Original article

Generation of Chest X-Ray Image Datasets for Training Deep Neural Networks

Generación de Conjuntos de Imágenes Radiográficas de Tórax para el Entrenamiento de Redes Neuronales Profundas

Henry Blanco Lores[1*] 0000-0003-3132-5759
Jef Vandemeulebroucke[2] 0000-0001-5714-3254

[1] Center for Medical Biophysics. Santiago de Cuba. Cuba.
[2] ETRO Department at Vrije Universiteit Brussel. Brussels. Belgium.

[*]Autor para la correspondencia. Correo electrónico: henry.blanco@uo.edu.cu

ABSTRACT

Deep neural network models represent the main reference for addressing automatic image classification problems. The successful training of this type of models depends on large amounts of labeled images. The current shortfall of labeled images in the radiology domain is a major obstacle for applying deep neural network models to this environment, and it is that the availability of labeled medical images for training this type of models remains insufficient.

In this work, we address this problem through the creation of an “inverted index” of medical images. This is a data structure taken from the field of information retrieval and adapted to the radiology application domain. The fundamental idea is to organize images of an imaging repository, just using the image tags as an index. This way, it is possible to query the inverted index for different sets of anomalies or labels and to efficiently generate a wide variety of image sets for training deep neural network models.

As a use case, we applied this solution to chest X-ray images from the PadChest repository. It was possible to efficiently organize its 160,000 images using an inverted index based on 174 anomalies (labels). Regarding the image access mechanism, provided by the authors of PadChest, the inverted index helped reduce the number of steps required to access images associated with a given anomaly by 10 times. By combining the inverted index with a hierarchy of radiological terms, which interrelates the anomalies present in the repository, it is possible to generate a huge variety of image sets to train deep neural network models for image classification tasks.

Keywords: deep learning; inverted index; medical images classification; convolutional neural networks; dictionary; machine learning; supervised learning; labeled image repositories; generated training datasets; indexing criteria.

RESUMEN

Los modelos de redes neuronales profundas, principal referente para abordar problemas de clasificación automática de imágenes, dependen de grandes cantidades de imágenes etiquetadas para su entrenamiento. Actualmente, esto resulta ser un importante obstáculo para aplicar exitosamente modelos de redes neuronales profundas al entorno radiológico. Y es que la disponibilidad de imágenes médicas etiquetadas para entrenar este tipo de modelos, es aún insuficiente.

En este trabajo, esta problemática es abordada a través de la creación de un índice invertido de imágenes médicas. Esta es una estructura de datos tomada del campo de recuperación de información y adaptada al dominio de aplicación radiológico. La idea fundamental es organizar las imágenes de repositorios imagenológicos, utilizando como índice las etiquetas asociadas a las imágenes. De aquí, la posibilidad de generar eficientemente una amplia variedad de conjuntos de imágenes para entrenar modelos de redes neuronales profundas.

Como caso de uso, aplicamos esta solución a imágenes radiográficas de tórax del repositorio, *PadChest*. Fue posible organizar sus 160 mil imágenes de forma eficiente a través de un índice invertido, basado en 174 anomalías (etiquetas). Respecto al mecanismo de acceso a las imágenes, brindado por los autores de *PadChest*, el índice invertido contribuyó a reducir 10 veces la cantidad de pasos necesarios para acceder a imágenes asociadas a una anomalía dada. Al combinar el índice invertido con una jerarquía de términos radiológicos, que interrelaciona las anomalías presentes en el repositorio, es posible generar una enorme variedad de conjuntos de imágenes para entrenar modelos de redes neuronales profundas en tareas de clasificación de imágenes.

Palabras clave: aprendizaje profundo; índice invertido; clasificación de imágenes médicas; redes neuronales convolucionales; diccionario; aprendizaje automático; aprendizaje supervisado; repositorios de imágenes etiquetadas; conjuntos de datos de entrenamiento generados; criterios de indexación.

Recibido: 07/04/2025
Aprobado: 06/05/2025

Introduction

It could be said that, if a picture is worth a thousand words, for convolutional neural networks, thousand images are worth a good prediction. The availability of appropriate datasets for training convolutional neural networks is a major issue in achieving high levels of prediction in image classification tasks[1,2]. This problem is particularly relevant in the field of medical imaging. Despite of the enormous amount of images generated by imaging equipment, the lack of labeled datasets poses strong limitations regarding automatic image classification tasks[3,4]. The shortfall of expert or trained personnel (e.g., radiologists) capable to properly label medical images[5,6], plus the strong ethical-legal implications that this information entails by itself[7,8] makes this scenario difficult to navigate for artificial intelligence researchers. Hence the need to take full advantage of the labeled medical image repositories currently available.

Convolutional neural networks (CNNs) are currently the reference models for tackling visual classification tasks automatically[9,10]. These models have outperformed other traditional models already established in the field of machine learning, even in important image classification competitions such as: "ImageNet Large Scale Visual Recognition Challenge"[11,12]. Inspired by the neural organization of the visual cortex of the human brain, CNNs are capable to automatically identify underlying patterns in the processed data. On this basis, and based on the observation of many examples (training process), CNNs can perform classification tasks with a high degree of accuracy[13]. This success is largely due to the tens or hundreds of neural layers that make up these models. This facilitates the learning of a large variety of patterns (not perceptible to the naked eye) in the processed data. However, this success comes at a high price. The availability of large quantities of labeled images (above thousands of images) is essential. This is similar to the case of a linear equation with multiple variables, which requires multiple equations for its solution. In the same way, a CNN model requires many images to correctly tune millions of neural connection towards a particular expected output.

There exists several online medical image repositories providing labeled data for training machine learning models on classification tasks[14,15]. However, the availability of labeled images covering a wide variety of anomalies or pathology, is still not sufficient. Chest X-rays (CXR) image repositories do not escape from these limitations. Only few repositories exhibit a wide variety of labels, anomalies or findings in X-rays images. Currently, the most relevant CXR image repositories online are the following: *PadChest*[16,17], *Mendeley*[18], *MIMIC-CXR*[19], *ChestXray*[20] and *CheXpert*[21]. In particular, the last two repositories stand out for the large number of images available (|*ChestXray*| > 112K images and |*CheXpert*| > 224K images). However, only 14 anomalies or findings have been labeled in their images: *atelectasis*, *consolidation*, *pneumonia*, *cardiomegaly*, *pneumothorax*, *pleural effusion*, *infiltration*, *edema*, *emphysema*, *hernia*, *nodule*, *fibrosis*, *tumor mass* and *pleural thickening*. This limitation in terms of "anomaly or label diversity" makes harder to train CNN-based models on other anomalies of interest such as: COPD signs, scoliosis, aortic elongation, heart insufficiency, among others.

The exception to this low variety of anomalies and labeled images is the CXR image repository, "*PadChest*". This is an open-access repository, with more than 160K images associated with 67K studies collected from 2009 to 2017 and reported by radiologists from the San Juan hospital, in Alicante, Spain. The radiology reports cover more than 170 findings or anomalies, located in more than 100 anatomical regions. No fewer than 60 of these findings are present in at least 1,000 images (see Table 1). This scenario provides better opportunities to train CNN models in classification tasks on CXR images, labeled with a large variety of anomalies. Potentially, it is possible to consider 2^{60} different sets of CXR images with anomalies or findings of interest. Hence the importance of organizing these images efficiently for a faster retrieval and for the flexible generation of large and diverse sets of images.

Table 1. Findings in the repository *PadChest* with the largest number of images associated.

anomaly or finding	n. imgs.	anomaly or finding	n. imgs.
normal	50,390	kyphosis	5,215
COPD signs	23,280	laminar atelectasis	5,190
cardiomegaly	15,022	vertebral degenerative change	4,878
unchanged	14,334	vascular hilar enlargement	4,517
aortic elongation	10,824	nodule	3,748
pleural effusion	9,853	fibrotic band	3,713
scoliosis	8,333	apical pleural thickening	3,625
pneumonia	7,953	pacemaker	3,508
interstitial pattern	7,799	venous catheter via jugular vein	3,204
chronic changes	7,337	calcified granuloma	3,194
infiltrates	7,089	callus rib fracture	2,967
costophrenic angle blunting	6,784	atelectasis	2,905
air trapping	6,147	sternotomy	2,849
alveolar pattern	5,738	volume loss	2,757
NSG tube	5,390	bronchiectasis	2,698

The default indexing mechanism in *PadChest* is a list of records, encoded as a table, describing each image in the repository. Such data structure is computationally inefficient for a fast access and retrieval of the images. Particularly, each record in this table is a descriptor that characterizes a CXR image by using 36 features or attributes. Most of these features correspond to metadata extracted from the DICOM files which originally hosted the images. Two of these features are particularly relevant for classification tasks: "Labels" and "Localizations". The feature "Labels", is basically a list of anomalies or findings in an image, e.g. ["alveolar pattern", "bronchiectasis", "interstitial pattern"]. The "Localizations" attribute, represents a list of anatomical regions where the anomalies or findings have been localized, e.g.: ['loc basal', 'loc infrahilar', 'loc bronchi']. Hence the possibility of building training pairs (i, F) for supervised learning, where i represents an image and F is a set of findings present in the image i. These findings are extracted from the "Labels" feature. However, this list of records requires the entire table to be checked (i.e., more than 160K records) just to identify all the images associated to a given anomaly or finding. For example, for building a training dataset composed by images labeled as 'pseudo-nodule' or 'calcified granuloma' or 'tumor mass', each image descriptor in the table must be analyzed. Particularly, for each image descriptor, the "Labels" attribute has to be

parsed. Then, if one of the aforementioned anomalies or findings is found, the corresponding image is included in the dataset for training purposes. This data processing approach is computationally expensive and inefficient. Consequently, the computational cost turns larger when new images are added to the repository, as expected in a real scenario. Therefore, a list of records or descriptors does not represent an appropriate data structure for fast access and retrieval of medical images associated to a given set of labels or anomalies.

A much better solution for this problem can be obtained from the field of information retrieval, that is, by the creation of an inverted index[22,23], adapted to organize labeled medical images. This is a suitable data structure for organizing large volumes of data. In fact, this is what web search engines use for indexing and organizing data in Internet. More recently this data structure has been adapted for using in the biomedical field[24,25]. This is motivated by the fact such data structure supports a fast access and retrieval of the data being indexed, e.g., medical images in our application domain. Consequently, this solution can potentially be applied to all type of labeled medical image repositories (CT, MR or CXR image repositories).

Without loss of generalization, we introduce in this paper, as a use case, the adaptation, implementation and evaluation of an inverted index to organize the images of the *PadChest* repository. The general goal is to provide a tool that facilitates fast access and retrieval of *PadChest* images, plus the efficient creation of diverse image sets in terms of anomalies. This represents an important benefit for further training of CNN-based models on a large diversity of classification tasks.

Methods

To address the problem described in the previous section, we aimed at two main goals: 1) to construct an inverted index, adapted to chest X-ray images from the *PadChest* repository; and 2) to implement a computational tool capable of generating training datasets supported by the inverted index.

Design Tools

We used the unified modeling language (UML)[26] for creating static and core views of the inverted index and the software tool that generates diverse CXR image training datasets.
We additionally considered the design and implementation of a relational database for storing and retrieving the settings which makes reproducible the generation of training datasets of interest.

Implementation Tools

1. Implemented tools were fully based on open-source software systems.
2. We used Python as programming language and Spyder as integrated development environment (IDE) system.

3. Ubuntu 22.04 was the operating system for which the designed tools were coded and evaluated.
4. Reading and processing of the *PadChest* repository was performed through the libraries: Pandas and NumPy, respectively.
5. The creation and management of a database, which preserves settings for reproducing generated training datasets, was based on the popular SQL query language and the SQLite library for Python.

Evaluation Tools

The performance of the inverted index was assessed theoretically and experimentally as follows:

1. For the theoretical assessment of the adapted inverted index, we used algorithmic complexity techniques[27].
2. We measured the average retrieval time of labeled images when using the inverted index and the mechanism given by *PadChest*, i.e., the table of CXR image descriptors.
3. We evaluated the potential correlation existing between a dataset retrieval time and its number of images when using both, the inverted index and the *PadChest*'s table of descriptors.
4. This evaluation on the retrieval time was performed on different datasets of images, with different dataset sizes. We observed the effect of dataset size on the retrieval time when the implemented inverted index. The datasets retrieved were associated to the following labels or anomalies in *PadChest*: normal (50K images), EPOC (23K images), cardiomegaly (15K images), aortic elongation (11K images), pleural effusion (9K images), nodule (3K images) and emphysema (1K images).

Results and discusion

Structure and Advantages of the Inverted Index

In figure 1 we illustrate the general structure of the inverted index, designed and implemented for organizing the images of the *PadChest* repository. The inverted index is formally defined as follows:

$$I = \{(a, S') : a \in PadChest.Labels \wedge S' \subseteq S\}$$
$$S = \{(i, R) : i \in PadChest.ImageID \wedge R \subseteq PadChest.Localization\}$$

That is to say, the inverted index I, is a set of pairs (a, S'), where 'a' is one of the 174 anomalies or findings in *PadChest*. This means that anomalies are used as an indexing criteria. This is a totally different idea compared to the *PadChest's* table of descriptors, where the 'ImageID' field represents the indexing criteria. On the other hand, *S'* is a set of pairs (i, R) where 'i' is an image from *PadChest* and R is a set of anatomical regions, where the anomaly or finding is localized.

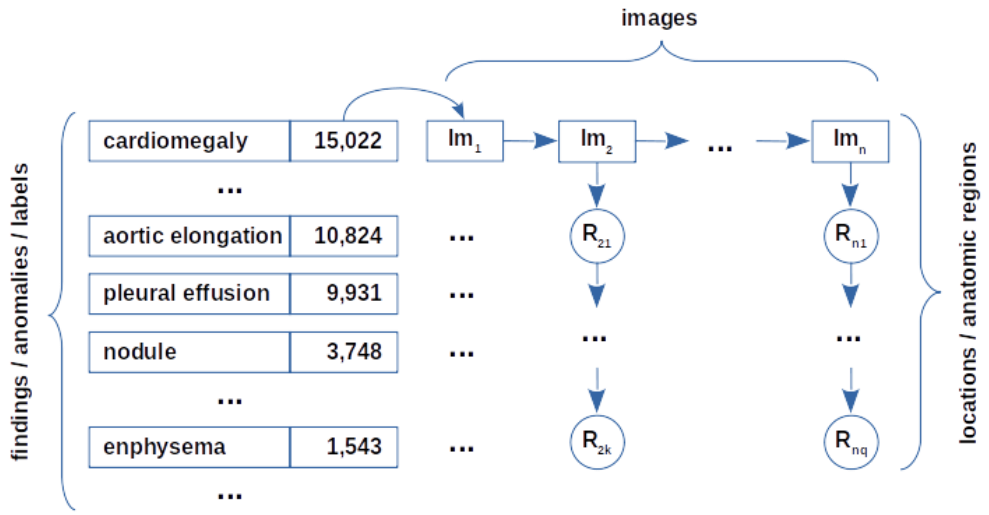

Figure 1. Inverted index of medical images, designed and implemented for organizing the images of the *PadChest* repository.

From the computer science perspective, the designed inverted index is basically a data structure supported, in our implementation, by another well-known data structure: a dictionary. The keys in this dictionary are the anomalies or findings. Each key has associated another data structure: a list of images. Each image in the list has associated a set of anatomy regions where a finding (represented by a dictionary key) is located.

This implementation approach of the inverted index is very advantageous with respect to the table of descriptors provided by the *PadChest* repository. Since a dictionary is the core data structure of the inverted index implemented, the access to images associated to a given anomaly is almost instantaneously. In fact, if we consider the "Big O" notation from the algorithm analysis, we can theoretically quantify to what extent one structure is more advantageous than the other. For example, let us consider n_a as the number of images associated to the anomaly or finding 'a'. Let N be the total number of images in *PadChest*. Hence, $n_a << N$. Let M be the total number of anomalies or findings. In case of using the descriptor table to access the images associated to a given anomaly or finding, the number of steps to be performed would always be O(N x M). In this case, each row of the table must be analyzed to find out if the anomaly or finding of interest is present in the image represented by that row. On the contrary, when using the inverted index, the number of steps to access the images associated to a finding or anomaly, would be O(1). That is, access to these images would almost be instantaneous. Moreover, in the inverted index, collecting all the images associated with finding 'a' would take a time proportional to $O(n_a)$. As a consequence, the number of steps for accessing to images associated to a given anomaly or finding is M times faster when using the inverted index than when using the *PadChest*'s table of descriptors.

Table 2 confirms the theoretical predictions, discussed above, regarding the access time to images when using the descriptor table or the inverted index. In the first case, as the number of images to be retrieved increases, the access time to these images exhibit high and almost invariable values, within the range of [40; 44] seconds and a

mean of 42.14 ± 1.68 seconds. As explained before, for the descriptor table all the records must be inspected, leading to an average access time of 42 seconds. This represents a moderate correlation between the number of images associated to a given anomaly and the access time to them, as confirmed by a 0.430 Pearson correlation.

Table 2. Access times to images, associated with terms from the *PadChest* repository, using the descriptor table and the implemented inverted index.

		Access time (s) using	
Finding / Anomaly	**Number of images**	**Table of Descriptors**	**Inverted Index**
normal	50,390	43.93	2.46
COPD	23,280	41.52	1.17
cardiomegaly	15,022	40.17	0.69
aortic elongation	11,780	43.38	0.51
pleural effusion	9,931	44.04	0.46
nodule	3,748	40.08	0.21
emphysema	1,543	41.83	0.07

However, when using the inverted index, we observed a strong and positive linear correlation (Pearson correlation = 0.999) between the number of images and the retrieval time of these images, as illustrated in Figure 2. The retrieval time to images is short (see the column: inverted index) and below 2.5 seconds, even for a large number of images. In general, when using the inverted index, the retrieval time is more than 10 times faster with respect to the table of descriptors. This experiment results support the algorithm complexity analysis aforementioned.

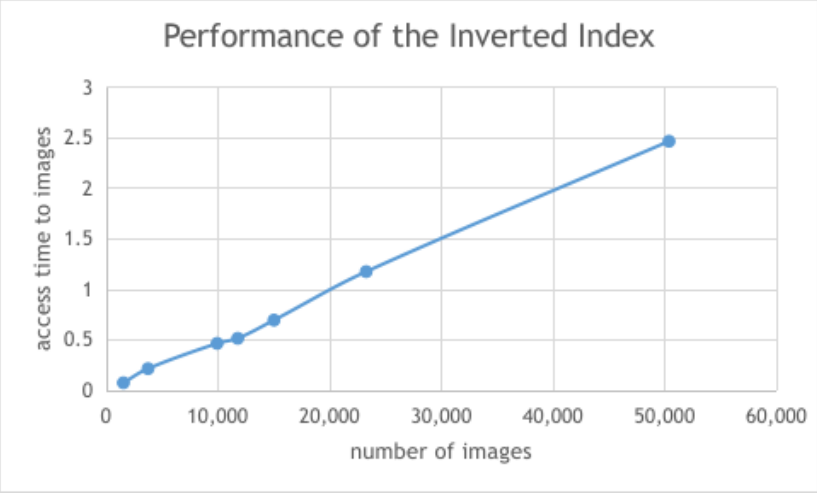

Figure 2. when using the inverted index, the access time to images associated with a finding in *PadChest* is positively linear.

Generation of Training Datasets

The task of generating image datasets for training CNN models was achieved by three object classes: “ExperimentData”, “InvertedIndex” and “CXRImage”. The relationship between these classes is illustrated as a class diagram in figure 3. The class “ExperimentData” is in charge of collecting images and creating the inverted index

through the aggregated class “InvertedIndex”. The “CXRImage” class encodes a CXR image (keeps image size, pixel depth, etc.) and also keeps track of the anatomic regions where certain anomaly or finding is localized within the image.

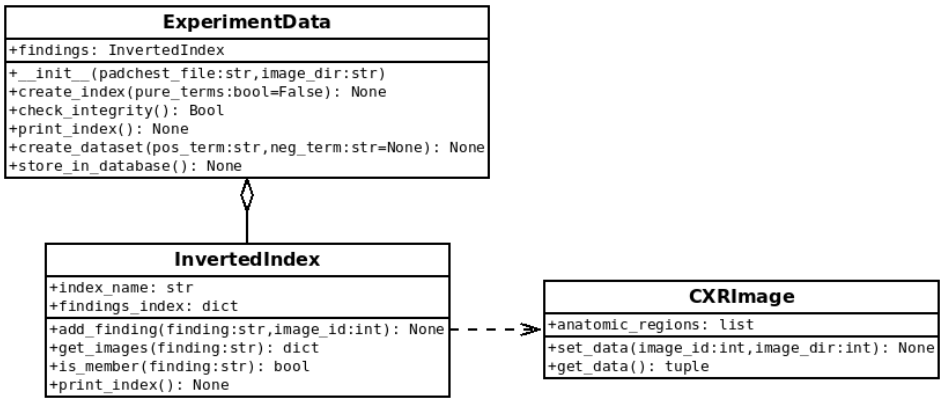

Figure 3. Object classes designed and implemented for creating an inverted index on the *PadChest* repository and for generating training datasets.

The responsibility for generating datasets for training relies on four methods or functions of the class “ExperimentData”, as follows:

1. **Function**: create_index(pure_terms: bool = False)

   This function creates an inverted index based on two attributes of the ExperimentData class: 1) the disk directory where the CXR images are stored; and 2) the *PadChest*’s descriptor table file. This data is captured by the ExperimentData’s constructor method (__init__(...)). This function can additionally generate “pure” training datasets. These are datasets of CXR images where "only one type" of anomaly or finding is found within the image. This option can be specified by setting the parameter *pure_terms = True*. Since “pure” training datasets have fewer images than datasets with more than one anomaly or finding, it is important to further compensate this deficit by using data augmentation techniques.

2. **Function**: check_integrity()

   This function checks the integrity of images to be indexed by the inverted index. Damaged images are excluded from the inverted index.

3. **Function**: create_dataset(term_list: list)

   Based on a list of radiology terms term_list = $[t_1, \cdots, t_k]$, given as an argument, this function generates a training dataset, where each image contains one or more of the radiology terms specified. To perform this task, the function relies on two aspects: 1) the inverted index created and 2) a predefined hierarchy of radiology terms (anomalies or findings) as shown in figure 4. This hierarchy encodes the relationship between anomalies, inspired on a systematic diagnosis approach (a map or a guide) used by radiologists for reading CXR images. Hence, given a radiology term t_j in $[t_1, \cdots, t_k]$, all the anomalies associated to t_j’s children leaf

nodes are retrieved as part of the training dataset associated to the term t_j. Figure 5 shows the browsing algorithm of from any sub-tree associated t_j .

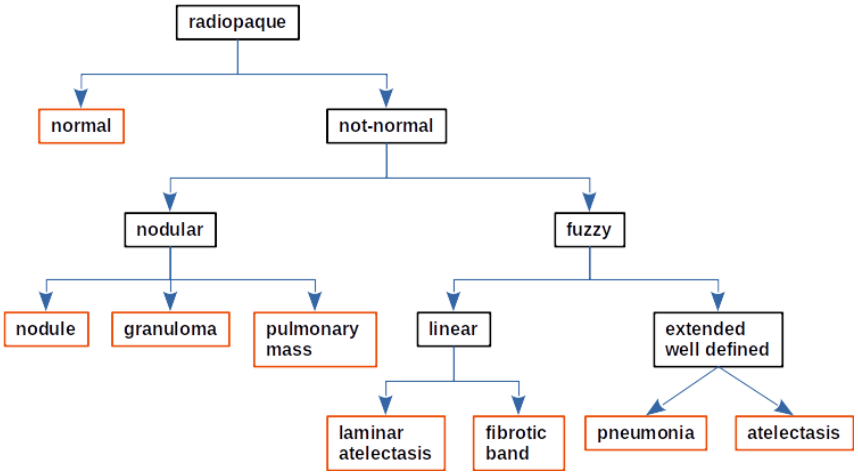

Figure 4. Hierarchy of radiology terms, corresponding to a systematic diagnosis approach for reading CXR images.

As an example, based on the hierarchy shown in figure 4, let term_list = [linear, nodular]. The leaf nodes (highlighted in red) associated with the term "linear" would be {laminar atelectasis, fibrotic band} (see the "linear" node and its children in figure 4). The images associated with "laminar atelectasis" and "fibrotic band" are retrieved through the created inverted index. Subsequently, the retrieved images are archived as a training dataset associated with the radiology term "linear". Similarly, for the radiology term 'nodular', its children leaf nodes are: {nodule, granuloma, pulmonary mass}. Then, the inverted index retrieves the images associated to these three anomalies or findings. The retrieved images are stored as a training dataset labeled as 'nodular'.

```
function TRAVERSE(node)
    if is_a_leaf(node) then
        images ← findings.GET_IMAGES(node)
        STORE_TO_DISK(images, node)
    else
        for all child in children(node) do
            TRAVERSE(child)
        end for
    end if
end function
```

Figure 5. Auxiliary function used for exploring the sub-tree associated to a given radiology term. The goal is to find the term's children leaf nodes (anomalies or findings).

The create_dataset(...) function makes also possible to consider an exclusion list in the generation of training sets. This way, images which are associated to certain anomalies are excluded, therefore adjusting the created dataset to specific user's needs. This facility is formally expressed as follows:

Let $[t_1, \cdots, t_k]$ be a list of radiology terms and let $[L_1, \cdots, L_k]$ be a list of image datasets, where each dataset L_j corresponds to the radiology term t_j. Let $[e_1, \cdots, e_p]$ be a list of excluded radiology terms, and let also be $[E_1, \cdots, E_p]$a list of

excluded image datasets, where E_i corresponds to the excluded radiology term e_i. Hence, each training set is defined as: $I'_j = I_j - E_i j = 1, \cdots, p$.

It is important to highlight that a dataset generated by the function create_dataset(...) it is always balanced in terms of number of images associated to each radiology term. Moreover, the function makes an appropriate distribution of images, organized as: training, validation and evaluation sets. To do so, it first takes the minimum cardinality between the generated image datasets, $c_{min} = min(\{|I_1|, \cdots, |I_k|\})$. Then the cardinality of each set is reduced to c_{min} and the images are selected randomly in each case. Subsequently, the images from each dataset are organized into three mutually exclusive subgroups. These are the training, validation and test groups, with a distribution of the images of (70%, 20% and 10%) respectively. The create_dataset(...) function ensures that these three subgroups are disjoint in terms of patients. That is, images of the same patient never appear in more than one of these subgroups of images. This avoids both, biased data and biased performance metrics.

4. **Function**: store_in_dataset(...)

This function preserves all the information necessary to retrieve each training dataset generated. For this purpose, a portable SQLite database was created. The database structure is illustrated in Figure 6. As illustrated, in the database structure one Experiment may produce one or more Datasets. The "Experiment" table captures the context where the experiment is run. Similarly, the "Dataset" table captures the attributes which describe a single training dataset generated (e.g., dataset name, date, dataset type). Moreover, one training dataset is composed by one or more findings (anomalies). These findings are captured by the "Findings" table. Finally, one finding or anomaly is connected with one or more CXR images, which in turn are recorded by the table "Image". The main utility of this database is to reproduce any training dataset generated, without having to store the images multiple times in storage devices.

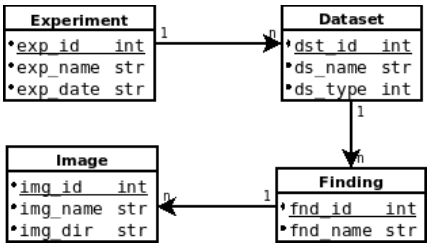

Table 3. Entity-relationship diagram illustrating the information considered in the portable SQLite database implemented for archiving the training datasets generated.

The implemented tool was found to be flexible and fast for the creation of diverse datasets in terms of anomalies. As a use case we first created an inverted index from the PadChest repository, using single-label images. This is achieved by calling the function: *create_index(true)*); then, we called the function: *create_dataset(['normal', 'not-normal'])* for creating dataset intended to train a model capable to discriminate between "normal" and "not normal" CXR images. Figure 6 shows the composition of both sets of images. The normal dataset was composed of "normal" labeled images.

Whereas, the “not-normal” dataset was composed of images labeled as child nodes labels in the subtree below the “not-normal” node, as shown in figure 4. This selection was automatically generated by the function: *create_dataset(...)*. Moreover, the function split the dataset into the subsets: train, validation and test; it also balanced both datasets: |normal|=2400 and |not-normal|=2359; For each dataset, the function kept the image distribution: 70%, 20% and 10%, for the “train”, “validation” and “test” sub-datasets respectively.

in folder: normal	test	train	valid
normal	240	1680	480

in folder: not-normal	test	train	valid
atelectasis	9	66	19
calcified granuloma	34	246	71
laminar atelectasis	53	387	109
granuloma	12	86	24
nodule	52	370	105
fibrotic band	15	103	30
pneumonia	45	322	90
pulmonary mass	11	78	22

Figure 6. Findings (labels) distribution for the automatically generated datasets of CXR images: "normal" and "not-normal".

Conclusions

The main contribution of this work is the creation of an inverted index adapted to the radiology application domain. By using this tool, it is possible to address the problem of organization and the efficient access and retrieval of medical images labeled with anomalies or findings of interest. This tool was adapted to manage images from the popular radiology image repository *PadChest*, with more than 160K chest X-ray images and more than 170 labels. We verified in practice that access and retrieval time of images associated to a given finding or anomaly is 10 times faster by using the inverted index than by using the *PadChest* image descriptor table.

We additionally produced a software tool, supported by an inverted index and a predefined hierarchy of radiology terms, which automatically generates a wide diversity of training datasets of CXR images. These datasets are automatically structured for supervised learning tasks through deep learning models. We implemented a relational database for preserving experiments data involved in the generation of training datasets. A use case was presented as a practical example of using the implemented inverted index.

References

1. Thian YL, Ng DW, Hallinan JTPD, Jagmohan P, Sia SY, Mohamed JSA, et al. Effect of Training Data Volume on Performance of Convolutional Neural Network Pneumothorax Classifiers. Journal of Digital Imaging [Internet]. 2022;35(4):881-892. Disponible en: https://doi.org/10.1007/s10278-022-00594-y
2. Willemink MJ, Koszek WA, Hardell C, Wu J, Fleischmann D, Harvey H, et al. Preparing Medical Imaging Data for Machine Learning. Radiology. 2020;295(1):4–15.
3. Pourvaziri A, Narayan AK, Tso D, Baliyan V, Glover M, Bizzo BC, et al. Imaging Information Overload: Quantifying the Burden of Interpretive and Non-Interpretive Tasks for Computed Tomography Angiography for Aortic Pathologies in Emergency Radiology. Current Problems in Diagnostic Radiology [Internet]. 2022. ;51(4):546-551 Disponible en: https://www.sciencedirect.com/science/article/pii/S036301882200007X
4. Treviño M, Birdsong G, Carrigan A, Choyke P, Drew T, Eckstein M, et al. Advancing Research on Medical Image Perception by Strengthening Multidisciplinary Collaboration. JNCI Cancer Spectrum [Internet]. 2021;6(1). Disponible en: https://doi.org/10.1093/jncics/pkab099
5. Jeganathan S. The Growing Problem of Radiologist Shortages: Australia and New Zealand's Perspective. Korean journal of radiology. Korea (South). 2023.:1043–5.
6. Konstantinidis K. The shortage of radiographers: A global crisis in healthcare. Journal of Medical Imaging and Radiation Sciences. 2024;55(4):101333.
7. Segal JP, Hansen R. Medical images, social media and consent. Nature Reviews Gastroenterology & Hepatology. 2021;18(8):517–8.
8. Abouelmehdi K, Beni-Hessane A, Khaloufi H. Big healthcare data: preserving security and privacy. Journal of Big Data. 2018;5(1):1–5.
9. Zhao X, Wang L, Zhang Y, Han X, Deveci M, Parmar M. A review of convolutional neural networks in computer vision. Artificial Intelligence Review. 2024;57(4):99.
10. Li Z, Liu W, Yang S, Peng J, Zhou W. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. IEEE Transactions on Neural Networks and Learning Systems. 2022;33(12):6999–7019.
11. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV). 2015;115(3):211–52.
12. Abrol A, Fu Z, Salman M, Silva R, Du Y, Plis S, et al. Deep learning encodes robust discriminative neuroimaging representations to outperform standard machine learning. Nature Communications. 2021;12(1):353.
13. Lindsay GW. Convolutional Neural Networks as a Model of the Visual System: Past, Present, and Future. Journal of cognitive neuroscience. 2021;33:2017–31.
14. Kennedy DN, Haselgrove C, Riehl J, Preuss N, Buccigrossi R. The NITRC image repository. NeuroImage. 2016;124:1069–73.
15. Prior FW, Clark K, Commean P, Freymann J, Jaffe C, Kirby J, et al. TCIA: An information resource to enable open science. En: 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). 2013: 1282–5.

16. Bustos A, Pertusa A, Salinas JM, de la Iglesia-Vayá M. PadChest: A large chest x-ray image dataset with multi-label annotated reports. Medical Image Analysis. 2020;66:101797.
17. Castro DC, Bustos A, Bannur S, Hyland SL, Bouzid K, Wetscherek MT, et al. PadChest-GR: A Bilingual Chest X-ray Dataset for Grounded Radiology Report Generation [Internet]. Nueva York: arxiv; 2024. Disponible en: https://arxiv.org/abs/2411.05085
18. Asraf A, Islam Z. COVID19, Pneumonia and Normal Chest X-ray PA Dataset. Ireland: Mendeley Data; 2021.
19. Johnson AEW, Pollard TJ, Berkowitz SJ, Greenbaum NR, Lungren MP, Deng C ying, et al. MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports. Scientific Data. 2019;6(1):317.
20. Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM. ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases [Internet]. Nueva York: arxiv ;2017. Disponible en: http://arxiv.org/abs/1705.02315
21. Irvin J, Rajpurkar P, Ko M, Yu Y, Ciurea-Ilcus S, Chute C, et al. CheXpert: a large chest radiograph dataset with uncertainty labels and expert comparison. En: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence [Internet]. Honolulu, Hawaii, USA: AAAI Press; 2019. Disponible en: https://doi.org/10.1609/aaai.v33i01.3301590
22. Baeza-Yates R, Ribeiro-Neto B. Modern Information Retrieval: The Concepts and Technology behind Search. 2 ed. USA: Addison-Wesley Publishing Company; 2011.
23. Manning CD, Raghavan P, Schütze H. Introduction to Information Retrieval [Internet]. England: Cambridge University Press; 2008. Disponible en: http://nlp.stanford.edu/IR-book/information-retrieval-book.html
24. Bittrich S, Burley SK, Rose AS. Real-time structural motif searching in proteins using an inverted index strategy. PLoS Comput Biol. 2020;16(12):e1008502.
25. Nasr R, Vernica R, Li C, Baldi P. Speeding up chemical searches using the inverted index: the convergence of chemoinformatics and text search methods. J Chem Inf Model. 2012;52(4):891–900.
26. Hoffman W. Unified Modeling Language [Internet]. Milford: UML; 2022. Disponible en: http://uml.org
27. Yousaf QH, Shah MA, Naseem R, Wakil K, Ullah G. An effective approach to analyze algorithms with linear O(n) worst-case asymptotic complexity. International Journal of Advanced Computer Science and Applications. 2019;10:337–42.

Conflict of interest statement

The authors have no conflicts of interest to declare. All co-authors have seen and agree with the contents of the manuscript. There is no financial interest to report. We certify that the submission is original work and is not under review at any other publication.

Authorship contribution statement

Conceptualization	Henry Blanco Lores
Data curation	Henry Blanco Lores
Formal analysis	Henry Blanco Lores
Acquisition of funds	Jef Vandemeulebroucke
Research	Henry Blanco Lores
Methodology	Jef Vandemeulebroucke
Project management	Jef Vandemeulebroucke
Resources	Jef Vandemeulebroucke
Software	Henry Blanco Lores
Supervision	Jef Vandemeulebroucke
Validation	Henry Blanco Lores
Visualization	Henry Blanco Lores
Writing – original draft	Henry Blanco Lores
Writing – review and editing	Jef Vandemeulebroucke