data. There is a lack of approaches with a practical application, considering real-world scenarios, and there is no proper DBMS support to carry similarity-based queries nor to store and use past cases. Past successful cases can be useful to empower the analysis of new patients' data. Existing practical approaches to support ulcer patients' care in remote areas or at home focus on exchanging textual information regarding their conditions [Pedro et al. 2011]. Alternative solutions provide simple information regarding an input image, such as highlighting the wound area using a temperature-based image processing [Fraiwan et al. 2018], but requires specialized equipment such as a mobile thermal camera.

In this work, we propose **ULEARn**, a framework to support health professionals in the analysis of skin ulcer. **ULEARn** is a simple, but accurate and cheap solution for ulcer image analysis, storage and organization. Figure 1 shows how the physician interacts with the system. Briefly, the contributions of this work are:

- The DBMS-based **ULEARn** framework, employed to store images and related information, which supports content-based image retrieval tasks and analysis appli-

Figure 1. Interaction between the physician, `ULEARn-App` and `ULEARn`.

cations with indexing support and a self-organizing structure.

- An ulcer image analysis application, which communicates with **`ULEARn`**, and provides helpful information regarding the ulcer images taken by the physician.
- The mobile application **`ULEARn-App`**, which allows the use of **`ULEARn`** in the medical practice.
- Analysis of a case study that validates **`ULEARn`** and **`ULEARn-App`**.

Paper outline. Section 2 describes relevant background. Section 3 presents related work. Section 4 introduces **`ULEARn`**, the proposed framework to analyze ulcer images. Section 5 presents a experimental analysis. Finally, Section 6 gives the conclusions.

2. Background and Basic Concepts

This section is divided in two parts. First we describe the content-based comparison of images, which can be carried within a database management system. Then we cover image analysis approaches to infer knowledge from ulcers' visual characteristics.

2.1. Content-based comparison of images

Traditional DBMS deals with scalar data such as numbers and dates, which are compared by order $(<, \leq, \geq, >)$ and identity relations $(=, \neq)$ [Traina-Jr. et al. 2000]. However, they have very little use to compare images, where similarity-based operators are better suited. Similarity is studied when performed over a dataset of complex objects. To use those concepts in a DBMS, we assume that the data to be searched is the set of complex values stored in an attribute of a relation. This approach is advantageous, as it allows storing structured information about each object in other attributes in the same tuple.

To make the comparison of complex objects possible, we employ Feature Extractor Methods (FEMs) to represent them and Distance Functions (DFs) to compare the objects. FEMs outputs feature vectors as signatures of the objects' content, with low-level descriptions of the complex data. For instance, to represent images' visual content we can use color, shape and/or texture. A DF is a metric used to compute the (dis)similarity between pairs of complex objects. Examples of DFs are those from the Minkowski family, which includes the Euclidean and Manhattan distances. Let the feature vectors of two complex objects s_i and s_j be respectively v_i and v_j. A DF δ computes the distance between pairs of complex objects $< s_i, s_j >$, such that $v_i = FEM(s_i), v_j = FEM(s_j)$ and $\delta : \mathbb{R}^m \times \mathbb{R}^m$. The DF is a metric if it complies with the following properties:

- *Non-negativity:* $\delta(s_i, s_j) \geq 0$;
- *Identity of the indiscernible:* $\delta(s_i, s_j) = 0 \Rightarrow s_i = s_j$;
- *Symmetry:* $\delta(s_i, s_j) = \delta(s_j, s_i)$;
- *Triangular inequality:* $\delta(s_i, s_j) \leq \delta(s_i, s_k) + \delta(s_k, s_j)$.

Similarity search in datasets only requires defining the search operators, usually the range and k nearest neighbors queries. However, similarity search in DBMS also requires defining the underlying comparison operators, namely Range and k-Nearest Neighbors (k-*NN*) comparisons. Let s_1 and s_2 be two complex objects. The range comparison $s_1\ Rng(\delta, \xi)\ s_2$ returns *TRUE* if and only if $\delta(s_1, s_2) \leq \xi$, where ξ is a similarity threshold (or similarity radius), and δ measures the similarity between complex objects. The k-*NN* comparison $s_1\ k\text{-}NN(\delta, k)\ s_2$ returns *TRUE* when s_1 is one of the k nearest elements to s_2 in the attribute's active domain, according to the similarity measure δ.

We are particularly interested in retrieving images within a similarity threshold from an image used as a query center. Content-based image retrieval (CBIR) systems aim at retrieving images using similarity-based queries. Examples of well-known FEMs for images are those defined by the MPEG-7 standard [MultiMedia 2002], which includes, among others, the Color Layout, Color Structure, and Edge-Histogram FEMs. Other examples, employed on the ulcer analysis task [Chino et al. 2018], are the Local Binary Patterns (LBP) and the Color Histogram. Let $\mathbb{S}$ be an image domain, $S \in \mathbb{S}$ be the set of stored images, $s_i, s_j \in S$ be two image objects, and φ be the FEM employed to obtain the image's signature. When employed over s_i, φ outputs an m-dimensional feature vector $v_i = (f_1, f_2, \ldots, f_m) \in \mathbb{R}^m$, where each dimension $f_j \mid 1 \leq j \leq m$ represents a visual feature, and is represented as an attribute.

A Metric Space (MS) is defined by a pair $< \mathbb{S}, \delta >$, where $\mathbb{S}$ represents the domain of valid objects and δ is a metric [Ciaccia et al. 1997]. This definition allows using Metric Access Methods (MAMs) to index complex objects, speeding-up similarity queries. MAMs consist of indexing structures for complex data storage and retrieval, and examples of well-known MAMs are the Slim-Tree [Traina-Jr. et al. 2000] and the M-Tree [Ciaccia et al. 1997]. We employed the Slim-Tree to index images in this work.

2.2. Image analysis approaches

An image I consists of a set P of n pixels. Let R, G and B be respectively the color channels red, green and blue in the RGB color space. A pixel $p_i \in P$ is a tuple $p_i = (R_i, G_i, B_i)$, which is the pixel intensity of each color channel of RGB color space. A superpixel S_j corresponds to a subset of w pixels $S_j = \{p_j \mid 1 \leq j < w, w \leq n\}$, obtained from a region from I, such that $S_j \subseteq I$, and $w = n$ if $S_j = P$. Superpixels are usually used to subdivide images in color-coherent regions, supporting classification and segmentation tasks. Examples of superpixel methods from literature are SLIC, Felzenswalb and Quick Shift [Achanta et al. 2012].

Let $\mathcal{C}$ be a set of categorical classes, $\mathcal{V} \subset \mathbb{R}^m$ be a set of vectors that represent a particular set of images S_i by using FEM φ_i, and $\mathcal{T}$ be the training set of images, previously classified by an expert with classes from $\mathcal{C}$. A supervised classifier ζ, trained over $\mathcal{T}$, builds a model able of predicting a class $c \in \mathcal{C}$ for each given unclassified image I_u, considering its corresponding feature vector v_u. Simply putting, the classifier corresponds to a function ζ that maps an image feature vector v to a class $c \in \mathcal{C}$. ζ can also be used to classify a superpixel S using the set of available classes. In this work, we consider a set of classes $\mathcal{C}_{ulcers} = \{normal, granulation, fibrin, necrosis\}$, which is useful to the ulcer application scenario. The classifiers used in our analysis and retrieval modules are the Naive-Bayes, K-Means, IBL, and MLP approaches [Zaki and Meira Jr. 2014]. We combined these classifiers with SLIC method to segment and classify skin ulcer images.

A feature vector with the content of the entire image is a global representation of the image. In contrast, feature vectors from specific image regions, such as superpixels, are region-based representations, or local features. Examples of local features are the Bag-of-Visual-Words (BoVW) technique and its variations. BoVW assigns the local features extracted from an image region into one or more visual words in a dictionary. In [Chino et al. 2018] the authors proposed ICARUS, a highly-accurate approach based on BoVW to identify patterns in skin ulcers and improve the content-based retrieval task. We use ICARUS to retrieve similar wound regions of skin ulcer images.

3. Related Work

Existing works related to our approach refers to (i) DBMS-like systems, supporting content-based image retrieval and associated tasks, as well as (ii) existing applications focused on the remote support and treatment of patients with skin ulcer conditions.

Regarding (i), the benefits of CBIR and knowledge discovery on complex data through similarity queries can be applied to several real-world applications, including medicine. However, when CBIR uses a DBMS, the latter is limited to supply the CRUD operations (`CREATE`, `READ`, `UPDATE` and `DELETE`). As a consequence, queries based on predicates over the image attribute and scalar attribute associated to it – such as the capture date or the author – must execute the similarity predicate (over the image) in a separated query engine and then combine it with the result of the scalar predicate from the DBMS. By allowing the DBMS to perform CBIR, both similarity and scalar predicates are integrated, and the queries can be answered using optimized query processing techniques.

The inclusion of CBIR in DBMS requires, at the very least, the provision of FEMs and DFs. Additionally, MAMs can be employed to speed-up queries processing. In that regard, several works integrating CBIR in DBMS can be found in the literature. SIREN [Barioni et al. 2006] proposed a seamless integration of similarity and scalar predicates by extending the SQL to represent similarity operators and implementing a middleware API that combines similarity predicates executed externally with scalar predicates executed by the DBMS. However, the middleware is limited to a query plan that always performs similarity predicates first. RAFIKI [Nesso-Jr. et al. 2018] and FMI-SiR [Kaster et al. 2010] tend to overcome this limitation by implementing similarity retrieval resources through user-defined-functions and extensible interface provided by the DBMS, with RAFIKI being a solution for PostgreSQL and FMI-SiR for Oracle. Both solutions were tested on medical image retrieval scenarios. Since this work uses Oracle as its DBMS, we employed the FMI-SiR similarity retrieval solution to perform queries over the ulcer images. During the execution of a query that uses similarity predicates, the query processor changes the context from SQL to the solution, executes the user-defined function for similarity retrieval and returns information to the query executor. Since it is integrated into the DBMS, these functions can be invoked anywhere in the query plan.

Existing applications focused on the support and treatment of skin ulcers (ii) usually do not analyze the images sent by the users. Those are often referred to as telemedicine applications for wound care, which involves the application of telecommunication technologies to remotely exchange medical information [Chittoria 2012]. In [Fraiwan et al. 2018], the authors proposed a mobile application to work with feet skin ulcers, which uses temperature-based image processing to detect ulcers. However, they

limit their application to images of feet, and also require a mobile thermal camera for image acquisition. Additionally, all processing tasks are performed in the mobile system itself, working with only one image at a time. mULCER [Pedro et al. 2011] is a mobile application to help the ulcer patients' care. mULCER performs image analysis of a user photo, as well as related information regarding the resulting analysis. They do not provide information related to how the images and information are stored, maintained nor processed, and limit the analysis results to only informative texts regarding the sent photo. A systematic review of applications for the prevention of pressure ulcers is presented in [Marchione et al. 2015]. The majority of the reported studies work with sensors and specialized equipment, patient bed position control, and informative textual descriptions. None of the studies described in their work provide a fully-automated analysis of image ulcers to support physicians in daily medical practice.

In this work, we show a DBMS-based solution that support the processing, organization, and communication among different components of a framework to assist the analysis of ulcer images. The proposed image analysis tasks, connected with an similarity-aware extended DBMS, provide the proper tools to assist the medical practice.

4. `ULEARn`: The Proposed Framework

This section presents the **UL**cer Imag**E** **A**nalysis and **R**etrieval (**`ULEARn`**) framework, developed to analyze and store skin ulcer images and related information. Figure 2 depicts an overview of **`ULEARn`**. The framework supports healthcare practitioners through images' content-based retrieval and processing tasks. The professional takes a photo of the ulcer being treated and submits it for analysis. The framework stores relevant information, allowing him/her to follow the evolution of the patient. **`ULEARn`** controls the interaction among different modules that can be swiftly integrated to a DBMS to take into account data from a set of images, and not from just one, as is the current state-of-the-art. This work is the first effort in such integration, pointing out resources for storage and query processing required from the DBMS. The ability to easily integrate modules of customized image analysis to a CBIR system is our leading research contribution.

`ULEARn`'s workflow is composed of four main layers: (I) Mobile Application and Data Exchange; (II) Application Manager; (III) Image Analysis; and (IV) Storage and Similarity-Based Retrieval. Layers III and IV are inside **`ULEARn`**, while Layers I and II work as services above the proposed framework, making use of its features and controlling the user/framework interaction. We explain each layer in the next sections.

4.1. Layer I: Mobile Application and Data Exchange

`ULEARn` provides an infrastructure for mobile applications to communicate with the database and the image analysis methods. As mobile devices have limited capability to store and process large amounts of information, the support of a DBMS to store and process the data is primordial. We implemented the mobile app **`ULEARn-App`** as a proof of concept to test the proposed framework (see implementation details in Section 5). **`ULEARn-App`** provides a proper authentication for physicians to access the app and past patients' information. As the application stores personal information, the access to the app is restricted for trusted users. **`ULEARn-App`**'s main task is sending an image taken from the patient. The user also stores textual information in the database, such as patients'

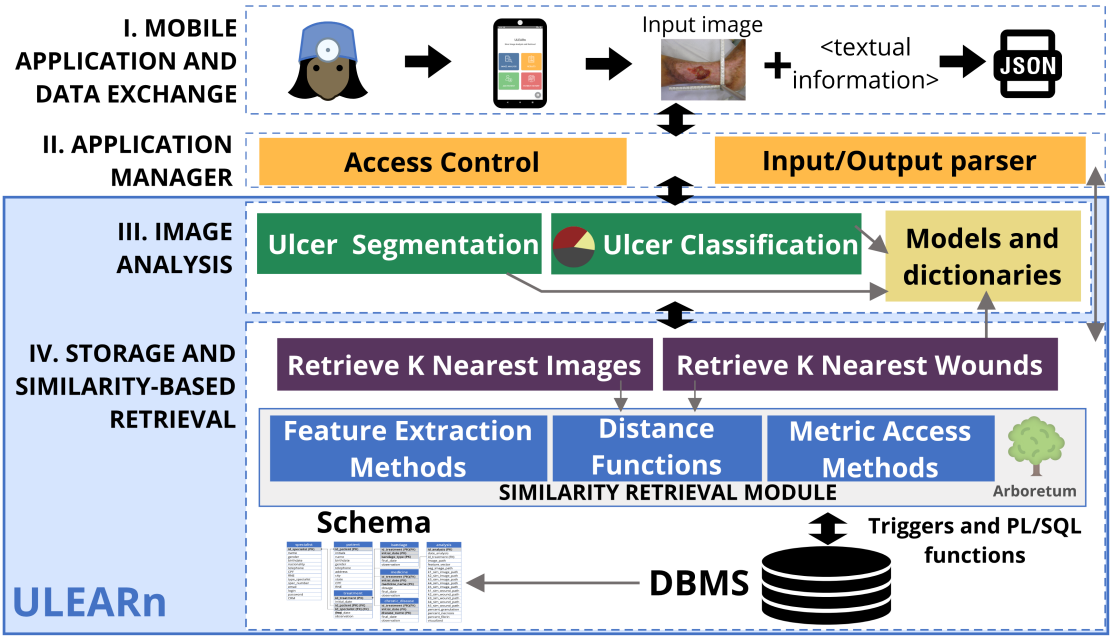

Figure 2. `ULEARn` framework overview.

data and treatment history. All data is encapsulated in a `JSON` file and sent to the server with a request for the analysis results. When the remaining layers finish processing, the app receives the results to be displayed for the physician, also as a `JSON` file.

4.2. Layer II: Application Manager

The application manager intermediates the communication between the app and the framework, and performs the access control and the input/output parser. The access control corresponds to authenticating the physician using the app, connecting to the framework, consulting, and inserting associated information. For security reasons, a physician can only access the information of patients that are linked to he/she. The input/output parser calls each of the image analysis and the storage/retrieval modules. Once the analysis are done, the parser sends the results back to Layer I.

4.3. Layer III: Image Analysis

In order to provide the relevant feedback to the physician, **`ULEARn`** implements two image analysis tasks, Ulcer Segmentation (*US*) and Ulcer Classification (*UC*). We analyzed *US* and *UC* with different off-the-shelf classifiers. MLP and 3NN presented the best F-Measure results for *US* and *UC* respectively. However, as **`ULEARn`** is a modularized solution, its customization by exchanging each functionality as a black box inside the framework is feasible and direct.

Ulcer Segmentation (US). Given the patient image as input, *US* segments the image into a set of superpixels $S_i \in \mathcal{S}$. *US* extracts the feature vector of each superpixel region using the Color Histogram FEM. Then, *US* classifies each feature vector according to the class set $\mathcal{C}_{wounds} = \{wound, not\_wound\}$. *US* uses the 3NN classifier and a pre-trained model stored in the framework. As a result, *US* composes a segmented image with only the regions classified as *wound* ones. Regions classified as *not_wound* are painted black, so

only the segmented wound will be depicted in the resulting image. The segmented image is used as input for the next step, the ulcer classification (*UC*).

Ulcer Classification (UC). Given the segmented image, *UC* extracts its feature vector with Color Histogram FEM. *UC* classifies the feature vector according to the different healing stages $\mathcal{C}_{stages} = \{fibrin, granulation, necrosis\}$. *UC* uses the MLP classifier, and outputs the probability of the image to contain each class in $\mathcal{C}_{stages}$.

4.4. Layer IV: Storage and Similarity-Based Retrieval

Following we detail how **ULEARn**: (i) stores all information regarding the physicians, the patients, the treatments, and the analysis results in a relational database; (ii) retrieves the most similar images (*kNI*) and wounds (*kNU*) to each query. **ULEARn** extends the DBMS Oracle, primarily used for storage purposes only, to also perform feature extraction, distance calculations, complex data indexing, and the execution of similarity queries. These functionalities correspond to the similarity retrieval module, depicted in Figure 2, provided by the Arboretum library[1]. **ULEARn** provides an interface to Arboretum, leaning on Oracle for calling the functions through triggers and PL/SQL functions.

(i) Storage. Figure 3 presents **ULEARn**'s data schema. Tables *specialist* and *patient* store personal information, and Table *treatment* links the specialist to the patient and stores specific information regarding its condition. A single treatment can have one or more dressings, medication, and chronic diseases registered, which allows the framework to maintain the medical records of the patient for further inquiries. Also, table *analysis* stores the patient's photos sent by the app, and all image analysis results performed in Layer III (Image Analysis).

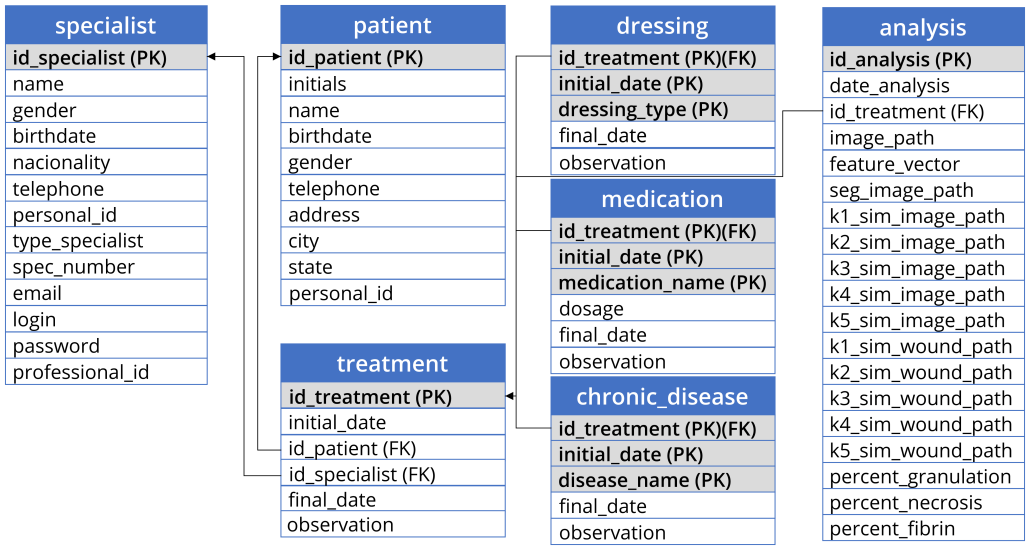

Figure 3. ULEARn schema, with information regarding the patients, professionals, treatment and analysis results.

(ii) Similarity-Based Retrieval. **ULEARn** relies on past cases to perform similarity-based queries for the two following tasks. They rely on the extensions performed over the DBMS, in order to allow the similarity-based representation and comparison of images.

[1]The Arboretum library: https://bitbucket.org/gbdi/arboretum/src/master/.

Retrieve K Nearest Ulcers (kNU). Given the patient's image as input, *kNU* segments the image into a set of superpixels, using an approach based on bag-of-visual-words to represent the regions' visual patterns. With the segmented image, *kNU* retrieves the k most similar images stored in the database, using the similarity retrieval module, which was included in the DBMS as depicted in Figure 2.

Retrieve K Nearest Images (kNI). Given the patient's image as input, *kNI* inserts the received image into the DBMS, which performs a kNN query using past cases, supported by a metric access method to speed-up the task. This is also relevant because the wound heals/reacts differently in different body parts, and considering the skin around the wound area is relevant to evaluate the patients' condition. The k most similar images are returned to the user, considering the entire image's similarity to the previously stored ulcer images, from other patients.

For *kNU*'s and *kNI*'s similarity retrieval tasks, the DBMS is in charge of receiving the image from the application manager layer of **ULEARn**, extracting its feature vector and storing it as a `BLOB` type attribute. The query of Statement 1 performs this step.

Statement 1. Receiving, processing and storing a new image.

```
DECLARE
   fvector BLOB;
   result PLS_INTEGER;
BEGIN
   dbms_lob.createtemporary(myblob, true);
   result := extractFeatureVector('image_path',fvector);
   INSERT INTO ANALYSIS(...,fvector,...) VALUES (...,fvector,...);
   dbms_lob.freetemporary(myblob);
END;
```

Statement 1 exemplifies feature extraction and storage, executed during a new image analysis, using the Color Layout FEM, which is executed by the similarity retrieval module. The corresponding feature vector is the *fvector* variable, which is then stored in the database along with other information (represented by '...'). To retrieve the k most similar images considering the Color Layout FEM and the Euclidean DF, **ULEARn** performs the query shown as Statement 2. Consider a newly inserted image using Statement 1, identified by *id_analysis*$= 10$ as the query center. The query in Statement 2 will retrieve its 5 most similar images considering Color Layout and the Euclidean distance. Remember from Section 3 that our employed Similarity Retrieval Solution implements its resources as user-defined-functions and extensible interfaces. Figure 4 shows the query workflow, for the insertion of a new image.

Statement 2. Retrieving the 5 most similar images from image 10.

```
SELECT id_analysis, image_path FROM analysis
WHERE euclidean_knn(fvector,(SELECT fvector FROM analysis
                             WHERE id_analysis = 10)) <= 5;
```

5. Experimental Analysis

Setup. We implemented **ULEARn-App** for Android OS using Kotlin, the web framework Flask to manage the communication between the app and the server, and all algorithms

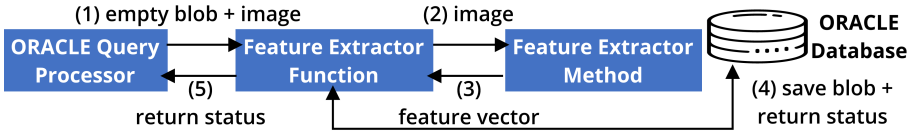

Figure 4. Interactions of the Similarity Retrieval Module to insert a new image.

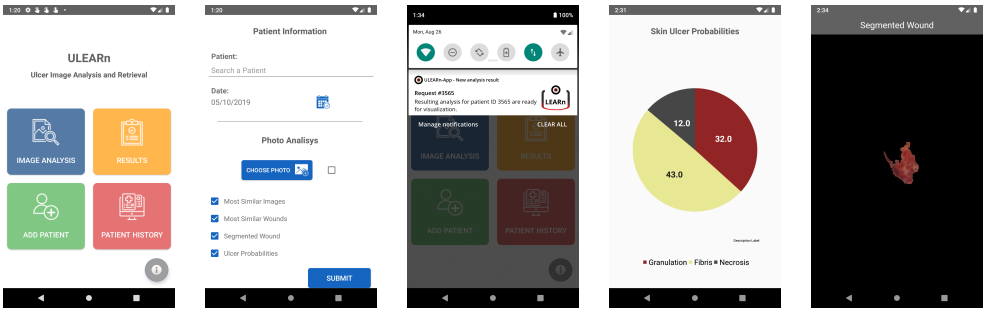

Figure 5. Screenshots of `ULEARn-App`.

were implemented in Python 3.6.5. The server is an Intel Xeon(R) X5650 2.67GHz 4vCPU, 16GB virtual machine, running Ubuntu 16.04 LTS, and the library *cx_Oracle* 7.1.3 to connect to the Oracle DB-12c - 64bit server.

Case Study. Figure 5 shows examples of screenshots taken from **`ULEARn-App`**[2]. In summary, the app has the following features implemented: login, physician, patient and treatment registration, analysis' results, and patient history, serving as a follow-up chart for the physician. We consider as the analysis focus the feedback from a physician from the Federal University of São Carlos, that works with ulcer patients. The ulcers specialist emphasizes that obtaining a response *within five minutes* is primordial. This allows them to take a photo from the patient, at the beginning of the visit, and obtain a proper response to improve the intervention while they are still working with the patient for treatment/therapy. Still, **`ULEARn-App`** is interactive, and while the physician waits for a response, the app will release its functionalities, not freezing up while **`ULEARn`** remotelly processes the analysis request. This way, the user can use their smartphone normally, and can also concomitantly send other images for analysis. The app will show a notification informing when the analysis is ready for visualization (3rd screen of Figure 5). We evaluate our framework using a physicians' provided dataset of ulcer images, testing the various components of **`ULEARn`** in terms of execution time. The dataset contains 216 images, consisting of ulcer photos taken by specialists from ulcer patients using smartphones.

Time Analysis. Figure 6 gives the execution time of each image analysis component, with different resolution (width varying from 320 to 3,264 pixels) using $k = 5$ for k-*NN* Retrieval. The segmentation and classification tasks (a) take up to 40.9 seconds but is faster as image resolution decreases. The wound retrieval (b) presents nearly constant execution time from the resolution of 1280 pixels, requiring at most 10.3 seconds. The image retrieval (c) used 50 random query objects, and it was much faster than the other analysis tasks, as it is performed in the DBMS using the Slim-Tree MAM. Its execution time was higher regarding the increased resolution, performing in up to 4.1 seconds.

Figure 7 shows the full processing execution time considering the sum of every

[2]The app and further details are given on https://github.com/mtcazzolato/ulearn-app.

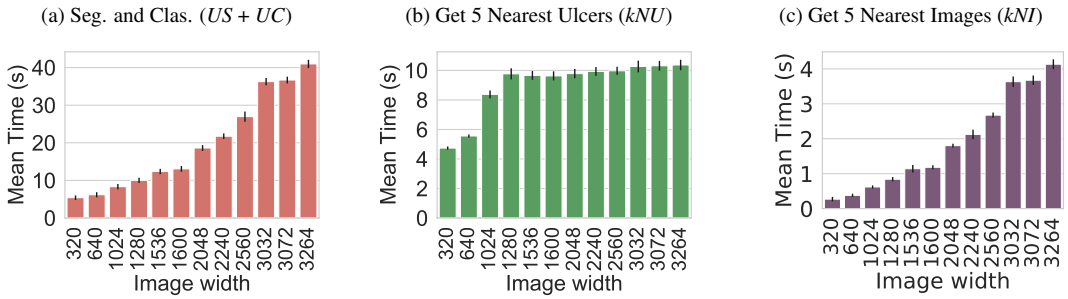

Figure 6. Mean execution time for the image analysis task of `ULEARn`.

analysis and database operations (d), and also considering the average app interaction and network latency times from **ULEARn** (e). In (d), we observe that the sum of every analysis algorithm shows a linear growing behavior on the image resolution. Now regarding **ULEARn-App** and **ULEARn** (e), we observe that the entire processing pipeline occurs within an execution time of up to 65.8 seconds, for high-resolution images. Notice that, part of this time corresponds to sending the image through the network. All in all, **ULEARn** has shown to perform in a feasible interval of time, serving its purpose and allowing the physician to obtain feedback within the desirable time frame.

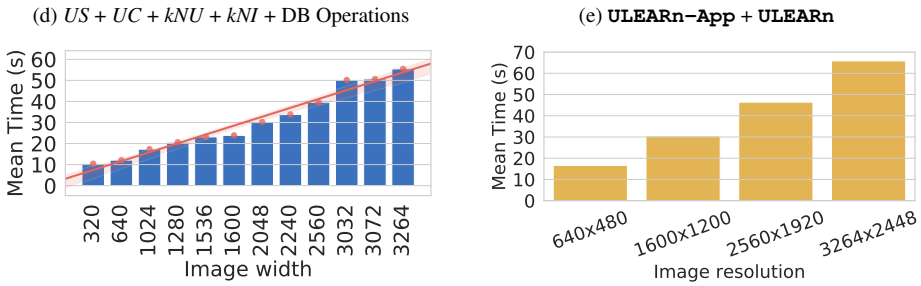

Figure 7. Mean execution time for all processing of `ULEARn` and `ULEARn-App`.

6. Conclusions

We presented **ULEARn**, a framework that supports the analysis and storage of skin ulcer images. **ULEARn** provides relevant feedback for health professionals regarding patients' ulcer, based on a DBMS to support the processing, organization and communication between its different components. **ULEARn-App** is an Android app that servers as a proof of concept regarding the usefulness of **ULEARn**. Experiments showed that **ULEARn** is up to *4.6 times faster* than the specialist's expectation, even when working with high resolution images. Specialists can use **ULEARn-App** in the day-to-day clinical scenario, for a patient checkup or in places that lack specialized wound care intervention. Future work include adding a temporal analysis on the patients' condition, and further test of **ULEARn-App** in real scenarios. Among the many motivations for proposing **ULEARn** and **ULEARn-App** is to allow physicians in the data acquisition, storage, and retrieval for analysis. We foresee that **ULEARn** will make it possible for physicians and data analists, in the near future, to collect a larger number of images and associated information, further improving the classification and segmentation models, including the use of Deep Learning approaches. Also, **ULEARn** can be easily adapted to other application contexts.

Acknowledgments

This research was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, by the São Paulo Research Foundation – FAPESP (grants #2018/24414-2, #2016/17078-0, #2016/17330-1, #2014/25125-3), and the National Council for Scientific and Technological Development (CNPq).

References

Achanta, R. et al. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 34(11):2274–2282.

Barioni, M. C. N., Razente, H. L., Traina, A. J. M., and Traina-Jr., C. (2006). SIREN: A similarity retrieval engine for complex data. In *VLDB*, pages 1155–1158.

Chino, D. Y. T. et al. (2018). Icarus: Retrieving skin ulcer images through bag-of-signatures. In *CBMS*, pages 82–87.

Chittoria, R. K. (2012). Telemedicine for wound management. *Indian J Plast Surg*, 45(2):412–417.

Ciaccia, P., Patella, M., and Zezula, P. (1997). M-tree: An efficient access method for similarity search in metric spaces. In *VLDB*, pages 426–435.

Dorileo, E. A. G., Frade, M. A. C., Rangayyan, R. M., and Azevedo-Marques, P. M. (2010). Segmentation and analysis of the tissue composition of dermatological ulcers. In *CCECE*, pages 1–4.

Fraiwan, L., Ninan, J., and Al-Khodari, M. (2018). Mobile application for ulcer detection. *TOBEJ*, 12:16–26.

Gamus, A., Keren, E., Kaufman, H., and Chodick, G. (2019). Synchronous video telemedicine in lower extremities ulcers treatment: A real-world data study. *IJMI*, 124:31–36.

García-Zapirain, B., Elmogy, M., El-Baz, A., and Elmaghraby, A. S. (2018). Classification of pressure ulcer tissues with 3d convolutional neural network. *MBEC*, 56(12):2245–2258.

Goyal, M., Yap, M. H., Reeves, N. D., Rajbhandari, S., and Spragg, J. (2017). Fully convolutional networks for diabetic foot ulcer segmentation. In *SMC*, pages 618–623.

Kaster, D. S., Bugatti, P. H., Traina, A. J. M., and Traina-Jr., C. (2010). FMI-SiR: A flexible and efficient module for similarity searching on oracle database. *JIDM*, 1(2):229–244.

Lu, W., Hou, J., Yan, Y., Zhang, M., Du, X., and Moscibroda, T. (2017). MSQL: efficient similarity search in metric spaces using SQL. *VLDB J.*, 26(6):829–854.

Marchione, F., Araújo, L., and Araújo, L. (2015). Approaches that use software to support the prevention of pressure ulcer: A systematic review. *IJMI*, 84(10):725–736.

MultiMedia, I. (2002). Mpeg-7: The generic multimedia content description standard, part 1. *IEEE MultiMedia*, 9(2):78–87.

Nesso-Jr., M. R. et al. (2018). RAFIKI: retrieval-based application for imaging and knowledge investigation. In *CBMS*, pages 71–76.

Pedro, L. M. C. C., Rodrigues, J. J. P. C., and Martins, H. M. G. (2011). mUlcer – a mobile ulcer care record approach for integrative care. In *EIS*, pages 392–401.

Traina-Jr., C., Traina, A. J. M., Seeger, B., and Faloutsos, C. (2000). Slim-trees: High performance metric trees minimizing overlap between nodes. In *EDBT*, pages 51–65.

Zaki, M. J. and Meira Jr., W. (2014). *Data Mining and Analysis - Fundamental Concepts and Algorithms*. Cambridge University Press, New York, NY, USA.