



جامعة أبو بكر بلقايد - تلمسان

Université Abou Bakr Belkaïd de Tlemcen

Faculté de Technologie

Département de Génie Biomédical

MEMOIRE DE PROJET DE FIN D'ETUDES

pour l'obtention du Diplôme de

MASTER en GENIE BIOMEDICAL

Spécialité : Informatique Biomédicale

présenté par : Melle Benyelles Fatima Zohra & Melle Sekkal Amel

**Medical Image Retrieval using
Stacked Autoencoders : COVID-19 Application**

Soutenu le 31 Aout 2020 devant le Jury

M.	CHIKH Mohammed Amine	<i>Prof</i>	Université de Tlemcen	Président
Mme	SETTOUTI Nesma	<i>MCA</i>	Université de Tlemcen	Encadreur
M.	BECHAR Mohammed El Amine	<i>MCB</i>	Université de Tlemcen	Examineur

Année universitaire 2019-2020

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ ABOU BEKR BELKAID
FACULTÉ DE TECHNOLOGIE
DÉPARTEMENT DE GÉNIE BIOMÉDICAL

MÉMOIRE DE FIN D'ÉTUDES

pour obtenir le grade de
MASTER EN GÉNIE BIOMÉDICAL
Spécialité : **Informatique Biomédicale**
présenté et soutenu publiquement
par

Melle BENYELLES Fatima Zohra
&
Melle SEKKAL Amel

le 31 août 2020

Titre:

Medical Image Retrieval using Stacked Autoencoders : COVID-19 Application

Jury

Président du jury. Pr. CHIKH Mohammed Amine, UABB Tlemcen
Examineur. Dr. BECHAR Mohammed El Amine, MCB UABB Tlemcen

Directrice de mémoire. Dr. SETTOUTI Nesma, MCA UABB Tlemcen

Année Universitaire 2019-2020.

*À l'exemple de l'amour et du dévouement ...
À celui qui n'a épargné ni temps ni effort pour m'aider ...
À mon modèle et à mon idole dans la vie ...
Mon père mon amour.*

*À l'exemple de l'inspiration et du don ...
À la source de l'amour et de la joie de vivre ...
Ma Chère maman.*

*Aux gens qui m'aiment ...
Bassidi, Mima, MA et BA.*

*À la belle et folle ...
Nour.*

*À mon petit vilain...
Hamid.*

*À mon père spirituel...
Cheikh Al-Naim.*

*À celle qui est devenue ma sœur au lieu de mon professeur ... À mon modèle
Nesma.*

*A tous ceux qui m'aiment sincèrement ...
Je vous dédie cet humble travail avec fierté et vous souhaite le meilleur dans la vie.*

Fatima Zohra

Tous les mots ne sauraient exprimer ma gratitude, mon amour, mon respect, ma reconnaissance, et c'est tout simplement que je dédie ce travail à :

À ma tendre Mère : tu représentes pour moi la source de tendresse et l'exemple de dévouement qui n'a pas cessé de m'encourager. Tu as fait plus qu'une mère puisse faire pour que ses enfants suivent le bon chemin dans leur vie et leurs études.

À mon magnifique père : aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours pour vous. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien-être. Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation le long de ces années.

À mon adorable grand-mère : depuis mon jeune âge, tu as participé à mon éducation avec ta tendresse et ta bonté, tu as toujours été fière de moi et de ce que je suis devenue. À toi ma meilleure personne dans ce monde je te dédie ce travail.

À mes feux grands-parents paix à leurs âmes. Merci pour vos Douaas qui m'ont accompagnés et béni toute ma vie.

À mes chers Frères Mohammed et Mustapha et ma Sœur Zineb : je voudrais par ce travail vous exprimer mon amour éternel, mes respects et ma fierté pour vous tous en espérant d'être le bon exemple à suivre dans cette vie.

À mes chères amies et collègues, avec qui j'ai traversé cinq années d'études pleines de souvenirs et de bonne entente, a nos connaissances que nous avons construit qui nous ont apporté de l'aide et de l'encouragement.

À mes collègues et camarades de la promotion Génie Biomédical 2019-2020.

À toute personne qui m'a aidée, soutenue et crue en moi.

Amel

Acknowledgements

In the first place, we thank God the Almighty for giving us the strength to do this work and for blessing us in our quest for knowledge in this lower world.

We express our deep thanks to our thesis supervisor, **Dr. SETTOUTI Nesma** for the competent help she gave us, for her patience and her encouragement. Her critical eye was invaluable to us in structuring the work and in improving the quality of the different sections. We want you to know that we were very lucky to have you as a mentor and our thanks will never be enough for all that you have given us as well as your moral support during the delicate situation which we are living.

We also thank **Prof. CHIKH mohammed Amine** and **Dr. BECHAR Mohammed El Amine** our members of the jury for their interest in this work and for taking the time to evaluate this thesis.

We would also like to thank our dear professors and teachers who have guided us throughout our university course in biomedical engineering training, and who have been a source of knowledge and inspiration.

Without forgetting the *HD Consulting* firm, which opened its doors to us to allow us to work in a professional environment. A special thank's to **Mr. BOUIDA Djelloul** the manager and our dear friends **Ms. BOUAYADI Nour** and **Mr. KAZI AOUAL Samad**.

Finally, our gratitude and thanks go to our dear families, our parents and our brothers and sisters.

Résumé

La COVID-19 est une maladie infectieuse récemment découverte causée par le corona virus connu par la provocation des infections respiratoires chez l'humain. Cette pandémie se propage rapidement dans le monde, causant de multiples dommages dans différents secteurs. Dans ce projet de fin d'études, nous nous intéressons à la reconnaissance de cette maladie à l'aide d'images médicales. Pour ce faire, nous présentons une application dédiée aux épidémiologistes pour l'investigation du patient-0 contaminé et établir le chemin de propagation dans les différentes zones du pays. A ces fins, nous proposons un système de recherche d'image médicale par son contenu CBMIR (Content Based Medical Image Retrieval) se basant sur les réseaux d'encodeurs en cascades qui sont dédié à la recherche d'images COVID Chest X-Ray. Notre modèle se base sur les mesures de similarité apprises au travers d'une base d'images regroupant différentes pathologies comme le SRAS et d'autres espèces virales ou bactériennes de maladies de pneumonie.

Mots clés

Recherche d'image par son contenu, Auto-encodeurs en cascades, COVID-19, investigation, reconnaissance, images à rayons X médicales, extractions des caractéristiques

Abstract

COVID-19 is a recently discovered infectious disease caused by the coronavirus known to cause respiratory infections in humans. This pandemic is spreading rapidly around the world, causing multiple damages in different areas. In this graduation project, we are interested in the recognition of this disease using medical images. For this purpose, we present an application dedicated to epidemiologists for the investigation of the Patient 0 infected and establish the propagation path in different areas of the country. A Content Based Medical Image Retrieval (CBMIR) system based on stacked-encoder networks is proposed, our model is dedicated to search for target COVID Chest X-Ray images using similarity measurements learned through an image database of different pathologies as SARS and other viral or bacterial species of pneumonia diseases.

Keywords

Content based image retrieval, Stacked autoencoders, COVID-19, investigation, recognition, X-rays medical images, Features extraction.

Contents

Acknowledgements	ii
Résumé	iii
Abstract	iv
Contents	v
List of Figures	vii
List of Tables	viii
Glossary of terms	ix
Introduction	1
1 Content-based Medical Image Retrieval	3
1 Content-based Image Retrieval (CBIR)	4
2 Basics of Content-based Image Retrieval (CBIR)	4
2.1 Feature Extraction	5
2.2 Similarity Comparison	5
2.3 Indexing Schema	6
2.4 Interactive Query Interface	6
3 Image Features in CBIR	6
3.1 Color	6
3.2 Texture	6
3.3 Shape	8
3.4 Spatial relationships	8
4 Content-based Medical Image Retrieval (CBMIR)	8
5 Related Works	9
6 Synthesis	12
7 Conclusion	12
2 Auto-encoders networks for Content-Based Medical Image Retrieval	13
1 Auto-encoder network	13
2 Architecture of a traditional Auto-encoder	15
3 Auto-encoders structure types	17
3.1 Denoising auto-encoder	18
3.2 Sparse auto-encoders	20
3.3 Variational auto-encoders	22
3.4 Stacked auto-encoders	23
4 Related works of Auto-encoders in medical image retrieval	24
5 Synthesis	26
6 Conclusion	26

3	Content-Based Medical Image Retrieval for COVID-19 investigation	27
1	Proposed framework	29
2	Experiments and results	30
2.1	COVID-19 Images dataset	30
2.2	Process of the proposed approach	32
2.3	Results and discussion	33
3	Conclusion	36
	Conclusion	37
	Bibliography	39

List of Figures

1.1	Component of the CBIR system	5
1.2	The different component and type of CBIR system	8
2.1	Auto-encoder network	14
2.2	The auto-encoder system.	15
2.3	Result of the simple AE	17
2.4	Denoising auto-encoders system	18
2.5	Some examples from the noisy images	19
2.6	Result of the denoising AE	20
2.7	Sparse auto-encoder system	21
2.8	Result of the sparse AE	21
2.9	Variational auto-encoder system	22
2.10	Stacked auto-encoder Architecture.	23
2.11	Result of the stacked AE	24
3.1	Scheme of the proposed framework	30
3.2	Image samples of the dataset	31
3.3	Image samples of the dataset	32
3.4	Summary of the encoder and the decoder architecture from our model.	34
3.5	Summary of the SAE architecture.	34
3.6	Some retrieval examples with different cases using K-nn algorithm.	35

List of Tables

1.1	A comparison between Domain-specific semantic features general visual features	7
1.2	Recent works about developing the feature extraction phase	10
2.1	Recent works about auto-encoders in the image analysis domain. .	25
3.1	Count of type of pneumonia with species and genus.	31

Glossary of terms

AE: Auto-encoders
AI: Artificial intelligence
ANN: Artificial neural networks
ARBC: Autoencoders on Radon barcodes
CAD: Computer-Aided Diagnostic System
CBIR: Content-based image retrieval
CBMIR: Content-based medical image retrieval
CT: Computed tomography
DAE: Denoising auto-encoder
DL: Deep learning
GLCM: Gray Level Co-occurrence Matrix
HFS: Hybrid feature space
HSV: Hue, saturation, value
KL: Kullback-Leibler divergence
KNN: k-nearest neighbors
LSTM: Long-Short term memory
MAP: mean average precision
ML: Machine learning
mse: mean squared error
PCA: principal component analysis
PCR: polymerase chain reaction
PMCVNN: Predictor model based Cross validation neural network
RBC: Radon barcodes
RBM: Restricted Boltzmann Machines
ReLU: Rectified Linear Unit
RF: relevance feedback
RGB: Red, Green and Blue
ROI: Region of interest
SAE: Sparse auto-encoder
SAEs: Stacked auto-encoders
SM: Similarity Matching
SOM: self-organizing map
SVM: support vector machine
TL: Transfer Learning
VAEs: Variational auto-encoders
YOLO: You Only Look Once

Introduction

Medical images are considered as an important source of dependable information to serve medicine as it is the origin of important information for the diagnosis, disease management, and controlling treatment responses of patients. Yet with the growth of multimedia technology, many image databases are available worldwide, which brings us to the notion of BIG DATA.

To use these databases, a retrieval approach is needed that is dynamic, strong, and robust. Content-based image retrieval (CBIR) presents the best solution that can efficiently retrieve the images, on this wise it can be possible to access the visual content from a large database.

Such systems are more specially required in these times, where we face the COVID-19 pandemic that spread very quickly in our world and make a big damage to the human lives. Meanwhile, Artificial intelligent (AI) applications are spotted in a different stage of this pandemic; the most useful application in this task is the detection of COVID-19 cases. Either to be a help in the diagnosis or to compare with other cases, or even to guide the medical students.

In this project, we proposed a Content Based Image Retrieval (CBIR) system to retrieve (which lead to detect) the similar COVID-19 images from an existing database using the "Stacked Auto Encoders" (SAEs), one of the most powerful methods in Deep Learning (DL) which encode images into a binary vector as a smaller, compact representation for a faster retrieve.

CBIR work in a way where it identifies similar medical images with previous cases. As COVID -19 applications, this would help the epidemiologist for the investigation of the Patient 0 infected and establish the propagation path in different areas of the country.

The main idea behind our system, is the extraction of characteristics from X-Ray images from the latent space of the SAE, and retrieve the similar images from various cases using a K-nn classifier.

This Master thesis is structured as a fellow:

- Chapter 1: present the basis of content-based image retrieval (CBIR) and its general methods with an overview of current content-based medical image retrieval (CBMIR) techniques.

- Chapter2: introduce the principle of Auto-encoders with citing all its different properties, architectures, and different types using a coding tutorial part.
- Chapter3: present our proposed model for the recognition of COVID-19 disease from other cases.
- A conclusion that summarizes our work with a future perspective.

Chapter 1

Content-based Medical Image Retrieval

Introduction

With the development of multimedia technology, large number of image databases are available worldwide, whereby an increasing usage of image databases becomes possible. To leverage this volume of data, an effective and robust retrieval and search approach is required [1,2].

The content-based image retrieval (CBIR) received a lot of attention in areas such as: medical research and diagnosis [3], education [4], geographic information systems [5], engineering design, preventing crimes [6], digital libraries, military sector and many more.

Retrieve a targeted image, from a large collection is one of the main research topics; Initial CBIR systems were developed to search databases based on image color, texture, and shape properties, and those systems are used in order to automatically index, search, retrieve, and browse image databases, apparently it helps to reduce time, money, and relieves humans instead of what they used to do with the traditional methods.

The CBIR systems are very useful in the medical domain, indeed, CBMIR helps doctors in retrieving similar images and case histories for understanding the specific disease of the patient, it also helps the radiologists for preparing the report of diagnosis more accurate. Besides from the diagnostics, CBMIR also assists in teaching and research.

In this chapter, we present a brief review about the meaning of CBIR and its general methods with an overview of current content-based medical image retrieval (CBMIR) techniques. We discuss also, some related works proposed in previous researches.

1 Content-based Image Retrieval (CBIR)

The purpose of the image retrieval system is to find and retrieve the required and targeted images from a large collection of image sets. As technology advances, traditional methods of retrieval are being replaced by computer-assisted methods, reducing time, money and manpower required for manual searching.

However, a set of images contains images of different colors, shapes, orientations, and sizes [7]. The surface structure of the object in one image may be different from the object in another image. These factors complicate the image retrieval process.

The traditional image retrieval process involves describing each image using a text annotation and retrieving images by searching the keywords. Due to the rapidly increasing number of images and the variety of image content, this process is laborious and ambiguous. Nevertheless, such textual approach re-establishes significant boundaries. Manual commenting is subjective, time-consuming, and expensive [8], also the content of very large image databases surpasses the manual indexing capabilities of human experts.

For this reason, CBIR has received a lot of attention. Since the early 1990s [2], the CBIR has been an active area of multimedia research, and over the past 10 years, various researches were working on retrieving images from large repositories by analyzing the content of images.

Therefore, to remove the restrictions on the text-based image retrieval a CBIR system is proposed to allow the user to extract a request-based image from a database containing many images. The most important problem when designing a content-based image retrieval system is to select from data the best image features that describe image content [9]. Besides, CBIR remains a very difficult task, in literature, some works focus on how to represent an image, i.e. how to extract the visual features of an image, some other works focus on understanding the image, such as extract objects from the image and describing the relationship between the objects.

2 Basics of Content-based Image Retrieval (CBIR)

A typical CBIR system is depicted in Figure 1.1. In the first step, the system extracts the visual content of each image in the database, and then use an off-line feature extraction process to calculate a set of distinctive features (multi-feature vectors) to store it in a feature meta-data repository.

As a second step, the user enters a query image to the system, and via an on-line feature extraction process the image is converted into an internal feature vector. And for the third step the system calculates the similarity between the feature vector of the user's query image and the features meta-data items and rank it during similarity comparison stage. In the fourth step an indexing schema is ap-

plied to support fast retrieval and to make the system scalable to large databases. The last step allows the user to give some positive or negative feedback to the system based on the retrieval results via an interactive relevance feedback, this step is repeated until the satisfaction of the user. Those feedbacks can be a help to improve the system performance. Among the various components mentioned above feature extraction, similarity comparison, the indexing scheme, and interactive query interface are four key issues in any CBIR system, in the rest of this section we detail each part.

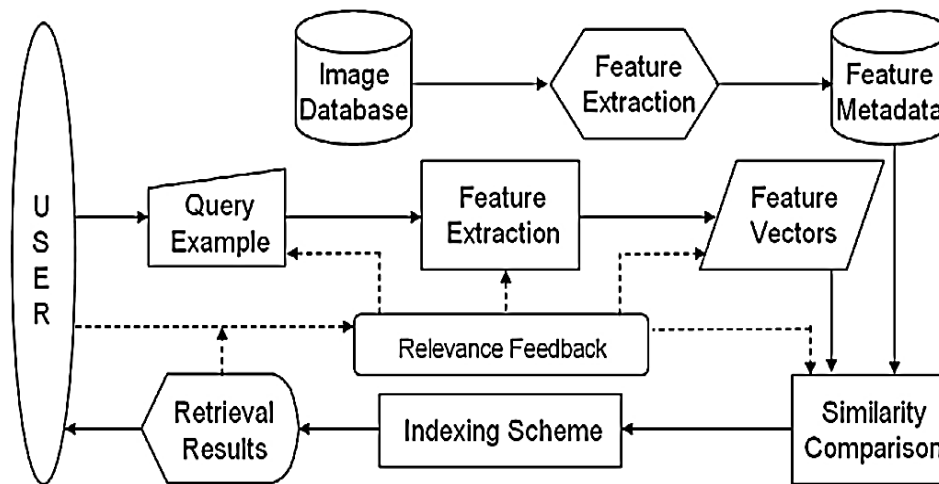


Figure 1.1: Component of the CBIR system

2.1 Feature Extraction

This is the basis component of the CBIR system. In a broader sense, features can include text-based features (key words, annotations) and visual features (color, texture, shape, spatial relationships). Image representation should consider which features are most useful for representing image content, and what methods can effectively encode image attributes [10]. Because of the subjectivity perception, a particular representation does not have a good image, but it does use a number of representations to describe features from different perspectives.

2.2 Similarity Comparison

To measure similarity, the overall approach involves presenting the features of an image into multidimensional vectors. The choice of dimensions directly influences the performance of the CBIR system [10]. The similarity function defines the pair of feature vectors and the chosen positive real number as a representation of the visual similarity of the two images. As a result, there's not only a single image, but a list of images ranked by their similarities with the query image. Different similarity comparison methods for image retrieval have been developed based on empirical estimates of the distribution of the features.

2.3 Indexing Schema

Using a small number of images in a database, a continuous linear search can provide reasonable performance. However, large collections of images require support for similarity query, indexing and can help prevent permanent crawling [11,12]. Directory structures filter all unrelated images by checking the image attributes of the user request and only stores the associated images without analyzing the entire database. Retrieved images are ranked in order of similarity to a query.

2.4 Interactive Query Interface

Or known also as "Relevance Feedback", the user can formulate and change queries via the interactive query interface. In any CBIR system, it is important that the user can express his research needs precisely and easily. The ability to refine on-line searches based on user response, referred to as relevance feedback, is particularly useful to improve the effectiveness of CBIR systems [13,14]. The main idea of relevance feedback is to use positive and negative user examples to improve system performance.

3 Image Features in CBIR

Obviously, the most important basis of the CBIR system is feature extraction, which uses quantitative image properties that automatically calculate the image content. Essentially, image features can be classified into general visual and specific semantic features domain.

The specific semantic features domain used to be the first method utilized in the image retrieval system, by its textual annotation. However, due to several problems, the general visual features appeared in the area. Table 1.1 describe a comparison between those two types of features.

We briefly present the general visual features that can be used in most CBIR applications.

3.1 Color

Its unchanged relation to image scaling, translation and rotation, and its three-color values (such as RGB or HSV) give it a higher discrimination ability than the gray scale [10,15,16].

3.2 Texture

Texture measures try to catch the characteristics of the image including the changes in certain directions and the scale of the changes [17]. This is most useful for images with homogeneous texture.

TYPE OF FEATURE	SPECIFIC SEMANTIC DOMAIN	GENERAL VISUAL
YEAR	From 1970	From 1980
REFERS TO	The meaning of an image	The make-up of an image
DESCRIBING BY	High-level	Low-level
INDEXING	key words, annotations	color, texture, shape, size, intensity, location, spatial relationships
ADVANTAGE	-Humans tend to use high-level features, such as keywords, text descriptors	-More efficiently in the medical domain (based on the visual representation). -Many visual features in images, such as irregular shapes and jumbled textures, are extremely difficult to describe in text. -Features can be either extracted from the entire image or from regions.
DISADVANTAGE	-The text is cumbersome, -Highly error-prone, -Unable to sufficiently describe the rich visual properties or features inside the images content, -Manual annotation is subjective, -Time-consuming, -Expensive, -Text-based approaches perform retrieval only at a conceptual level, In the medical domain, the majority of acquired medical images are currently stored with a limited text-based description of their content,	-Computer vision techniques are mostly low-level features, -Low-level contents often fail to describe the high-level semantic concepts in user's mind.

Table 1.1: A comparison between Domain-specific semantic features general visual features

3.3 Shape

Can be used to identify an object or region as an important geometric shape. As a result, shape features elements are usually depicted after the image is segmented into objects or regions. Due to the difficulty of fully automated image segmentation and the ability to project a given 3D object into 2D images, shape based CBIR is considered one of the most common tasks [10, 18]. In general, the techniques that represent shape are divided into two broad categories: boundary-based and region-based approaches.

3.4 Spatial relationships

Multiple objects or areas of an image, usually capture the most relevant and general information in the image content. They are very useful for retrieving and searching for images. Spatial relationships can be divided into two classes: directional (or orientation) and topological relationships [10].

- Directional relationships capture the “left”, “above”, and “front” sides of each other relative objects.
- Topological relationships describe the neighborhood and distribution of objects such as “disjoint”, “adjacency”, and “overlapping”.

Figure 1.2 describe the CBIR system with his components and the different type of features used in CBIR system.

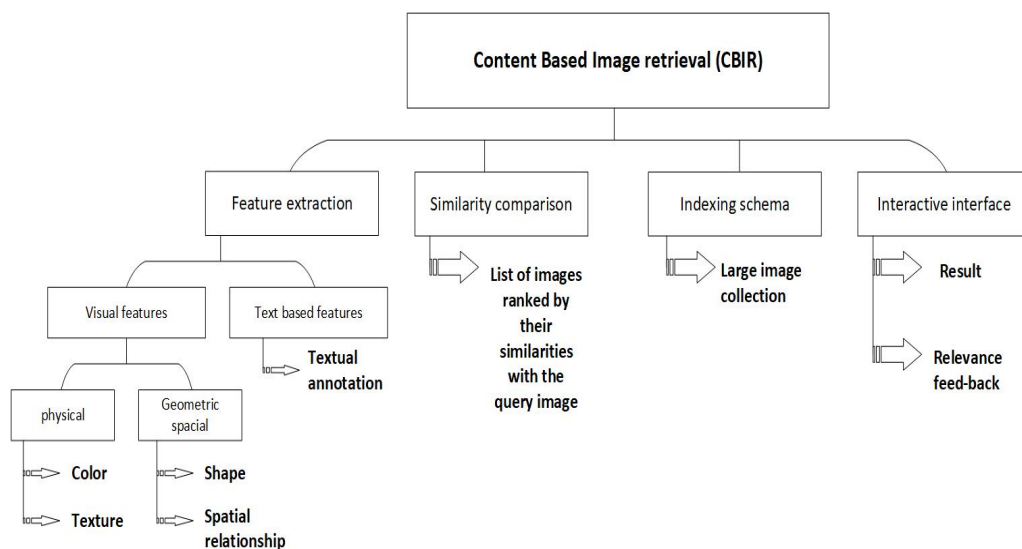


Figure 1.2: The different component and type of CBIR system

4 Content-based Medical Image Retrieval (CBMIR)

CBIR enables the radiologist to identify similar medical images by recalling a previous diagnosis. Although many algorithms have been implemented to retrieve

medical image content, this process remains a challenge because of the feature extraction, where most are extracted at a lower level. Interpreting medical images is certainly a difficult task that requires detailed knowledge. To help radiologists interpret medical images, researchers have developed support systems such as the Computer-Aided Diagnostic System (CAD) and the Content Based Medical Image Retrieval system (CBIMR) [19]. CAD can help the radiologist's diagnosis task, by bringing them a second opinion. On the other hand, Content-Based Image Recovery (CBMIR) uses visual content to help users navigate, search and retrieve similar medical images from a database.

CBMIR can help diagnosis in: Observing an abnormality in a diagnostic image, a physician can query a database of known cases to obtain images (and accompanying textual information) of areas with similar features to those observed in the image of interest. Knowing the pathological features that match the features of the selected area will give the clinician more confidence in the diagnosis and extend the differential diagnosis to include previously unrelated pathological features.

In addition, CBMIR could be used to represent cases that are not only similar in diagnosis, but also similar in appearance and cases with visual similarity but different diagnoses [8]. Therefore, it can be a useful teaching tool for medical students, residents, and researchers to browse and search large collections of disease-related images based on their visual characteristics.

Various methods for biomedical image indexing and retrieval is presented over the past years, like descriptors, bar-codes, machine learning (includes fuzzy and neural network, support vector machine, unified medical language system), and many more techniques [20]. The selection of method is very important in image retrieval as it directly affects the performance of the retrieval system.

The success of CBMIR will open many perspectives on medical services and research, such as patient surveillance, differential diagnosis, non-invasive surgical planning, clinical practice, and outcome research.

5 Related Works

The CBMIR systems share a common architecture on preprocessing phase based on extraction and characterization of relevant feature vectors, and an on-line processing that relies on retrieval and images classification.

In Table 1.2, we present recent works that focused on developing the feature extraction phase through the manipulation and selection of the visual features, as well as relevant works on developing new methods of similarity comparison measurements.

AUTHOR	TITLE	FEATURES USED	SIMILARITY MEASUREMENT	DATABASE
Mirasadi and Foruzan 2019 [21]	<i>CBMIR of CT images of liver lesions using manifold learning</i>	Statistical texture using Gray Level Co-occurrence Matrix (GLCM)	The correlation graph distance as a nonlinear approach measuring the similarity of pathological images	CT liver images data
Shamna et al. 2018 [22]	<i>CBMIR by spatial matching of visual words</i>	Visual words (low level features descriptors)	The Skip Similarity Index	Multimodal dataset (MR, CT, PT, PET, OPT, X-ray etc.) IRMA 2009 dataset Kvasir dataset
Mishra and Panda 2018 [7]	<i>CBMIR using self-organizing map on texture features</i>	Local texture features	SOM (self-organizing map) algorithm	TCIA collections dataset of brain magnetic resonance images
Kumar and Singh 2018 [23]	<i>CBMIR System to diagnose Hepatobiliary Images</i>	Hu-moments and GLCM methods on visual features	Euclidean distance Method	hepatobiliary images
Mustapha et al. 2015 [24]	<i>Design and development of CBMIR system for spine vertebrae irregularity</i>	Shape boundaries features	Integration between PMCVNN (Predictor model based Cross validation neural network) and SM (Similarity Matching)	NHANES II database
Ergen and Baykara 2014 [25]	<i>Texture based feature extraction methods for CBMIR systems</i>	texture descriptors using GLCM matrix, GLRL matrix and Gabor wavelet	Euclidean distance	brain, lung, sinus, and bone medical images database

Table 1.2: Recent works about developing the feature extraction phase

Additionally, some CBIR works applied in different areas outside the medical field came with new inspiring methods, as Kommineni et al. 2015 [9] who present a method for efficient classification and database classification with SVM classifier using color feature. Also, Mistry et al. 2017 [26] proposed a CBIR system using hybrid features collection (color auto-correlogram, color moments, HSV histogram, frequency domain and other color descriptor features) and using various distance metrics for similarity measurements. While M.K. Alsmadi 2017 [2] take a new approach to extract more significant features such as color signature with the shape and color texture features from the given query image and saving it features repository and he proposed a new method to measure the similarity using a mimetic algorithm: A meta-heuristic algorithm (a combination of genetic algorithm and great deluge). Those lectures have implemented their methods on the same database "Corel collection's dataset images".

Besides of using low-level features of the images, researchers have been focusing on designing sophisticated algorithms in order to reduce the semantic gap and to improve the retrieval accuracy of CBMIR systems [27], accordingly they have associated high-level semantic features to this system, those features are mostly annotations or feedbacks given by specialists in the matter. Here we present some related works:

- Imon et al. 2018 [28] proposed a retrieval framework that uses a hybrid feature space (HFS) that is created by integrating low-level image features and high-level semantic terms through rounds of relevance feedback (RF) and performs retrieval based on similarity, to support semi-automatic image interpretation. The innovation of this method is that it can assume the semantic characteristics of the query image by rephrasing the representation of the query vector in the HFS via user feedback.
 - Application on radiographic database of 811 images with 69 unique types of bone tumors,
 - Authors start with an off-line processing to identify ROI and to record semantic and radiometric features and put them in HFS, an on-line processing to measure the cosine distance between database of the images vector's features and the query images vector's features,
 - As evaluation of the system in the presence of two specialist, this work achieved mean average precision (MAP) value 0.90, where the initial MAP with baseline CBIR was 0.20 high prediction accuracy (> 0.8) for the majority of the semantic features automatically predicted by the system.
- Ling Ma et al. 2017 [29] present a new method of CBMIR by considering fused, context-sensitive similarity.
 - The main idea is to fuse both of visual and semantic similarities between query image and each image in the database as their pairwise similarities, then construct a weighted graph of which nodes represent the images and edges, measure their pairwise similarities. Finally,

a new similarity measure is measured between the query image and each database image to complete the retrieval process,

- The Common CT Imaging Signs of Lung Diseases (CISLs) was used in this experiment,
- This method gave more accurate result for finding similarities between images and query images.

6 Synthesis

According to the different works reported in literature, CBIMR is a very large domain to work on and has become a major topic of research. After all these lectures, we have shown that feature extraction and similarity measurements are the most important keys of CBMIR. It is evident that the proposed methods have improved the performances of images retrieval, meanwhile the presence of some gaps make a new challenges for researchers to deal with it in this domain as reducing feature vectors, finding relevant features, reducing time processing, viability of large homogeneous dataset, presence of experts to validate the works, etc [20]. However, many researchers work on it and try to enhance the efficiency of the medical image retrieval system based on the visual features.

7 Conclusion

At the end of this chapter, we come out with a various knowledge about CBMIR systems and its importance and why researchers over the world are interested to develop it. To conclude, we have summarized all of the components, usages, procedures and the improvements of CBMIR throw the last decades. Hopefully, we attempt to add our touch in implementing it with auto-encoders architecture in the next chapters.

Chapter 2

Auto-encoders networks for Content-Based Medical Image Retrieval

Introduction

The most challenging about CBMIR, is the large amount of medical data images that give the rush to work on it and find relevant information to improve the medical care system. However, talking about large data brings to the trendy technologies of artificial intelligence where the notion of BIG DATA is treated by DEEP LEARNING.

Among the Deep learning techniques, Auto-encoders networks are one of the powerful tools, it's commonly used for classification tasks and feature extraction, it is commonly used for data dimension reduction, image processing, speech and facial recognition. Its advantage for dealing with complex structure inspires researchers to work with it to purchase more satisfying results.

In this chapter, we will introduce the theory of Auto-encoders with citing all its different properties, architectures and different types using a coding tutorial part to study and compare the differences between those types and architecture.

1 Auto-encoder network

Auto-encoders (AE) are a structure of neural network, a specific type that uses feed forward architecture [30,31], its particularity lies in the fact that the input is the same as the output. AE work by reducing data into a latent space representation and then reconstruct the output from this representation.

To understand how AE works, it is important to define the 3 main layers that characterize this later as shown in Figure 2.1, the input layer, the hidden layer where all the work is done and the output layer.

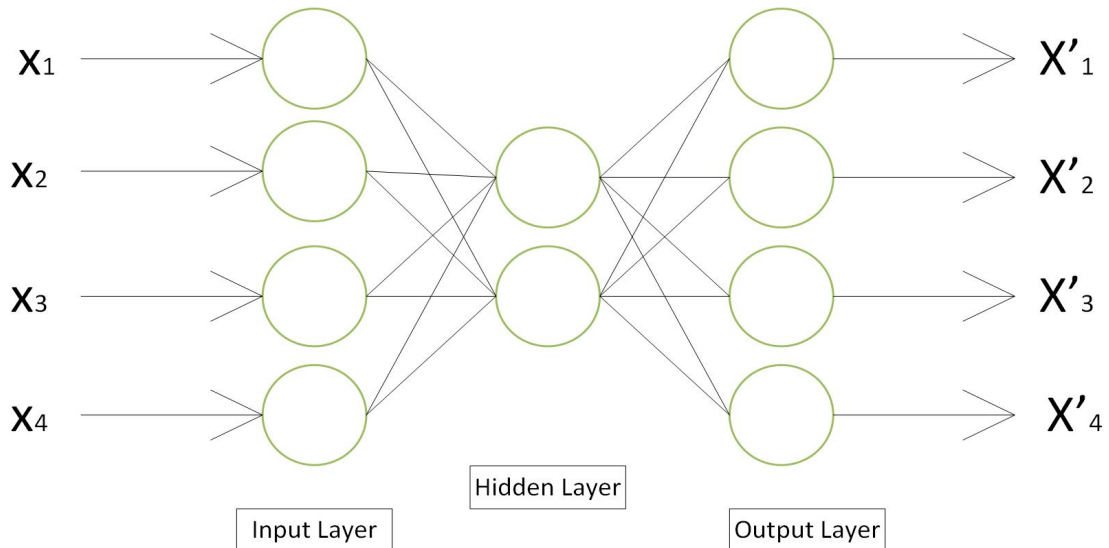


Figure 2.1: Auto-encoder network

The AE network can be divided into two consisting parts: encoder and decoder, between them there is the Code that represent the data in the latent space as it is shown in Figure 2.2 .

- Encoder: it works from the first layer to the hidden layer (see Figure 2.2), where the input data is compressed or reduced, and the code is produced into a latent space with an encoding function (eq.2.1) [30,32].

$$h = f(x) \quad (2.1)$$

h : represent the hidden layer, x : represent the samples, f : represent the activation function (as sigmoid or any rectified linear unit)

- Code: represent the hidden layer where the reduced data is compacted, in this layer all the treatments on the data are done, it is also called “**bottleneck layer**” [31,33,34] and represent the latent space mentioned earlier it contains the important features of the data .
- Decoder: it works from the hidden layer to the output layer (see Figure 2.2), it reconstructs the output using the code with a decoding function (eq.2.2) [30,32].

$$r = g(x) \quad (2.2)$$

r : represent the reconstruction, x : represent the samples, g : represent the reconstruction function

The whole auto-encoder is described by the function $g(f(x)) = r$ where r is the closest to the original input x .

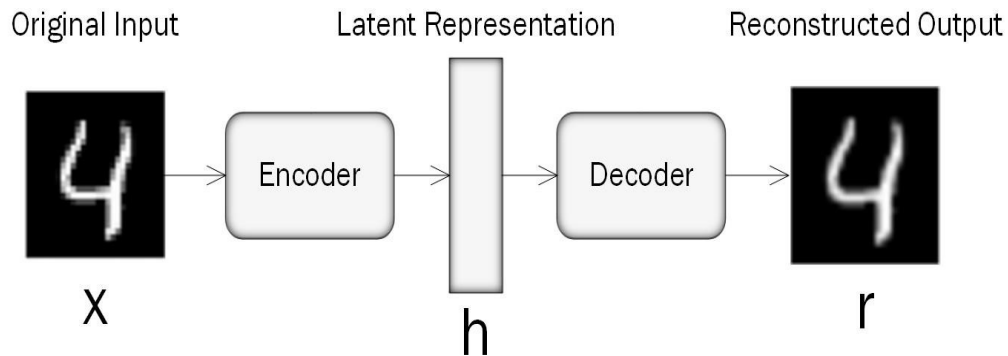


Figure 2.2: The auto-encoder system.

Eventually, AE are dimension reduction algorithm, hence, it takes the input which is a high dimension representation (like an image or a vector) and compresses it into a smaller dimension representation [33], However, AE came with important properties [30]:

- **Data-specific:** it means that AE is only able to treat or compress data similar to what they have been trained on [35], the difference between AE and other standard algorithm of data compression is that we can't expect that an AE trained on handwriting digits to compress landscape images [30].
- **Lossy:** the similarity between input data and output data is not 100% accurate, the result will be degraded compared to the original inputs [35].
- **Unsupervised:** AE is learned automatically from instances data; it doesn't need labels data because it generates its own labels from training data [30].

In other circumstances, AE might be confused with principal component analysis (PCA) because they are quite related when the activation function of the AE is linear, the latent space or known as bottleneck layer would be directly corresponded to the major component of the PCA. Nevertheless, the general activation function used in AE is non-linear [36] as ReLU (Rectified Linear Unit) and sigmoid.

2 Architecture of a traditional Auto-encoder

AE architecture is simply an artificial neural networks (ANN's) where the encoder and the decoder are fully-connected feed-forward layer, meanwhile code is the only layer where its size dimension (the number of nodes layer) depends on a hyper-parameter that is set before the training of the AE [30]. This implies that AE require to have the same dimensionality of the input and the output while the bottleneck layer can be malleable.

There are 4 hyper-parameters that need to be fixed before the AE training:

- **Code size:** number of nodes of the bottleneck layer,
- **Number of layers:** the network of AE can be deeper as needed,

- **Number of nodes per layer:** it depends on the type of the fixed structure of AE (see section 3) ,
- **Loss function:** the most used are mean squared error (mse) and the binary cross-entropy.

After setting all these parameters, the training is proceeded as the ANNs through back-propagation [30].

Simple auto-encoder's code tutorial ¹

First of all we will start to create the encoder layer and the decoder layer

```

1 from keras.layers import Input, Dense
2 from keras.models import Model
3
4 encoding_dim = 32 # size of the encoded representations
5 input_img = Input(shape=(784,)) # input placeholder
6 encoded = Dense(encoding_dim, activation='relu')(input_img)
7 # the encoded representation of the input
8 decoded = Dense(784, activation='sigmoid')(encoded)
9 # the lossy reconstruction of the input
10 autoencoder = Model(input_img, decoded)
11 #this model maps an input to its reconstruction
12
13 #create an encoder and decoder model
14 encoder = Model(input_img, encoded)
15 encoded_input = Input(shape= encoding_dim,)
16 # retrieve the last layer of the autoencoder model
17 decoder_layer = autoencoder.layers[-1]
18 decoder = Model(encoded_input, decoder_layer(encoded_input))

```

Then the training of the auto-encoder to reconstruct MNIST digits.

```

1 autoencoder.compile(optimizer='adadelta', loss='binary_crossentropy')
2 #configuration of the model

```

The preparation of the input data(MNIST). We're ignoring the labels.

```

1 from keras.datasets import mnist
2 import numpy as np
3 (x_train, _), (x_test, _) = mnist.load_data()
4
5 # normalize all values between 0 and 1 and collecting the
6 #28x28 images into vectors of size 784
7 x_train = x_train.astype('float32') / 255.
8 x_test = x_test.astype('float32') / 255.
9 x_train = x_train.reshape((len(x_train), np.prod(x_train.shape[1:])))
10 x_test = x_test.reshape((len(x_test), np.prod(x_test.shape[1:])))
11 print(x_train.shape)
12 print(x_test.shape)
13 # (60000, 784)
14 # (10000, 784)
15
16 #train the autoencoder for 50 epochs
17 autoencoder.fit(x_train, x_train,

```

¹Note: to run this code, Keras version 2.0.0 or higher is required.

```

18         epochs=50,
19         batch_size=256,
20         shuffle=True,
21         validation_data=(x_test, x_test))

```

After **50 epochs**, the auto-encoder reach a stable train/test loss value of about **0.1028**. Finally, we will visualize the reconstructed inputs and the encoded representations. The result is shown in Figure 2.3 .

```

1 # encode and decode some digits from the test set
2 encoded_imgs = encoder.predict(x_test)
3 decoded_imgs = decoder.predict(encoded_imgs)
4
5 # use Matplotlib
6 import matplotlib.pyplot as plt
7 n = 10 # how many digits we will display
8 plt.figure(figsize=(20, 4))
9
10 for i in range(n):
11     # display original
12     ax = plt.subplot(2, n, i + 1)
13     plt.imshow(x_test[i].reshape(28, 28))
14     plt.gray()
15     ax.get_xaxis().set_visible(False)
16     ax.get_yaxis().set_visible(False)
17
18     # display reconstruction
19     ax = plt.subplot(2, n, i + 1 + n)
20     plt.imshow(decoded_imgs[i].reshape(28, 28))
21     plt.gray()
22     ax.get_xaxis().set_visible(False)
23     ax.get_yaxis().set_visible(False)
24 plt.show()

```



Figure 2.3: Result of the simple AE, The top row is the original digits, and the bottom row is the reconstructed digits.

3 Auto-encoders structure types

Diverse types of auto-encoders structures have been developed for different application frameworks: data compression, data generation, classification ...etc. This diversity remains in the middle layer because it is the only part that can be adjusted. Those adjustments depend on study cases, each case [37] need an ap-

propriate type as the following cases:

An example of data embedding onto a very low dimensional space for visualization and exploratory analysis.

A case where noisy signals are to be repaired by the model.

A different example where very high dimensional sparse data, such as text documents, is to be compressed onto compact binary codes in a semantic way.

Additionally, we study anomaly detection, the situation where abnormal patterns are to be detected in sequences but no anomalies are available to learn from.

As a last case study, we propose the generation of new instances which do not belong to the training set.

3.1 Denoising auto-encoder

One of the most known AE regularized types is the denoising auto-encoder (DAE). In order to force the AE to learn more useful features and keeping the hidden layer small, the AE could learn intelligent representation of data [30], DAE tends to add a stochastic random noise to the input data before it goes to the encoder as it is shown in Figure 2.4. This operation makes the AE learning how to remove the added noise [32] and reconstruct the original data from that corrupted data, beyond, DAEs will obtain a robust latent space with more meaningful features extracted [38]. It is important to note that changing the reconstruction term error of the loss function it's going to be $L(x, r)$ instead of $L(x', r)$, where : L = loss function, x = input data, x' = corrupted data, r = reconstructed data "output". Basically, the system is comparing the output with the original input and the decoder has to reconstruct the original free-noise data [32,38].

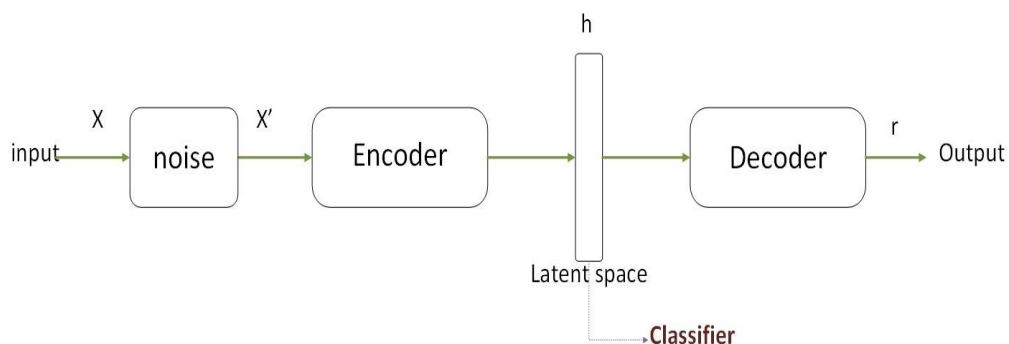


Figure 2.4: Denoising auto-encoders system

DAEs are generally used for data denoising, feature selection and extraction and dimensionality reduction, we can find some related works as the work of Vincent et al. [39] who proposed a paper on extracting and composing robust features with DAEs, the authors found that they could improve the robustness of their internal layers by purposely introducing noise to their signal. In addition, the same authors in [40], proposed the idea of using stacked DAEs to learn useful representations in a deep network with local denoising criterion.

Denoising auto-encoder's code tutorial

This type of auto-encoder can be trained to map noisy digits images to clean it. At first, we need to generate a noise by applying a Gaussian noise. Figure 2.5 show how the noisy images looks like.

```

1 from keras.datasets import mnist
2 import numpy as np
3
4 (x_train, _), (x_test, _) = mnist.load_data()
5
6 x_train = x_train.astype('float32') / 255.
7 x_test = x_test.astype('float32') / 255.
8 x_train = np.reshape(x_train, (len(x_train), 28, 28, 1))
9 x_test = np.reshape(x_test, (len(x_test), 28, 28, 1))
10
11 noise_factor = 0.5
12 x_train_noisy = x_train + noise_factor *
13     np.random.normal(loc=0.0, scale=1.0, size=x_train.shape
14     )
15 x_test_noisy = x_test + noise_factor *
16     np.random.normal(loc=0.0, scale=1.0, size=x_test.shape
17     )
18 x_train_noisy = np.clip(x_train_noisy, 0., 1.)
19 x_test_noisy = np.clip(x_test_noisy, 0., 1.)
20
21 import matplotlib.pyplot as plt
22 n = 10
23 plt.figure(figsize=(20, 2))
24 for i in range(n):
25     ax = plt.subplot(1, n, i+1)
26     plt.imshow(x_test_noisy[i].reshape(28, 28))
27     plt.gray()
28     ax.get_xaxis().set_visible(False)
29     ax.get_yaxis().set_visible(False)
30 plt.show()

```



Figure 2.5: Some examples from the noisy images

Now let's start building the auto-encoder:

```

1 input_img = Input(shape=(28, 28, 1))
2
3 x = Conv2D(32, (3, 3), activation='relu', padding='same')(input_img)
4 x = MaxPooling2D((2, 2), padding='same')(x)
5 x = Conv2D(32, (3, 3), activation='relu', padding='same')(x)
6 encoded = MaxPooling2D((2, 2), padding='same')(x)
7

```



```

8 x = Conv2D(32, (3, 3), activation='relu', padding='same')(encoded)
9 x = UpSampling2D((2, 2))(x)
10 x = Conv2D(32, (3, 3), activation='relu', padding='same')(x)
11 x = UpSampling2D((2, 2))(x)
12 decoded = Conv2D(1, (3, 3), activation='sigmoid', padding='same')(x)
13
14 autoencoder = Model(input_img, decoded)
15 autoencoder.compile(optimizer='adadelta', loss='binary_crossentropy')

```

We train the network for **100 epochs**, Figure 2.6 illustrate the results.

```

1 from keras.callbacks import TensorBoard
2 autoencoder.fit(x_train_noisy, x_train,
3               epochs=100,
4               batch_size=128,
5               shuffle=True,
6               validation_data=(x_test_noisy, x_test),
7               callbacks=[TensorBoard(log_dir='/tmp/tb',
8               histogram_freq=0, write_graph=False)])

```



Figure 2.6: Result of the denoising AE, The top row is the noisy digits, and the bottom row is the reconstructed digits.

In this case the loss value is calculated between the original input(not the noisy images) and the decoded images. And it produce a loss value of **0.094**.

3.2 Sparse auto-encoders

Another popular AEs regularized type is Sparse auto-encoder (SAE) commonly used for data dimension reduction and feature selection [33, 38], instead of pre-processing the input vectors before it enters to the AE system like DAEs, SAE as presented in Figure 2.7, implement it in adding a sparsity constraint on the encoded vector. It allows to add a penalty term to the loss function and make a few nodes active that have nonzero values, this operation force the SAE system to represent each input as a combination of small number of nodes, this enables to discover interesting structure in the data [30].

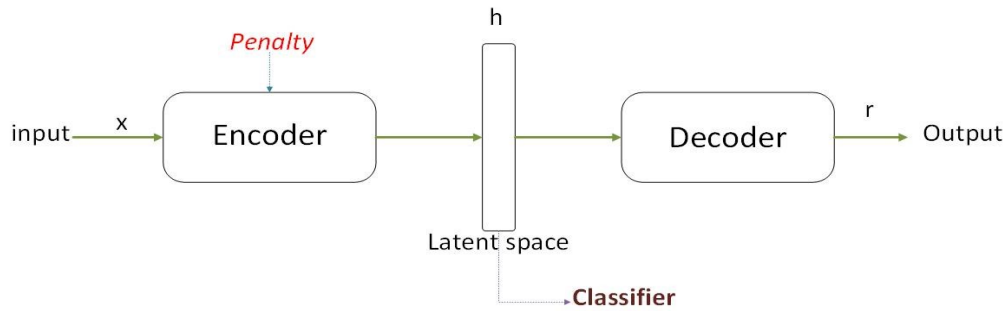


Figure 2.7: Sparse auto-encoder system

SAEs are becoming a trend in the in classification research filed because it has the ability of finding good features specially in speech and face recognition as it was demonstrated by Verma et al. [41] on their comparison study between sparse AEs and SVM classifier in the fault diagnosis of air compressors. For facial recognition, Zeng et al. [42] used SAEs in deep learning for facial expression recognition.

Sparse auto-encoder's code tutorial

For a sparse model, we need to add a sparsity constraint on the activity of the hidden representations. In Keras, this can be done by adding an `activity_regularizer` to our Dense layer. We use an L1 regularization, which mean adding an "absolute value of size" of coefficients as penalty term.

```

1 from keras import regularizers
2
3 encoding_dim = 32
4
5 input_img = Input(shape=(784,))
6 # add a Dense layer with a L1 activity regularizer
7 encoded = Dense(encoding_dim, activation='relu',
8                 activity_regularizer=regularizers.l1(10e-5))(input_img)
9 decoded = Dense(784, activation='sigmoid')(encoded)
10
11 autoencoder = Model(input_img, decoded)

```

We train this model for **100 epochs**. The models ends with a train loss of **0.11**. In Figure 2.8 a visualization of our new results.



Figure 2.8: Result of the sparse AE, The top row is the original digits, and the bottom row is the reconstructed digits.

3.3 Variational auto-encoders

Unlike the previous types, Variational auto-encoders VAEs are more defined as a Generative model where the system is viewed in different way as shown in Figure 2.9.

The encoder presents an inference procedure and the decoder is presented as a stochastic mapping which allows the AE system to produce a probabilistic description of the data through inference (calculating the latent space variables) and learning (optimizing the hyper-parameters) [38]. The particularity in this architecture is that the input is mapped to a probabilistic distribution in the latent space and represented in two vectors instead of one vector like the other types: one is representing the mean of the distribution and the other represent the standard deviation of the distribution, which leads to a VAEs loss function with two terms [33]: the reconstruction loss and regularized with the KL metric (Kullback-Leibler divergence).

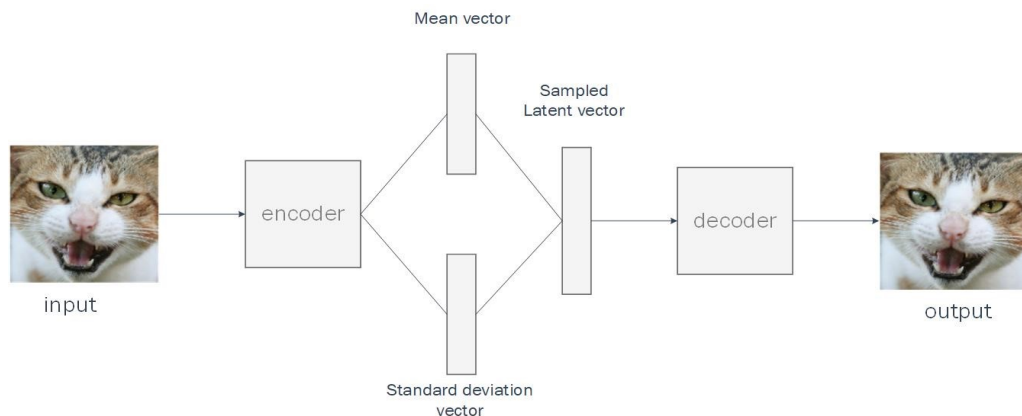


Figure 2.9: Variational auto-encoder system

VAEs have proven their power in classification in unsupervised and reinforcement modes [43], being a generative model ², it's a useful tool for reliability and risk analysis. Most recent researches are linked to the theory of graphs: Simonovsky and Komodakis [44] worked on the application of VAEs to generate small graphs, also in chemistry field, Liu et al. [45] proposed a graph model using VAEs to generate molecule design. Other examples of VAEs are used in prediction, like Walker et al. [46] to forecast future from static images.

Variational auto-encoder's code tutorial

From our knowledge and from what we read, this model does not apply for the CBIR domain. We can't present a tutorial example in this context. Thereby, we lead the readers of interest to this tutorial page ³.

²Generative models encode full probability distributions and specify how to generate data that fit such distributions. By sampling from this model, new data can be generated.

³https://github.com/keras-team/keras/blob/master/examples/variational_auto-encoder.py

3.4 Stacked auto-encoders

What about having many auto-encoders in one architecture! In Stacked AE we can go deeper having many auto-encoders put together with multiple layers of encoders and decoders [31]. This deep architecture (see Figure 2.10) makes the auto-encoders very powerful to learn complex coding, in the other hand it is important to handle this complexity carefully to avoid over-fitting problems [47]. The representations generated by SAEs are more robust and useful compared eventually to ordinary AEs, it is also had proven effectiveness in reducing the computation cost exponentially and the amount of training data [38]. Although SAEs are another form of AEs where the layers could be convolutional and deconvolutional layers especially when it is used for image processing [31].

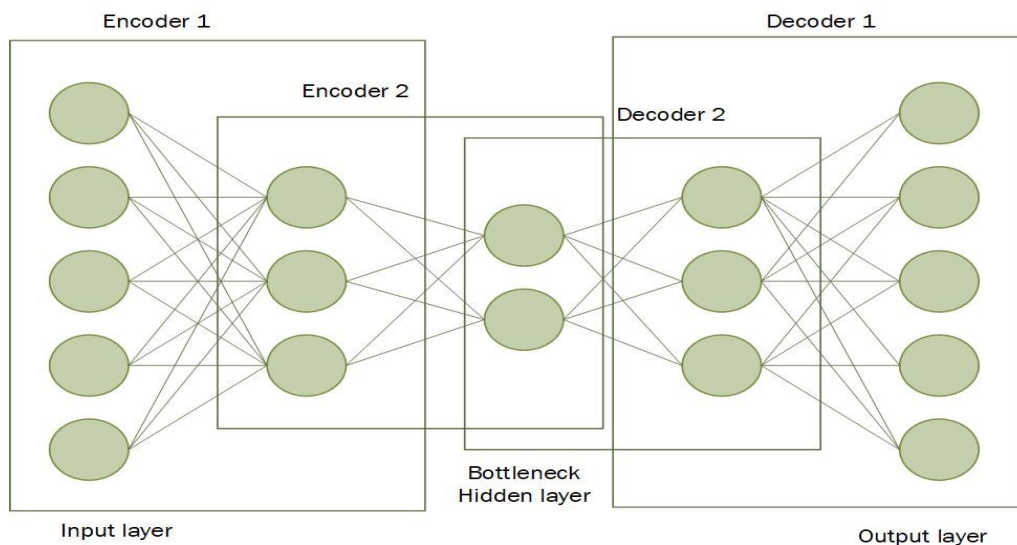


Figure 2.10: Stacked auto-encoder Architecture.

SAEs are generally used in classification and image processing with feature extraction. There are many interesting subjects for SAEs applications. We can cite the works of Janod et al. [48] for speech recognition, Bao et al. [49] who present a framework for financial time series using a combination of SAEs and LSTM (Long-Short term memory). For medical interest, Kannadasan et al. [50] used SAEs for comparison purpose for data classification of Type 2 diabetes and Berlin et al. [51] proposed a system for skin segmentation images and videos using SAEs.

Stacked auto-encoder's code tutorial

In this type of auto-encoder we can use a stack of layers.

```

1 input_img = Input(shape=(784,))
2 encoded = Dense(128, activation='relu')(input_img)
3 encoded = Dense(64, activation='relu')(encoded)
4 encoded = Dense(32, activation='relu')(encoded)
5
6 decoded = Dense(64, activation='relu')(encoded)

```

```

7 decoded = Dense(128, activation='relu')(decoded)
8 decoded = Dense(784, activation='sigmoid')(decoded)

```

And then we will put this part:

```

1 autoencoder = Model(input_img, decoded)
2 autoencoder.compile(optimizer='adadelta', loss='binary_crossentropy')
3
4 autoencoder.fit(x_train, x_train,
5               epochs=100,
6               batch_size=256,
7               shuffle=True,
8               validation_data=(x_test, x_test))

```

After 100 epochs, it reaches a train loss of **0.099**. A visualization is in Figure 2.11 .



Figure 2.11: Result of the stacked AE, The top row is the original digits, and the bottom row is the reconstructed digits.

However, it is important to note that all these modifications can cause the Over-complete Hidden Layer issue, which implies that the AE simply copies the input into the output and does not learn any meaningful representation, there will be no feature extraction but rather a perfect reconstruction of the input data, which means an over-fitting situation [30,31]. Note that this situation can only occur when the hidden layers is greater than the input and output layers.

4 Related works of Auto-encoders in medical image retrieval

Recently, the use of auto-encoders as a technique for feature extraction from medical imaging have raised a great interest, such approaches are used to create a compressed feature vector that can be a help in both reduce the storage space and computational cost of the retrieval operation. Table 2.1 presents some applications for image analysis using various types of auto-encoders.

TITLE	METHOD	DATASET	METHODOLOGY	REMARKS
<i>Auto-encoding the Retrieval Relevance of Medical Images [52]</i>	auto-encoder (AE)	IRMA	<ol style="list-style-type: none"> 1. Train a single auto-encoder over the database and record the error histogram of each class 2. Eliminate image blocks that are irrelevant for retrieval 3. use SVM to classify the images 	<ul style="list-style-type: none"> -This approach tries to eliminate unnecessary blocks of the image from feature extraction process. -Space cut down by 50% -Speed of the retrieval increased by 27%.
<i>Barcodes for Medical Image Retrieval using Auto-encoded Radon Transform [53]</i>	auto-encoders on Radon barcodes (ARBC)	IRMA	<ol style="list-style-type: none"> 1. Pre-Processing of Images: resized images to fixed dimensions; extract the Radon features 2. Pass the radon barcodes features to the auto-encoder (For n hidden layers, n barcodes can be generated) 	-The ARBC achieve better results than the traditional Radon barcodes (RBC)
<i>Stacked auto-encoders for Medical Image Search [54]</i>	Stacked auto-encoders (SAE)	IRMA	<ol style="list-style-type: none"> 1. Use the pixel values as features into the auto-encoder 2. Extract the complex anatomical features using the reconstruction error as its guide 	<ul style="list-style-type: none"> -This approach is used to create one global feature descriptor (vector) for a given image -A retrieval error of 376 for 1,733 test images with a compression of 74.61%
<i>Using Very Deep auto-encoders for Content-Based Image Retrieval [55]</i>	- Stacked auto-encoders (SAE) -semantic hashing	CIFAR-10	<ol style="list-style-type: none"> 1. Pre-training with RBM (Restricted Boltzmann Machines) 2. Use features to initialize a 28-bit and a 256-bit AE with the weights from two separately trained stacks of RBMs 3. Use the AE to map images to short binary codes 4. Use the codes to retrieve images that are like a query image by flipping a few bits in the code and performing a memory access, 	- For CBIR, binary codes have many advantages compared with directly matching pixel intensities; they are very cheap to store, and they are very fast to compare, -the best retrieval performance is given by the method "28-bit+256-bit deep codes"

Table 2.1: Recent works about auto-encoders in the image analysis domain.

5 Synthesis

In summary, there are multiple types of structure besides the ones cited previously, each one completes a precise purpose: data denoising, dimensionality reduction, or even data generating. Such as denoising auto-encoder found to be useful in data denoising, and a variational auto-encoder is so famous in the data-generating field; but when talking about dimensionality reduction, the stacked auto-encoder is the best with no competitor.

The main purpose of our project is to use one of those techniques to improve CBMIR. When talking about CBMIR immediately come to mind medical images and huge amount of data; where we thought the Stacked auto-encoders will give the best results.

In fact, there are a lot of data already available in digital archives of the hospitals that lead us to the world of “**Big Data**”. Those Data can help to reduce misdiagnosis through retrieving similar cases to exploit available knowledge from the past. However, it can be computationally expensive to give a new image, comparing it to the entire collection of medical images and find similar images.

Such task requires a technique that efficiently searches for large image data and still finds very similar images without costing that much. Multiple hidden layer neural networks can be helpful in solving this problem. Each layer can learn features at a different level of abstraction. However, the training of neural networks with many hidden layers can be difficult in practice. One way to effectively train a multilayer neural network is to train one layer at a time. You can achieve this by training a special type of network known as an auto-encoder (AE). This method can reduce the size images in a smaller, more compact representation to enable faster image comparisons during the retrieval task. Here we can see the importance of the auto-encoder approach, which has a power reduction data dimension and therefore increase the speed of the retrieval task.

6 Conclusion

In this chapter, the theory of auto-encoders was deeply discovered with its various types and properties, with diverse application. In summary, the power of the auto-encoders stands on its unique particularity of its architecture with two components encoder and decoder, and the latent space between them, where the retrieval on the compressed dimension of an input data is made with relevant feature extraction. From the study of the different types of auto-encoders, we conclude that the stacked auto-encoders are the best candidate for the CBMIR task. In the next chapter, this method will be used in developing of a proposed framework for the recognition of retrieval COVID-19 images.

Chapter 3

Content-Based Medical Image Retrieval for COVID-19 investigation

Introduction

Since the human being existed, the planet faced an amount of life threat coming from diseases and he learned to manage them with the supplies and tools that were available at that time. The COVID-19 pandemic that affected the whole planet and caused the death of thousands of people and still paralyzing the going of our lives is not different from the other previous pandemics. Fortunately, we are living in an era where the revolution of industrial technologies has reached its peak, where, AI is more solicited to provide its tools and applications to the health care systems through this pandemic [56], as a remark the number of companies that have implemented AI-related technologies has increased by 270% in the past four years [57]. Until today, where we are developing this work, the virus is spreading, and fighting it has become a worldwide challenge, however, every expert in the world agrees that AI tools can be our weapon of salvation, It is the only fast way that help to manage it in short time.

We must admit that AI provide relevant applications to promote the efficiency of medical treatment and give also better experience without proposing the medical staff to risk of contamination [56]. Researchers are working more on ML algorithms to procure more AI driven tools from the analyzing applications for the decision-making applications in order to identify the COVID-19 [58]. Meanwhile, AI applications are spotted in different stage of this pandemic where it can contribute to the fight against COVID-19 [59] as it cited below:

Detection: (diagnosis and prognosis) researchers are focusing and growing there effort to train and develop AI models that help to detect and diagnose the virus using mostly Chest Radiography images X-Rays and CT scans [60]. AI can be as accurate as humans, it can save radiologist time and present a faster and cheaper diagnosis than the standard COVID-19 tests. Alom et al. [61] used a Convolutional Neural Network with Transfer Learning (TL) approach to detect

COVID-19 and a NABLA-N¹ network model to segment the regions infected by COVID-19. Ozturk et al. [63] proposed "The DarkCovidNet model" using an already proven models "The DarkNet model" which implement the You Only Look Once (YOLO) real time object detection system, they implemented 17 convolutional layers and introduced different filtering on each layer. Rosebrock [64] provide a tutorial using Deep Learning to diagnose COVID-19 from Chest X-ray. It should be noted that new approaches and models are proposed on a daily basis, and much more initiatives are underway in this regard [65,66].

Prevention: (tracking and prediction) to follow the saying of "better to prevent that to cure", researchers have established AI models in order to analyses the behavior of this virus , track and predict how the COVID-19 disease will spread over time and over space [60], and to prevent the population from other circumstances of this disease. Such as the model proposed by Akhtar et al. [67] was developed with a dynamic neural network to predict COVID-19 spread. This may help to plan, prepare, and manage the pandemic.

Response: (social control) AI is offering its tools to manage the information from the fake news to relevant information, this is what it's called SOCIAL Control [60] used by some government as CHINA to detect people having COVID-19 disease by scanning the high temperature using a facial recognition system [68], also to manage the social distancing measures there was a start-up [69] who introduced a computed vision system in USA using drones.

Recovery: (early warnings and alerts) for this part AI contribution will help to monitor the economic crisis and its recovery via satellite, social networking and other data. This will help to learn and built an early warning for a future crisis [70].

There is also an honorable mention of the AI contributions in the development of treatments and vaccines, which is not a new thing done, AI has been for years used for biological studies of genes and proteins [71]. Regrettably, there was only one constraint for AI during this pandemic, which is the lack of DATA which is justified in front of this outbreak virus, but it hasn't destabilized the researchers have implemented some of AI driven tools that are useful in such scenarios: Active learning, Cross population train/test and multitudinal and multi-modal data [58].

To add COVID-19 has many unique features. While the diagnosis is confirmed with the use of polymerase chain reaction (PCR), a lot of reports [72–74] confirmed the use of chest computed tomography (CT) and chest X-ray analysis as a key method to examine suspected patients for COVID-19 confirmation. It's considered as a faster and easier approach to identify COVID-19 and available in all the clinical centers contrariwise the PCR test.

¹NABLA-N Network: A new DL architecture proposed by Alon Zahangir at al. [62] for skin Cancer segmentation

After introducing the AI aspects and contribution through this outbreak of COVID-19, in this chapter, we will present our proposed model for the recognition of COVID-19 disease from other cases. Our goal is to use medical images to develop AI-based approaches for CBMIR to detect COVID-19 cases.

1 Proposed framework

In the current situation of the COVID-19 pandemic (first and second wave), we proposed a framework dedicated to the epidemiologists as an application for the investigation of the first person (called PATIENT-0) infected by the virus across the region and the country also understanding and identifying the transmission chain and development of the pandemic.

Certainly, epidemiologists will have to present a report containing all statistics, studies, and information regarding this virus and defining from where it started would be relevant information to figure out the origin and how it developed, that's the major contribution of our work. For the record, some researchers are working now on this thematic, for example, we cite the system of surveillance and investigation established in France to define the first patients with COVID-19 [75] it used the historical traveling map to track all movements of the potentially contaminated subjects.

In our proposed approach, practitioners will be allowed to search similar images cases in a database using a CBMIR system, where the retrieval would be done on encoded images. The algorithm is done by creating a global feature descriptor in the form of vector for every image in the database, using the encoder part of the pre-trained SAE to generate a compressed feature vector. Then the same treatment for a query image. For retrieving similar cases with the query image a comparison will be made between the query image's vector and the database image's vectors to retrieve the most similar vectors using the K-nn approach by calculating the distance between these vectors. Finally, for a more visible of the results, the query image's vector and his similar vectors will pass on the decoder part of the pre-trained SAE to be decompressed into images.

The SAE training use the pixel values as features, with $I/p/n/m/n/p/I$ for the architecture (we use 3 AE's arranging them serially to obtain our SAE's) and a setting of $n < m < p < I$ so the dimensionality reduction will be done, encodes I input into p position as a first layer, and then for the second layer encode p input into n position, and finally encode n input into m position. m represents the global feature descriptor for a given image which means our vector. And for the decoder part, it will work in a contrariwise way.

Figure 3.1 shows the steps proceeded in our framework, when the query image is injected into the system, it will be encoded into a vector and then compared with the other images reduced into vectors earlier in the latent space. Similar vectors will be selected and then reconstructed as image results, the output of the framework will be a subset of similar images sorted historically from the newest

to the oldest ones, to track the first case of COVID-19 in the region.

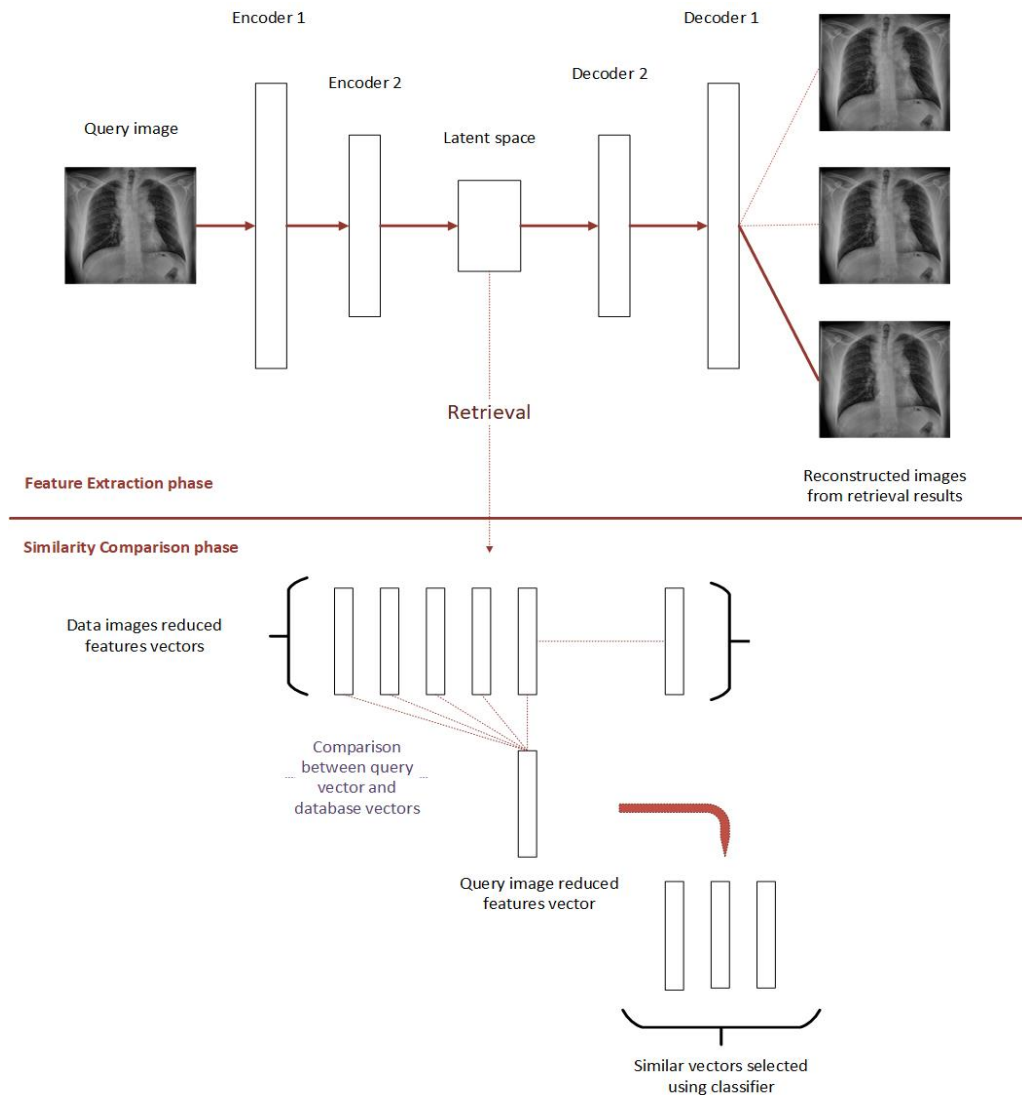


Figure 3.1: Scheme of the proposed framework

2 Experiments and results

In this section, we describe the COVID-CHEST-XRay IEEE8023 dataset used and the process of the proposed framework with results discussion.

2.1 COVID-19 Images dataset

With the declaration of the COVID-19 pandemic, researchers from around the world, either they are academics or professionals start to collect data images built from multiple spot where the virus was most diffused (eg. Italy, Wuhan-CHINA) to build databases available to be used for the search of possible solutions of this outbreak pandemic. In [76], we can find the different large collections that were picked minutely. Its should be noted that at this time of reading, more data are

collected daily.

In this project, we used COVID-CHEST-XRay IEEE8023 dataset that contains 760 images collected from public sources as well as through indirect collection from hospitals and physicians, it is built from chest X-ray and CT images, it comprises images with different type of pneumonia (COVID-19, SARS, MERS, Streptococcus spp, Pneumocystis spp...., ARDS), Table 3.1 below show exactly the types, modality and the exact number of samples for each case. Figure 3.2 an overview of the each types. This dataset collection was approved by the University of Montreal's Ethics Committee [77] and published in an open-source platform².

TYPE	CLASS (SPECIES, GENUS)	IMAGE COUNT	MODALITY
VIRAL	COVID-19(SARSr CoV-2)	533	453 X-Ray/ 80 CT
	SARS (SARSr-CoV-1)	16	X-Ray
	Varicella	5	X-Ray
	Influenza	5	X-Ray
BACTERIAL	Streptococcus spp.	17	X-Ray
	Klebsiella spp.	8	X-Ray
	Escherichia coli	4	X-Ray
	Mycoplasma spp.	8	X-Ray
	Legionella spp.	4	X-Ray
	Unknown	1	X-Ray
	Chlamydomphila spp.	2	X-Ray
FUNGAL	Pneumocystis spp.	17	
LIPOID	No applicable	5	
UNKNOWN	Unknown	17	X-Ray

Table 3.1: Count of type of pneumonia with species and genus.

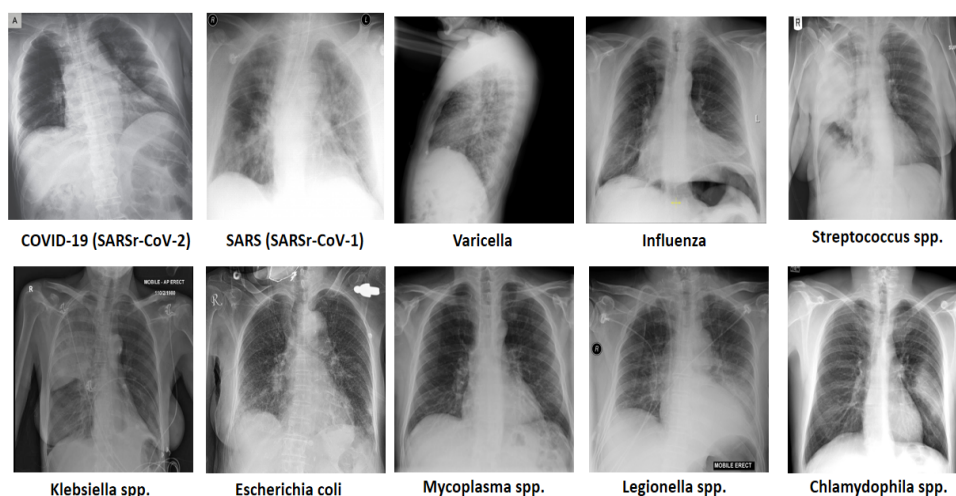


Figure 3.2: Image samples of the dataset

²<https://github.com/ieee8023/covid-chestxray-dataset>

2.2 Process of the proposed approach

In this study, we used Python programming language version 3.8 with DL and ML libraries as Tensorflow 2.2.0 and Keras 2.4.3.

We have 760 samples divided into 560 samples for the training and 200 samples for the validation.

Image processing: After loading the samples, each image is gray-scaled and resized into 128, reshaped into 128x182 vector, and normalized in [0,1]. To feed the first layer of the auto-encoder the data must be in a column vector of size equal to 16384.

auto-encoder's settings architecture: After transferring the full images dataset to vectors, it's time to feed it to the auto-encoder. The architecture of our stacked auto-encoder is as follow: 5 hidden layers represented in encoders and decoders and it serially arranging such as the output of the first layer is fed as input to the second layer and so on, with the same principle in the decoder part and the third layer gives us the final feature vector for the input image.

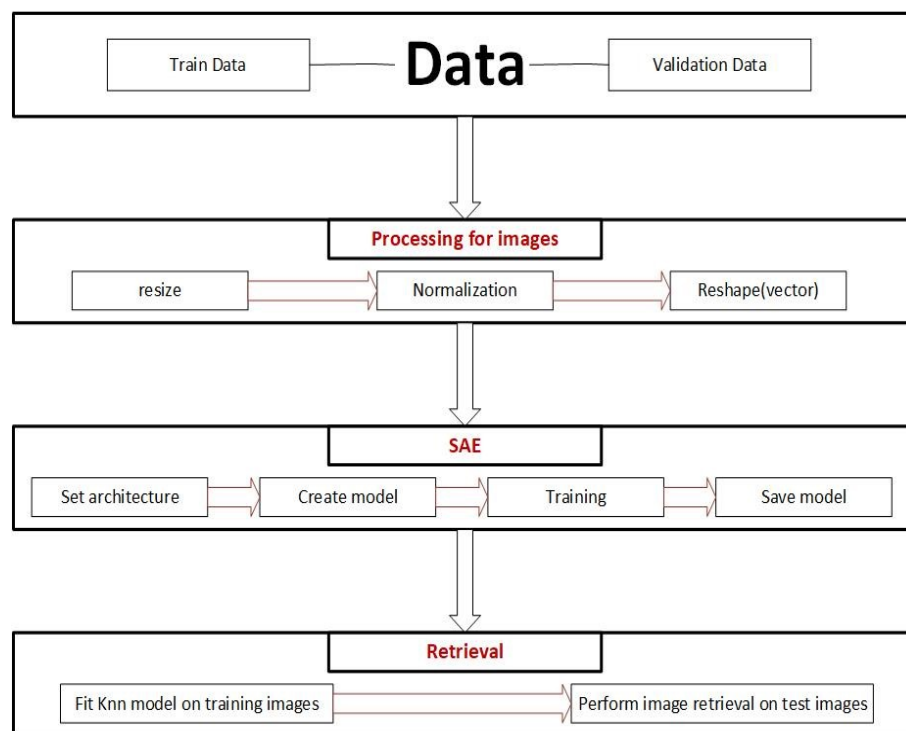


Figure 3.3: Image samples of the dataset

The auto-encoder obtains $I/p/n/m/n/p/I$ as an architecture where I is the size of our input vector and $n < m < p < I$. The structure of the proposed algorithm is resumed in Figure 3.3 which describes the important steps to achieve our goal.

The goal of this CBMIR system is to retrieve the most similar images from a dataset to the query image given by the user. To achieve a successful retrieval a specific algorithm should be followed. Processing the images and the feed it as a vector to the SEA, this step is done for the 760 images from the COVID-CHEST-XRay dataset and then those images will be compressed as a feature vector that describes the image content. The final step is the retrieval part, the query image will pass in the pre-trained AE and compressed into a feature vector, and then a comparison will be done by K-nn to find the most similar vectors.

2.3 Results and discussion

The configuration settings for our stacked auto-encoder (with 3 Encoders) is **16384/ 8100/ 4000/ 2000/ 4000/ 8100/ 16384**, with **5 hidden layers** and use the following settings:

- We divided the dataset into 1/3-2/3, this comes to a total of 560 samples for the training phase and 200 samples for the validation.
- Encoder 1 : 16384/ 8100/ 16384,
- Encoder 2 : 8100/ 4000/ 8100,
- Encoder 3 : 4000/ 2000/4000,
- Reshape 16384,
- 500 epochs,
- Batch size of 256 for stochastic gradient descent,
- **“Relu”** as the default activation function for the different layers, since we are working in non-linear space,
- **“mean squared error”** as the loss function, it is necessary for relaying the concepts of precision, bias, and accuracy during the statistical estimation of this study,
- **“adadelta”** used for the parameter’s optimizer, as a stochastic gradient descent method that adapts learning rates based on a moving window of gradient updates.

Figure 3.4 show what the encoder and the decoder models look like.

encoder.summary(): Model: "model_1"			decoder.summary(): Model: "model_2"		
Layer (type)	Output Shape	Param #	Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 16384)]	0	input_2 (InputLayer)	[(None, 2000)]	0
dense (Dense)	(None, 8100)	132718500	dense_3 (Dense)	(None, 4000)	8004000
dense_1 (Dense)	(None, 4000)	32404000	dense_4 (Dense)	(None, 8100)	32408100
dense_2 (Dense)	(None, 2000)	8002000	dense_5 (Dense)	(None, 16384)	132726704
Total params: 173,124,500 Trainable params: 173,124,500 Non-trainable params: 0			Total params: 173,138,884 Trainable params: 173,138,884 Non-trainable params: 0		

Figure 3.4: Summary of the encoder and the decoder architecture from our model.

While the whole SAE is in Figure 3.5.

autoencoder.summary(): Model: "model"		
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 16384)]	0
dense (Dense)	(None, 8100)	132718500
dense_1 (Dense)	(None, 4000)	32404000
dense_2 (Dense)	(None, 2000)	8002000
dense_3 (Dense)	(None, 4000)	8004000
dense_4 (Dense)	(None, 8100)	32408100
dense_5 (Dense)	(None, 16384)	132726704
Total params: 346,263,384 Trainable params: 346,263,384 Non-trainable params: 0		

Figure 3.5: Summary of the SAE architecture.

When the SAE finish the training, we obtain as a final value of the loss function (reconstruction error) **0.0588** for the training and **0.0532** for the validation.

After the encoding step exactly in the third layer that represents the latent space, the size of our final feature vector is 2000, as shown in Figure 3.5. Which is report a compression of 87% reduction.

For the retrieval step, we use a K-nn model with k=5 to retrieve the most identical images to the query image. By next, as we mention above the query image and all his retrieved images will pass to the decoder to be reconstructed and then proposed to the user.

We tested the model on a set of images from the dataset that represents different cases, we remove the labels in the beginning before the training step.

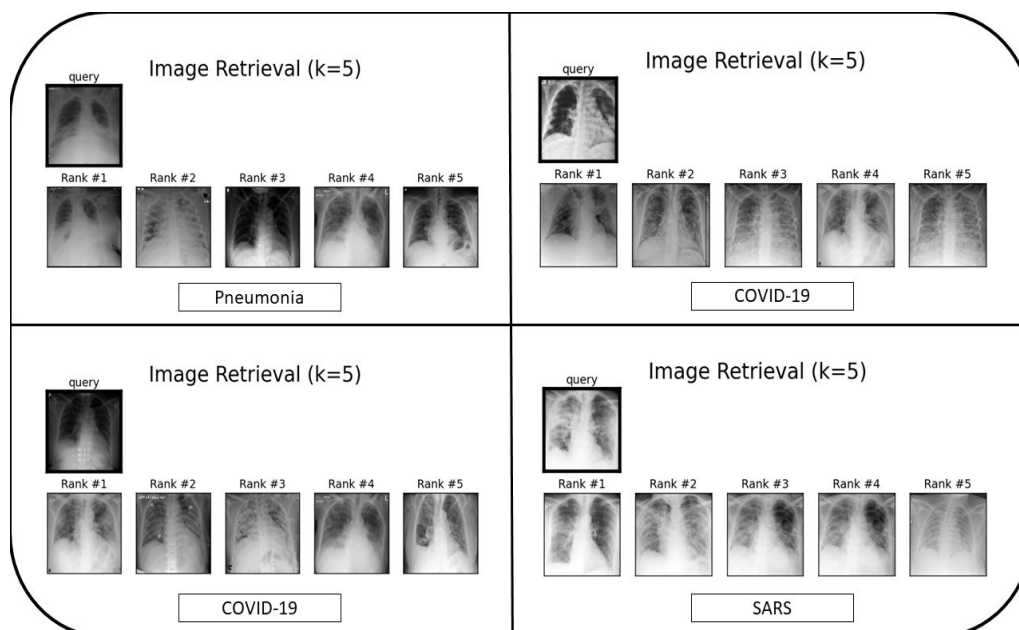


Figure 3.6: Some retrieval examples with different cases using K-nn algorithm.

In both training and test step we use the dataset without its label, and evaluate the result we add the label vector at the end.

As we can see in Figure 3.6 that represents some retrieval examples from the system, in the end, it gives similar images with the query image plus an expectation of some errors. The image reconstructed is pretty much clear even though there is some loss information.

For the COVID-19 examples the system returns an accurate result, out of 5 images 4 was the COVID-19 case, which means an accuracy equal to 80%, thus we affirm that our system became familiar with the COVID-19 images. However, for the other pneumonia type cases, the system retrieves only 2 similar images out of 5 that represent other cases, due to lack in those cases in the used dataset.

Therefore, the best performance is achieved in the COVID-19 images, and that creates an opportunity for the COVID-19 detection by using an SAE in the domain of CBIR systems including the K-nn algorithm to detect the similarity.

SAE is a perfect approach when it came to dimensionality reduction and good learning, the most important key is the dataset, and it should be relatively easy to learn more cases with more examples in the dataset. This would make the system more accurate to detect more cases efficiently.

3 Conclusion

At the end of this work, there is no doubt that using stacked auto-encoders for CBMIR is useful and make the retrieval go faster with gain of time and material resources because of the property of dimension reduction that the auto-encoder offers. Otherwise, it's still complex to set a perfect architecture of the whole process which causes the loss of relevant information, another point which is important in our study is to use a dataset with a big variety in and availability of samples of each case so the result could be more significant and accurate with all the cases presented. In addition, as an application for investigation, this system comes up very effective and convenient during this global outbreak of COVID-19 pandemic, it might be very useful for the doctors in a fast diagnosis of the virus.

Conclusion

Artificial intelligence has always proven the efficiency of its solutions regarding nowadays problems. Even in managing problems due to the fast growing and development of technology. In the medical field, managing a large number of medical images and searching for the relevant features that help medical practitioners in diagnosis and prognosis has already a given solution: the content-based image retrieval (CBIR) system. In literature, there is a large and interesting research that provides multiple frameworks that treat multiple cases in diagnosis and anomaly detection using the classical tools of machine learning.

In this master thesis, we proposed a content-based medical image retrieval (CBMIR) system dedicated to the investigation of the COVID-19 Patient-0 in order to search the first case of corona virus in an area. Since we are living this outbreak of a pandemic, we thought that proposing such application would be benefit to epidemiologists to make report about the virus spreading. In the other hand, this system could be used in further application linked to the subject of data recognition, data exploration and searching images.

In our application, we apply stacked auto-encoders for content-based medical image retrieval (CBMIR) system. SAE's are a deep learning technique known by its power of features extraction and data dimension reduction not to ignore the gain of timely processing. This method can reduce the size images in a smaller, more compact representation to enable faster image comparisons during the retrieval task. Here we can see the importance of the auto-encoder approach in a CBIR system as the result is showing quite decent in the recognition of COVID-19 images with a high accuracy of 80% assuming the capability and strength of the stacked encoders in content-based systems.

Eventually, this work will continue with future perspectives, exploring the most developed state in using varied dataset for a better discrimination; and to improve the accuracy try to combine the AE with other techniques as the GLCM to highlight the texture, or even in the level of similarity comparing by replacing the simple K-nn algorithm with a DL techniques; or then make the architecture more powerful. We can bring further applications to help medical users in diagnosis and prognosis in providing semantic features to

the image collections, application like this will offer to doctors and expert a quick diagnosis and treatment basing on previous similar cases.

Bibliography

- [1] Duan Guoyong, Yang Jing, and Yang Yilong, "Content-based image retrieval research," *Physics Procedia*, vol. 22, pp. 471–477, 2011.
- [2] Alsmadi K. Mutasem, "An efficient similarity measure for content based image retrieval using memetic algorithm," *Egyptian journal of basic and applied sciences*, vol. 4, pp. 112–122, 2017.
- [3] Vania V. Estrela and Albany E. Herrmann, "Content-based image retrieval (cbir) in remote clinical diagnosis and healthcare," in *Encyclopedia of E-Health and Telemedicine*, pp. 495–520. IGI Global, 2016.
- [4] Goran Artonovic and Dusan Starcevic, "Educational software for studying a cbir system," *International Journal of Electrical Engineering Education*, vol. 46, pp. 370–382, 2009.
- [5] A. Jenitta and R. Ravindran, "Content based geographic image retrieval using local vector pattern," *Brazilian Archives of Biology and Technology*, vol. 61, 2018.
- [6] KV. Shriram, PLK. Priyadarsini, and A. Baskar, "An intelligent system of content-based image retrieval for crime investigation," *International Journal of Advanced Intelligence Paradigms*, vol. 7, pp. 264–279, 2015.
- [7] Shashwati Mishra and Mrutyunjaya Panda, "Medical image retrieval using self-organising map on texture features," *Future Computing and Informatics Journal*, vol. 3, 2018.
- [8] Tom Weidong Cai, Jinman Kim, and David Dagan Feng, "Content-based medical image retrieval," in *Biomedical information technology*. Elsevier, 2008.
- [9] Kommineni Jenni, Satria Mandala, and Mohd Shahrizal Sunar, "Content based image retrieval using colour strings comparison," *Procedia Computer Science*, vol. 50, pp. 374–379, 2015.
- [10] Fuhui Long, Hongjiang Zhang, and David Dagan Feng, "Fundamentals of content-based image retrieval," in *Multimedia Information Retrieval and Management: Technological Fundamentals and Applications*, pp. 1–26. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.
- [11] David A. White and Ramesh C. Jain, "Similarity indexing: Algorithms and performance," in *Storage and Retrieval for Still Image and Video Databases IV*. International Society for Optics and Photonics, 1996, pp. 62–73.

- [12] Raymond T. Ng and Andishe Sedighian, "Evaluating multidimensional indexing structures for images transformed by principal component analysis," in *Storage and Retrieval for Still Image and Video Databases IV*, Ishwar K. Sethi and Ramesh C. Jain, Eds. International Society for Optics and Photonics, 1996, vol. 2670, pp. 50–61, SPIE Digital Library.
- [13] Yong Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: A power tool for interactive content-based image retrieval," *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 8, no. 5, pp. 644–655, Sept. 1998.
- [14] Jing Huang, S.Ravi Kumar, and Mandar Mitra, "Combining supervised learning with color correlograms for content-based image retrieval," in *Proceedings of the fifth ACM international conference on Multimedia*, 1997, pp. 325–334.
- [15] Yong Rui, Thomas S. Huang, and Shih-Fu Chang, "Image retrieval: Past, present, and future," *Journal of Visual Communication and Image Representation*, vol. 10, pp. 1–23, 1999.
- [16] Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain, "Content-based image retrieval at the end of the early years," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, pp. 1349–1380, 2000.
- [17] Nor Asma Mohd Zin, Rozianiwati Yusof, Saima Anwar Lashari, Aida Mustapha, Norhalina Senan, and Rosziati Ibrahim, "Content-based image retrieval in medical domain: A review," in *journal of physics: conference series*. IOP Publishing, 2018, vol. 1019.
- [18] Yong Rui, Thomas S. Huang, and Shih-Fu Chang, "Image retrieval: Current techniques, promising directions, and open issues," *Journal of visual communication and image representation*, vol. 10, pp. 39–62, 1999.
- [19] Henning Müller and Thomas M Deserno, "Content-based medical image retrieval," in *Biomedical Image Processing*. Springer, 2010.
- [20] Pranjit Das and Arambam Neelima, "An overview of approaches for content-based medical image retrieval," *International journal of multimedia information retrieval*, vol. 6, pp. 271–280, 2017.
- [21] Mansoureh Sadat Mirasadi and Amir Hossein Foruzan, "Content-based medical image retrieval of ct images of liver lesions using manifold learning," *International Journal of Multimedia Information Retrieval*, vol. 8, pp. 233–240, 2019.
- [22] P. Shamna, VK. Govindan, and KA. Abdul Nazeer, "Content-based medical image retrieval by spatial matching of visual words," *Journal of King Saud University-Computer and Information Sciences*, 2018.
- [23] Manoj Kumar and Kh. Manglem Singh, "Content based medical image retrieval system (cbmirs) to diagnose hepatobiliary images," in *Smart and Innovative Trends in Next Generation Computing Technologies*, 2018.
- [24] Aouache Mustapha, Aini Hussain, Salina Abdul Samad, and Hamzaini Abdul Hamid Mohd Asyraf Zulkifley, Wan Mimi Diyana Wan Zaki,

- “Design and development of a content-based medical image retrieval system for spine vertebrae irregularity,” *Biomedical engineering online*, vol. 14, pp. 6, 2015.
- [25] Burhan Ergen and Muhammet Baykara, “Texture based feature extraction methods for content based medical image retrieval systems,” *Bio-medical materials and engineering*, vol. 24, pp. 3055–3062, 2014.
- [26] Yogita Mistry, DT Ingole, and MD Ingole, “Content based image retrieval using hybrid features and various distance metric,” *Journal of Electrical Systems and Information Technology*, vol. 5, no. 3, pp. 874 – 888, 2018.
- [27] Ying Liu, Dengsheng Zhang, Guojun Lu, and Wei-Ying Ma, “A survey of content-based image retrieval with high-level semantics,” *Pattern recognition*, vol. 40, pp. 262–282, 2007.
- [28] Imon Banerjee, Camille Kurtz, Alon Edward Devorah, Bao Do, Daniel L. Rubin, and Christopher F. Beaulieu, “Relevance feedback for enhancing content based image retrieval and automatic prediction of semantic image features: Application to bone tumor radiographs,” *Journal of biomedical informatics*, vol. 84, pp. 123–135, 2018.
- [29] Ling Ma, Xiabi Liu, Yan Gao, Yanfeng Zhao, Xinming Zhao, and Chunwu Zhou, “A new method of content based medical image retrieval and its applications to ct imaging sign retrieval,” *Journal of Biomedical Informatics*, vol. 66, 2017.
- [30] Arden Dertat, “Applied deep learning - part 3: Autoencoders,” towardsdatascience.com, October 2017, Access–February 2020, <https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798>.
- [31] Shreya Chaudhary, “A high-level guide to autoencoders,” towardsdatascience.com, August 2019, Access–January 2020, <https://towardsdatascience.com/a-high-level-guide-to-autoencoders-b103ccd45924>.
- [32] Nathan Hubens, “Deep inside,” towardsdatascience.com, February 2018, Access–January 2020, <https://towardsdatascience.com/@nathan.hubens>.
- [33] Venkata Krishna Jonnalagadda, “Sparse, stacked and variational autoencoder,” Medium.com, December 2018, Access–January 2020, <https://medium.com/@venkatakrishtna.jonnalagadda/sparse-stacked-and-variational-autoencoder-efe5bfe73b64>.
- [34] Nathan Hubens, “Build a simple image retrieval system with an autoencoder,” towardsdatascience.com, August 2018, Access–February 2020, <https://towardsdatascience.com/build-a-simple-image-retrieval-system-with-an-autoencoder-673a262b7921>.
- [35] Francois Chollet, “Building autoencoders in keras,” blog.Keras.io, May 2016, Access–May 2020, <https://blog.keras.io/building-autoencoders-in-keras.html>.

- [36] Matthew Stewart, "Comprehensive introduction to autoencoders," towardsdatascience.com, 2019, Access-March 2020, <https://towardsdatascience.com/generating-images-with-autoencoders-77fd3a8dd368>.
- [37] Charte David, Charte Francisco, del Jesus María J, and Herrera Francisco, "An analysis on the use of autoencoders for representation learning: Fundamentals, learning task case studies, explainability and challenges," *Neurocomputing*, 2020.
- [38] Jiang, Peng, Maghrebi, Mojtaba, Crosky, Alan, Saydam, and Serkan, "Unsupervised deep learning for data-driven reliability and risk analysis of engineered systems," in *Handbook of Neural Computation*, pp. 417–431. Elsevier, 2017.
- [39] Vincent, Pascal, Larochelle, Hugo, Bengio, Yoshua, Manzagol, and Pierre-Antoine, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.
- [40] Vincent, Pascal, Larochelle, Hugo, Lajoie, Isabelle, Bengio, Yoshua, Manzagol, and Pierre-Antoine, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of machine learning research*, vol. 11, no. Dec, pp. 3371–3408, 2010.
- [41] Verma, Nishchal K, Gupta, Vishal Kumar, Sharma, Mayank, Sevakula, and Rahul Kumar, "Intelligent condition based monitoring of rotating machines using sparse auto-encoders," in *IEEE Conference on Prognostics and Health Management (PHM)*. IEEE, 2013, pp. 1–7.
- [42] Zeng, Nianyin, Zhang, Hong, Song, Baoye, Liu, Weibo, Li, Yurong, Dobaie, and Abdullah M, "Facial expression recognition via learning deep sparse autoencoders," *Neurocomputing*, vol. 273, pp. 643–649, 2018.
- [43] Jaskari, Joel, Kivinen, and Jyri J, "A novel variational autoencoder with applications to generative modelling, classification, and ordinal regression," *arXiv preprint arXiv:1812.07352*, 2018.
- [44] Simonovsky, Martin, Komodakis, and Nikos, "Graphvae: Towards generation of small graphs using variational autoencoders," in *International Conference on Artificial Neural Networks*. Springer, 2018, pp. 412–422.
- [45] Liu, Qi, Allamanis, Miltiadis, Brockschmidt, Marc, Gaunt, and Alexander, "Constrained graph variational autoencoders for molecule design," in *Advances in neural information processing systems*, 2018, pp. 7795–7804.
- [46] Walker, Jacob, Doersch, Carl, Gupta, Abhinav, Hebert, and Martial, "An uncertain future: Forecasting from static images using variational autoencoders," in *European Conference on Computer Vision*. Springer, 2016, pp. 835–851.

- [47] Deep Learning on Medium, "A beginners guide to build stacked autoencoder and tying-weights with it," MC.AI.com, December 2019, Access–February 2020, <https://mc.ai/a-beginners-guide-to-build-stacked-autoencoder-and-tying-weights-with-it/>.
- [48] Janod, Killian, Morchid, Mohamed, Dufour, Richard, Linares, Georges, De Mori, and Renato, "Deep stacked autoencoders for spoken language understanding," 2016, vol. 1, p. 2.
- [49] Bao, Wei, Yue, Jun, Rao, and Yulei, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PloS one*, vol. 12, no. 7, 2017.
- [50] Kannadasan, K, Edla, Damodar Reddy, Kuppili, and Venkatanaresbabu, "Type 2 diabetes data classification using stacked autoencoders in deep neural networks," *Clinical Epidemiology and Global Health*, vol. 7, no. 4, pp. 530–535, 2019.
- [51] S John Berlin, S Hari, and Anitha Rajam, "An effective approach to shadow removal and skin segmentation using deep neural networks," *IJIRST- International Journal for Innovative Research in Science & Technology*, vol. 4, May 2018.
- [52] Camlica, Zehra, Tizhoosh, Hamid R, Khalvati, and Farzad, "Autoencoding the retrieval relevance of medical images," in *International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, 2015, pp. 550–555.
- [53] Tizhoosh, Hamid R, Mitcheltree, Christopher, Zhu, Shujin, Dutta, and Shamak, "Barcodes for medical image retrieval using autoencoded radon transform," in *23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 3150–3155.
- [54] S. Sharma, I. Umar, L. Ospina, D. Wong, and Hamid R. Tizhoosh, "Stacked autoencoders for medical image search," *CoRR*, vol. abs/1610.00320, 2016.
- [55] Alex Krizhevsky and Geoffrey E. Hinton, "Using very deep autoencoders for content-based image retrieval.," in *ESANN*, 2011.
- [56] Mohd Javaid, Abid Haleem, Raju Vaishya, Shashi Bahl, Rajiv Suman, and Abhishek Vaish, "Industry 4.0 technologies and their applications in fighting covid-19 pandemic," *Diabetes & metabolic syndrome*, vol. 14, no. 4, pp. 419–422, Apr. 2020.
- [57] Janice C. Sipior, "Considerations for development and use of ai in response to covid-19," *International Journal of Information Management*, p. 102170, 2020.
- [58] K. C. Santosh, "Ai-driven tools for coronavirus outbreak: Need of active learning and cross-population train/test models on multitudinal/multimodal data," *Journal of Medical Systems*, vol. 44, no. 5, pp. 93–, 2020.
- [59] OECD Policy Responses to Coronavirus (COVID-19), "Using artificial intelligence to help combat covid-19," oecd.org/coronavirus, April 10 Access–July 2020,

<https://www.oecd.org/coronavirus/policy-responses/using-artificial-intelligence-to-help-combat-covid-19-ae4c5c21/>.

- [60] Wim Naudé, “Artificial intelligence vs covid-19: limitations, constraints and pitfalls,” *Ai & Society*, pp. 1–5, 2020.
- [61] Alom, Md Zahangir, Rahman, MM, Nasrin, Mst Shamima, Taha, Tarek M, Asari, and Vijayan K, “Covid_mtnet: Covid-19 detection with multi-task deep learning approaches,” *arXiv preprint arXiv:2004.03747*, 2020.
- [62] Alom Md Zahangir, Aspiras, Theus, Taha, Tarek M, Asari, and Vijayan K, “Skin cancer segmentation and classification with nbla-n and inception recurrent residual convolutional networks,” *arXiv preprint arXiv:1904.11126*, 2019.
- [63] Ozturk, Tulin, Talo, Muhammed, Yildirim, Eylul Azra, Baloglu, Ulas Baran, Yildirim, Ozal, Acharya, and U Rajendra, “Automated detection of covid-19 cases using deep neural networks with x-ray images,” *Computers in Biology and Medicine*, pp. 103–792, 2020.
- [64] Adrian Rosebrock, “Detecting covid-19 in x-ray images with keras, tensorflow, and deep learning,” pyimagesearch, March 16 Access–July 2020, <https://www.pyimagesearch.com/2020/03/16/detecting-covid-19-in-x-ray-images-with-keras-tensorflow-and-deep-learning/>.
- [65] A S Albahri, Rula A Hamid, Jwan K Alwan, Z T Al-Qays, A A Zaidan, B B Zaidan, A O S Albahri, A H AlAmoodi, Jamal Mawlood Khlaf, E M Almahdi, Eman Thabet, Suha M Hadi, K I Mohammed, M A Alsalem, Jameel R Al-Obaidi, and H T Madhloom, “Role of biological data mining and machine learning techniques in detecting and diagnosing the novel coronavirus (covid-19): A systematic review,” *Journal of Medical Systems*, vol. 44, no. 7, pp. 122–122, 2020.
- [66] Toğaçar, Mesut, Ergen, Burhan, Cömert, and Zafer, “Covid-19 detection using deep learning models to exploit social mimic optimization and structured chest x-ray images using fuzzy color and stacking approaches,” *Computers in Biology and Medicine*, vol. 121, pp. 103805, 2020.
- [67] Akhtar, Mahmood, Kraemer, Moritz UG, Gardner, and Lauren M, “A dynamic neural network model for predicting risk of zika in real time,” In *BMC medicine* [1], p. 171.
- [68] Andy Chun, “In a time of coronavirus, chinas investment in ai is paying off in a big way,” *South China Morning Post*, vol. 18, 2020.
- [69] Carter Maslan, “Social distancing detection for covid-19,” <https://blog.camio.com/>, March 30 Access–July 2020, <https://blog.camio.com/social-distancing-detection-for-covid-19-a5c9e8d6c760>.
- [70] Bogoch, Isaac I, Watts, Alexander, Thomas-Bachli, Andrea, Huber, Carmen, Kraemer, Moritz UG, Khan, and Kamran, “Potential for global spread of a novel coronavirus from china,” *Journal of travel medicine*, vol. 27, no. 2, 2020.

- [71] Wim Naudé, “Artificial intelligence against covid-19: An early review,” IZA Discussion Paper 13110, Institute of Labor Economics (IZA), Bonn, Apr 2020.
- [72] Guan, Wei-jie, Ni, Zheng-yi, Hu, Yu, Liang, Wen-hua, Ou, Chun-quan, He, Jian-xing, Liu, Lei, Shan, Hong, Lei, Chun-liang, Hui, David SC, et al., “Clinical characteristics of 2019 novel coronavirus infection in china,” *MedRxiv*, 2020.
- [73] Lei, Junqiang, Li, Junfeng, Li, Xun, Qi, and Xiaolong, “Ct imaging of the 2019 novel coronavirus (2019-ncov) pneumonia,” *Radiology*, vol. 295, no. 1, pp. 18–18, 2020.
- [74] Zu, Zi Yue, Jiang, Meng Di, Xu, Peng Peng, Chen, Wen, Ni, Qian Qian, Lu, Guang Ming, Zhang, and Long Jiang, “Coronavirus disease 2019 (covid-19): a perspective from china,” *Radiology*, p. 200490, 2020.
- [75] Stoecklin, Sibylle Bernard, Rolland, Patrick, Silue, Yassoungo, Mailles, Alexandra, Campese, Christine, Simondon, Anne, Mechain, Matthieu, Meurice, Laure, Nguyen, Mathieu, Bassi, Clément, et al., “First cases of coronavirus disease 2019 (covid-19) in france: surveillance, investigations and control measures,” *Eurosurveillance*, vol. 25, no. 6, 2020.
- [76] Tabik Siham, Gómez-Ríos Anabel, Martín-Rodríguez JL, Sevillano-García I, Rey-Area Manuel, Charte D, Guirado Emilio, Suárez JL, Luengo Julián, Valero-González MA, et al., “Covidgr dataset and covid-sdnet methodology for predicting covid-19 based on chest x-ray images,” *arXiv preprint arXiv:2006.01409*, 2020.
- [77] Joseph Paul Cohen, Paul Morrison, Lan Dao, Karsten Rothand Tim Q Duong, and Marzyeh Ghassemi, “Covid-19 image data collection: Prospective predictions are the future,” *arXiv 2006.11988*, 2020.

Abstract

COVID-19 is a recently discovered infectious disease caused by the coronavirus known to cause respiratory infections in humans. This pandemic is spreading rapidly around the world, causing multiple damages in different areas. In this graduation project, we are interested in the recognition of this disease using medical images. For this purpose, we present an application dedicated to epidemiologists for the investigation of the Patient 0 infected and establish the propagation path in different areas of the country. A Content Based Medical Image Retrieval (CBMIR) system based on stacked-encoder networks is proposed, our model is dedicated to search for target COVID Chest X-Ray images using similarity measurements learned through an image database of different pathologies as SARS and other viral or bacterial species of pneumonia diseases.

Keywords: Content based image retrieval, Stacked autoencoders, COVID-19, investigation, recognition, X-rays medical images, Features extraction.

Résumé

La COVID-19 est une maladie infectieuse récemment découverte causée par le corona virus connu par la provocation des infections respiratoires chez l'humain. Cette pandémie se propage rapidement dans le monde, causant de multiples dommages dans différents secteurs. Dans ce projet de fin d'études, nous nous intéressons à la reconnaissance de cette maladie à l'aide d'images médicales. Pour ce faire, nous présentons une application dédiée aux épidémiologistes pour l'investigation du patient-0 contaminé et établir le chemin de propagation dans les différentes zones du pays. A ces fins, nous proposons un système de recherche d'image médicale par son contenu CBMIR (Content Based Medical Image Retrieval) se basant sur les réseaux d'encodeurs en cascades qui sont dédié à la recherche d'images COVID Chest X-Ray. Notre modèle se base sur les mesures de similarité apprises au travers d'une base d'images regroupant différentes pathologies comme le SRAS et d'autres espèces virales ou bactériennes de maladies de pneumonie.

Mots clés: Recherche d'image par son contenu, Auto-encodeurs en cascades, COVID-19, investigation, reconnaissance, images à rayons X médicales, extractions des caractéristiques.