Visual Semantic Embedding Model based on DeViSE for Medical imaging

Ludgero da Silva Diogo

November 2020

Ludgero da Silva Diogo

Visual Semantic Embedding Model based on DeViSE for Medical imaging

December 2020

Universidade do Minho
Escola de Engenharia

Ludgero Diogo

Visual Semantic Embedding Model based on DeViSE for Medical imaging

Master's dissertation submitted to obtain the degree of

Master's in Informatics Engineering

(Portugal)

Work performed under the supervision of

Victor Alves

2020

DECLARATION

Full name: Ludgero da Silva Diogo

Email address: pg38417@alunos.uminho.pt

Dissertation title: Visual Semantic Embedding Model based on DeViSE for medical imaging

Supervisor: Victor Manuel Rodrigues Alves

Finish year: 2020

Master's degree: Master's in Informatics Engineering

I declare that I grant to the University of Minho and its agents a non-exclusive license to file and make available through its repository, in the conditions indicated below, my dissertation, as a whole or partially, in digital support.

I declare that I authorize the University of Minho to file more than one copy of the dissertation and, without altering its contents, to convert the dissertation to any format or support, for the purpose of preservation and access.

Furthermore, I retain all copyrights related to the dissertation and the right to use it in future works.

I authorize the partial reproduction of this dissertation for the purpose of investigation by means of a written declaration of the interested person or entity.

This is an academic work that can be used by third parties if internationally accepted rules and good practice with regard to copyright and related rights are respected.

Thus, the present work can be used under the terms of the license indicated below.

In case the user needs permission to be able to make use of the work in conditions not foreseen in the indicated licensing, he should contact the author through the RepositóriUM of the University of Minho.

Atribuição-NãoComercial-SemDerivações
CC BY-NC-ND

https://creativecommons.org/licenses/by-nc-nd/4.0/

Universidade do Minho, December 2, 2020

Signature: Ludgero Diogo

ACKNOWLEDGEMENTS

I would like to thank my supervisor Victor Alves, who provided consistent support when needed. I truly appreciate the independence and flexibility that was granted regarding the approach of this master's dissertation. The access to the server provided by him definitely made the research much more convenient. I also would like to thank my friends and family who supported and help me throughout my studies, as well in my personal life, and were by my side in both in hard and easy times, given me the strength and confidence to finish my studies. A special thanks to my parents who were able to provide me with an environment where I could focus on my studies at 100%.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

ABSTRACT

During the last decades, artificial intelligence algorithms have been evolving to the point that they can achieve some amazing results like, identify and navigate roads, identify fraudulent transactions, personalize crops to individual conditions, discover new consumer trends, predict personalized health outcomes, optimize merchandising strategies, predict maintenance, optimize pricing and scheduling in real-time, diagnose diseases, among many others.

However, although it can do all of that, it needs all the data to be correctly label, in other words, it can not, for example, diagnose a disease, such as a stroke, if it does not know what a stroke is, so if the algorithm has never been trained to identify strokes a new algorithm has to be created or the current one has to be retrained, similar issues happen in the other examples.

This work focuses on this problem and tries to solve it by using a related in a high dimensional vector space, called semantic space, where the knowledge from known classes can be transferred to unknown classes.

Key Words: Machine Learning, DeViSE, Medical Imaging, Semantic Model

Resumo

Durante as últimas décadas, os algoritmos de inteligência artificial têm evoluído ao ponto de alcançarem resultados incríveis, como identificar e navegar estradas, identificar transações fraudulentas, personalizar colheitas para condições individuais, descobrir novas tendências de consumo, prever resultados de saúde personalizados, otimizar merchandising estratégias, prever manutenções, otimizar preços e agendamentos em tempo real, diagnosticar doenças, entre muitos outros.

Porém, embora possa fazer tudo isso, precisa que todos os dados sejam identificados corretamente, ou seja, não pode, por exemplo, diagnosticar uma doença, como um acidente vascular cerebral, se não souber o que é um AVC, portanto, se o algoritmo nunca foi treinado para identificar AVC's um novo algoritmo precisa de ser criado ou o atual de ser retreinado, problemas semelhantes acontecem nos outros exemplos.

Esta tese foca-se neste problema e tenta resolvê-lo usando um espaço vetorial relacionado de alta dimensão, denominado espaço semântico, onde o conhecimento de classes conhecidas pode ser transferido para classes desconhecidas.

Palavras Chave: Aprendizagem Máquina, DeViSE, Imagem Médica, Modelo Semântico

TABLE OF CONTENTS

List of Figures

LIST OF TABLES

List of Abbreviations

A

AI	Artificial Intelligence
ANN	Artificial Neural Network

C

CNN	Convolutional Neural Network
CT	Computerized Tomography

D

DeViSE	Deep Visual-Semantic Embedding Model
DL	Deep Learning
DSR	Design Science Research

L

LR	Learning Rate

M

MAE	Mean Absolute Error
ML	Machine Learning
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error

N

NN	Neural Networks

R

ReLU	Rectified Linear Unit
RL	Reinforcement Learning
RMS	Root Mean Square
RMSProp	Root Mean Square Propagation
RNN	Recurrent Neural Networks

S

Sarsa	State–Action–Reward–State–Action

SGD Stochastic Gradient Descent

Z

ZSL Zero Shot Learning

GLOSSARY

Activation Function	In the context of Artificial Neural Networks, an activation function defines the output of a node, using the result of a linear combination of the input(s).
Adaptive Pooling	Pooling method where the output size is specified and the stride and kernel size are automatically selected to adapt to those needs, instead of the other way around.
AdaptiveAvgPool2d	Pytorch class that applies a 2D adaptive average pooling over an input signal composed of several input planes
Artificial Intelligence (AI)	Artificial intelligence is intelligence, produced by machines rather than by humans or other animals. Often it refers to the application of cognitive functions that humans associate with the demonstration of intelligence. E.g. being able to analyze its environment, solving problems, and learning.
Artificial Neural Networks (ANN)	Inspired by biological Neural Networks, an artificial Neural Network is built around a collection of nodes that interact. It aims to learn and adapt in order to perform a certain task, without having been programmed with task-specific rules. In order to do so, the Neural Network provides one or multiple outputs, formed through an interaction of nodes of which at least one is provided with an input.
Average Pooling	A sample-based discretization process, that down-sample an input representation, reducing its dimensionality by getting the average value of the features in the pool within the feature map and allowing for assumptions to be made about features contained in the sub-regions binned.
Batch Size	The number of samples that will be propagated through the network.
Batch Normalization	A technique to standardize the inputs to a network, applied to either the activations of a prior layer or inputs directly, it accelerates training, in some cases by halving the epochs or better, and provides some regularization, reducing generalization error.
Class	In the context of classification, the possible outputs of a classifier are commonly limited by a discrete number of classes. Each of those classes represents one possible output category that the classifier could associate when given the corresponding input.
Classifier	A classifier aims to map a given input to a certain output, using the characteristics of the input that correspond to the most likely output. It

	focuses on the differences between distinctive classes while neglecting the similarities between those classes.
Convolutional Neural Network (CNN)	A Convolutional Neural Network, commonly used in the context of visual imagery analysis, is a class of a deep artificial Neural Network. It is based on the animal visual cortex, working in a similar fashion as how humans or other animals process visual imagery. As opposed to fully connected Neural Networks, its hidden layers include convolutional layers, which tend to downsize the inputs for each neuron from each convolutional layer. This class of Neural Networks is proven to be effective to discover patterns in the input that are not necessarily spatially at the same location.
Dataset	A dataset is a collection of data samples.
Deep Learning (DL)	Deep learning is a subset of Machine Learning methods based on feature learning. As opposed to task-specific algorithms in which be provided with a specific strategy on how to use the features to obtain the requested outcome, Deep Learning methods aim to learn from the provided features.
Dropout	In the context of Neural Networks, dropout is a technique that attempts to reduce overfitting. When being configured in a certain layer, it drops out a certain percentage (for example 25%) of randomly selected units in that respective layer.
Feature	Features are the distinctive attributes of something that make it different from other things.
Feedforward Neural Network	In a feedforward Neural Network, no loops can be made within the network, i.e. every layer takes only input from previous layers.
Fully Connected Neural Network	In a fully connected Neural Network, each neuron with the exception of the input neurons is connected to all neurons from the previous layer.
Kernel Size	Is the size of the pool
Machine Learning (ML)	Machine Learning is a field, part of artificial intelligence, in which learning techniques are used that use statistical information from data in order to get better at the given task.
Max Pooling	A sample-based discretization process, that down-sample an input representation, reducing its dimensionality by getting the feature with the maximum value in the pool within the feature map and allowing for assumptions to be made about features contained in the sub-regions binned.
MaxPool2d	Pytorch class that applies a two-dimensional max pooling over an input signal composed of several input planes.

Overfitting Overfitting is the tendency of a classifier to use information from the provided features that are not contributing to a better overall performance with other data than the one that is used for training. This could lead to very high accuracies with the training data, while the performance of the classifier declines with any other data (e.g. the validation data or test data). Providing a big amount of training data could greatly reduce the tendency to overfit, as well as stopping the training before overfitting starts to occur. Other techniques include the introduction of dropout or pooling.

Plane A hyperdimensional segmentation that allows for segregation and visualization of functional layers.

Stride Hyperparameter of ANN and CNN that defines how much the pool moves per iteration.

Test dataset A test dataset is a set of samples that are used to evaluate a model that has been fit on training data. The samples in this set are supposed to be distinctive from the samples in the training set.

Training dataset The training dataset is a set of samples that are used to configure the parameters of a model. In the context of supervised learning, a set of input-output pairs is provided, each associating a certain input with the desired output. The model aims to find the best way to obtain the desired output when given the corresponding input.

Validation dataset A validation dataset is a set of samples, distinctive from the training data, that can be used to tune the architecture of a classifier. As opposed to the accuracy of the training dataset, the validation dataset could indicate when overfitting started to occur. It is also used to compare multiple classifier architectures (e.g. different amounts of hidden layers in a Neural Network).

Zero Shot Learning A form of extending supervised learning to a setting of solving for example a classification problem when not enough labeled examples are available for all classes.

1 Introduction

The present chapter is about:

- Context and Motivation: how medical imaging works and it's limitations.
- Objectives: develop a model capable of identifying disorders present in medical images regardless if the model as train to identify them or not.
- Investigation Methodology: how the solution was reached.
- Structure of the dissertation: how the dissertation is structured.

1.1 CONTEXT AND MOTIVATION

Nowadays, medical images such as Computed Tomography (CT), X-Rays, or Magnetic Resonance Imaging (MRI) are widely used to diagnose, plan and guide the treatment and monitoring of disease progression [1]. Image registration techniques, i.e., methods of aligning images on a computer model or aligning features in an image with locations in physical space are part of the clinical routine. They are the core of the interpretation and analysis of the medical image to connect the spatial information and the pathologies or anomalies. The image registration allows the combination of complementary information provided by different modalities, intermodality, or by the same modality, intramodality. It can also be used to align multiple images captured from the same subject, intrasubject registration, or from different subjects, intersubject registration. However, image manipulation is only possible with digital images consisting of picture elements, also referred to as pixels, which are small rectangular arrays with an associated image intensity value. They provide the coordinate system of the image and allow access to each element in the image through its respective two-dimensional position on the array. The 3D volumes are created when captured directly or when 2D images are stacked. Each pixel is now a small volume element, also called a voxel. To each voxel is also assigned an intensity that results from an average of a physical attribute measure over that volume [1].

One of the most commonly used 2D images are X-Rays because sensitive results can be obtained, and radiological devices are widely available [2]. The X-Rays allow visualization within the body due to the different densities of tissues. When the density is higher, the X-rays are effectively blocked, and different shadows are created for the tissue [3]. The interpretation by the radiologist, however, can be a challenge because it depends on his experience and a clear mind. Since the radiologist must work intensively, this misdiagnosis might increase due to exhaustion. There is also a lack of specialized physicians, mainly in the least developed areas [4]. For this reason, interest in combining X-Rays images with Deep Learning (DL) methods has increased over the years [5] [6] [7] [8]. DL techniques have also been studied in 3D

images, such as MRI images, which has providing promising results [9] [10] [11] [12]. MRI provides variable image contrast and high contrast for soft tissue (i.e., tissues that support other structures and organs of the body, such as tendons, fibrous tissues, fat, muscles, nerves, and blood vessels), and high spatial resolution [13]. The quantitative analysis of brain MRI is often used to study brain disorders, e.g. to quantify tissue atrophy by segmentation and corresponding measurements of brain tissue [14]. Manual segmentation requires experienced radiologists to analyze the MRI information, which takes a long time because they have to analyze several slices of the image. Manual segmentation is also subjective depending on the radiologist. It has been found that DL segmentation and classification techniques are useful for improved diagnosis, growth rate prediction, treatment planning, and treatment costs because their algorithms perform well [15] [16] [17] [18]. However, these algorithms can only work to identify disorders that they have been trained to recognize, so if the algorithm tries to identify, in this case, a disorder that it has never been trained for, it will fail. In Machine Learning, this is considered as the problem of zero-shot learning (ZSL) [19], this work tries to solve this problem through the use of a model based on Deep Visual Semantic Embedding (DeViSE).

1.2 OBJECTIVES

The objective of this dissertation is to develop a semantic visualization model capable of correctly identify disorders present in various medical images, regardless if the algorithm has been trained to identify such disorders or not, using DeViSE.

1.3 INVESTIGATION METHODOLOGY

Prior to developing this solution, Visual Semantic Embedding Model, careful and thorough research is required to analyze and select the appropriate development tools, methods, and research strategies to use to develop the solution. Therefore, the research strategy selected for this dissertation was Design Science Research (DSR), which is considered to be effective in computer science projects. The DSR methodology is a rigorous scientific research methodology that includes the techniques, principles, and procedures that must be followed to design and develop successful solutions to the problems [20] [21]. It can be divided into six main steps:

1) Problem identification and motivation, Chapter 1 - Identification of the research problems to be solved, and appropriate justification of the practical and scientific values of the solution.

2) Definition of the solution's objectives, Chapter 3 – Identification and presentation of the solution objectives of the solution based on the identified research problems.
3) Design and development of the solution, Chapter 3 – Design the solution, i.e. definition of the features and architecture, and then the development of the solution to solve the specified objectives.
4) Demonstration, Chapter 3 – Experiments and simulations to demonstrate the solutions developed, in order to evaluate the capacity of the solutions to solve the research problems.
5) Evaluation, Chapter 3 – Assessment of the solution and comparison of the results achieved in the demonstration step with the objective defined in the second step. This means assessing whether the solution worked out can support the research problems and the objectives set.
6) Communication, Chapter 4 – Communication of the problem as the developed solution to an audience, i.e., spread the importance of the problem and of the solution.

1.4 STRUCTURE OF THE DISSERTATION

In the next chapter, the technologies and concepts that are relevant to this research are documented. The chapter discusses the existing knowledge on which this study is built. When the reader of this dissertation is not familiar with ZSL, Neural Networks, and/or Reinforcement Learning, the chapter explains the necessary concepts in order to understand later chapters. It also contains and discusses the materials and methodology of this stud. It explains the presented model, as well as the different aspects of the built framework that was used for this study.

In chapter 3 results of the used methods are displayed and discussed. Rather than solely showing the results of the complete model, results are presented multiple times in order to prove the value of the individual components.

In chapter 4 conclusions are made, based on the obtained results of chapter 3.

2 Technologies and Concepts

The present chapter is about:

- Medical Imaging: how different medical imaging techniques work and the disorders that are able to detect.
- Imaging Preprocessing: what is image preprocessing and the different image augmentation techniques.
- Machine Learning: the different machine learning techniques used and their strong points
- State of the Art: one of the problems with today's image classification techniques, Zero Shot Learning.
- DeViSE Architecture: a solution to Zero Shot Learning, DeViSE, and it's architecture and limitations.

2.1 MEDICAL IMAGING

Medical imaging is the technique and process of creating visual representations of the interior of a body for clinical analysis and medical intervention, as well as visual representation of the function of some organs or tissues. Medical imaging seeks to reveal internal structures hidden by the skin and bones, as well as to diagnose and treat disease. Medical imaging also establishes a database of normal anatomy and physiology to make it possible to identify abnormalities. This process is widely used worldwide since that in 2005 there were 5 billion medical imaging studies conducted worldwide [22]. There are several types of medical imaging modalities, e.g. X-Ray, Computed Tomography or CT scan, magnetic resonance imaging, also called MRI, this study will focus on X-Ray and CT scan.

2.1.1 X-RAY

An X-Ray is a quick exam that produces images of the structures inside a person's body, particularly bones. X-Ray beams pass through the body and are absorbed in different amounts depending on the density of the material they pass through, dense materials such as metal or bones show up white, while less dense materials such as the air in the lungs show up black, or in shades of grey like fat and muscle. An example of an X-Ray can be seen in Figure 2.1.

Figure 2.1 – X-Ray of a human chest

X-Ray technology can be used to examine several parts of the body and detect several disorders such as:

- Bones and teeth
    - Fractures and infections. In most cases, fractures and infection in bones and teeth show up clearly.
    - Arthritis. X-Rays of the joints can reveal evidence of arthritis.
    - Dental decay. Dentists use X-Rays to check for cavities in the teeth.
    - Osteoporosis. Special types of X-Ray tests can measure bone density.
    - Bone cancer. X-Rays can reveal bone tumors.
- Chest
    - Lung infection and conditions. Evidence of pneumonia, tuberculosis, or lung cancer can show up on X-Rays.
    - Breast cancer. Mammography is a special type of X-Ray test used to examine breast tissue.
    - Enlarged heart. This sign of congestive heart failure shows up clearly.
    - Blocked blood vessels. Injecting a contrast material that contains iodine can help. highlight sections of the circulatory system to make them visible on X-Rays.
- Abdomen
    - Digestive tract problems. Barium, a contrast medium delivered in a drink or an enema, can help reveal problems in the digestive system.
    - Swallowed items. If an object is swallowed an X-Ray can show the location of that object.

2.1.2 COMPUTED TOMOGRAPHY

A computerized tomography scan combines a series of X-Ray images taken from different angles around your body and uses computer processing to create cross-sectional images of the bones, blood vessels, and soft tissues inside the body. CT scan images provide more detailed information than plain X-Rays do. An example of a slice of a CT scan can be seen in Figure 2.2.

Figure 2.2 – Slice of a CT scan of a human chest

A CT scan can be used to visualize nearly all parts of the body and can help:

- Diagnose muscle and bone disorders, such as bone tumors and fractures.
- Pinpoint the location of a tumor, infection, or blood clot.
- Guide procedures such as surgery, biopsy, and radiation therapy.
- Detect and monitor diseases and conditions such as cancer, heart disease, lung nodules, and liver masses.
- Monitor the effectiveness of certain treatments, such as cancer treatment.
- Detect internal injuries and internal bleeding.

2.1.3 MAGNETIC RESONANCE IMAGING

MRI is a medical imaging technique that uses a magnetic field and computer-generated radio waves to create detailed images of the organs and tissues in the body. When inside of an MRI machine, the magnetic field temporarily realigns water molecules in the body, radio waves cause these aligned atoms to produce faint signals, which are used to create cross-sectional MRI images, like slices in a loaf of bread, the MRI machine can also produce 3D images that can be viewed from different angles. MRI is a way for doctors to examine organs, tissues, and skeletal system. An example of a slice of an MRI can be seen in Figure 2.3.

Figure 2.3 – Slice of an MRI of a human chest, retrieved from [23]

It produces high-resolution images of the inside of the body that help diagnose a variety of problems, such as:

- MRI of the brain and spinal cord.
    - Aneurysms of cerebral vessels.
    - Disorders of the eye and inner ear.
    - Multiple sclerosis.
    - Spinal cord disorders.
    - Stroke.
    - Tumors.
    - Brain injury from trauma.
- MRI of the heart and blood vessels can assess:

  - Size and function of the heart's chambers.
  - Thickness and movement of the walls of the heart.
  - The extent of damage caused by heart attacks or heart disease.
  - Structural problems in the aorta, like aneurysms or dissections.
  - Inflammation or blockages in the blood vessels.
- MRI of other internal organs can check for tumors or other abnormalities of many organs in the body, including the following:
  - Liver and bile ducts.
  - Kidneys.
  - Spleen.
  - Pancreas.
  - Uterus.
  - Ovaries.
  - Prostate.
- MRI of bones and joints can help evaluate:
  - Joint abnormalities caused by traumatic or repetitive injuries, such as torn cartilage or ligaments.
  - Disk abnormalities in the spine.
  - Bone infections.
  - Tumors of the bones and soft tissues.
- MRI of the breasts can be used with mammography to detect breast cancer, particularly in women who have dense breast tissue or who might be at high risk of the disease.

Image Preprocessing

In order to classify an image, there are four steps to follow, first step is image preprocessing, the aim of this is to improve the image data, also known as features, by suppressing unwanted distortions and enhancing some important image features so that Computer Vision models can benefit from this improved data to work on. The second step is the detection of the object, detection refers to the localization of the object, which means the segmentation of the image and identifying the position of the object of interest. The next step is the feature extracting and training, this is a crucial step wherein statistical or Deep Learning methods are used to identify the most interesting patterns of the image,

features that might be unique to a particular class and that will, later on, help the model to differentiate between different classes. This process where the model learns the features from the dataset is called model training. The last step is the classification of the object, this step categorizes detected objects into predefined classes by using a suitable classification technique that compares the image patterns with the target patterns [24].

From these steps, the first is arguably the most crucial and important one, since it is how the computer "sees" the image, computers are able to perform computations on numbers, however, they are not able to interpret images in the way that humans do. The image has to somehow be converted to numbers for the computer to understand it. This process can be further divided into three sub-process, the first step is reading the image, in this step, simply store the path to the image dataset into a variable and then create a function to load folders containing images into arrays so that computers can deal with it. The next step is image resize, some images captured by a camera and fed to AI algorithms might vary in size, therefore, a base size should be established for all the images fed into the AI algorithms by resizing them. The last step is data augmentation, this is a way of creating new 'data' with different orientations. The benefits of this are two-fold, the first being the ability to generate 'more data' from limited data and secondly, it prevents overfitting, in other words, the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict reliably future observations. There are lots of data augmentation techniques, the most used ones are grey scaling, reflection or flip, gaussian blur, histogram equalization, rotation, and translation [24].

Image Augmentation Techniques

Grey scaling, where the image will be converted to greyscale, range of gray shades from white to black, the computer will assign each pixel a value based on how dark it is. All the numbers are put into an array and the computer does computations on that array. An example of grey scaling can be seen in Figure 2.4.

Figure 2.4 – Example of grey scaling

Reflection or flip, flip images horizontally and vertically. Example in Figure 2.5.

Figure 2.5 – Example of a vertical flip

Gaussian blur, also known as Gaussian smoothing, is the result of blurring an image by a gaussian function. It is a widely used effect in graphics software, typically to reduce image noise. An example of Gaussian blur can be seen in Figure 2.6.

Figure 2.6 – Example of Gaussian blur with 20-pixel radius

Histogram equalization, Histogram equalization is another image processing technique to increase the global contrast of an image using the image intensity histogram. This method needs no parameter, but it sometimes results in an unnatural looking image. An example of histogram equalization can be seen in Figure 2.7.

Figure 2.7 – Example of histogram equalization

Rotation is yet another image augmentation technique. Rotating an image might not preserve its original dimensions depending on what angle you choose to rotate it with. An example of rotation can be seen in Figure 2.8.

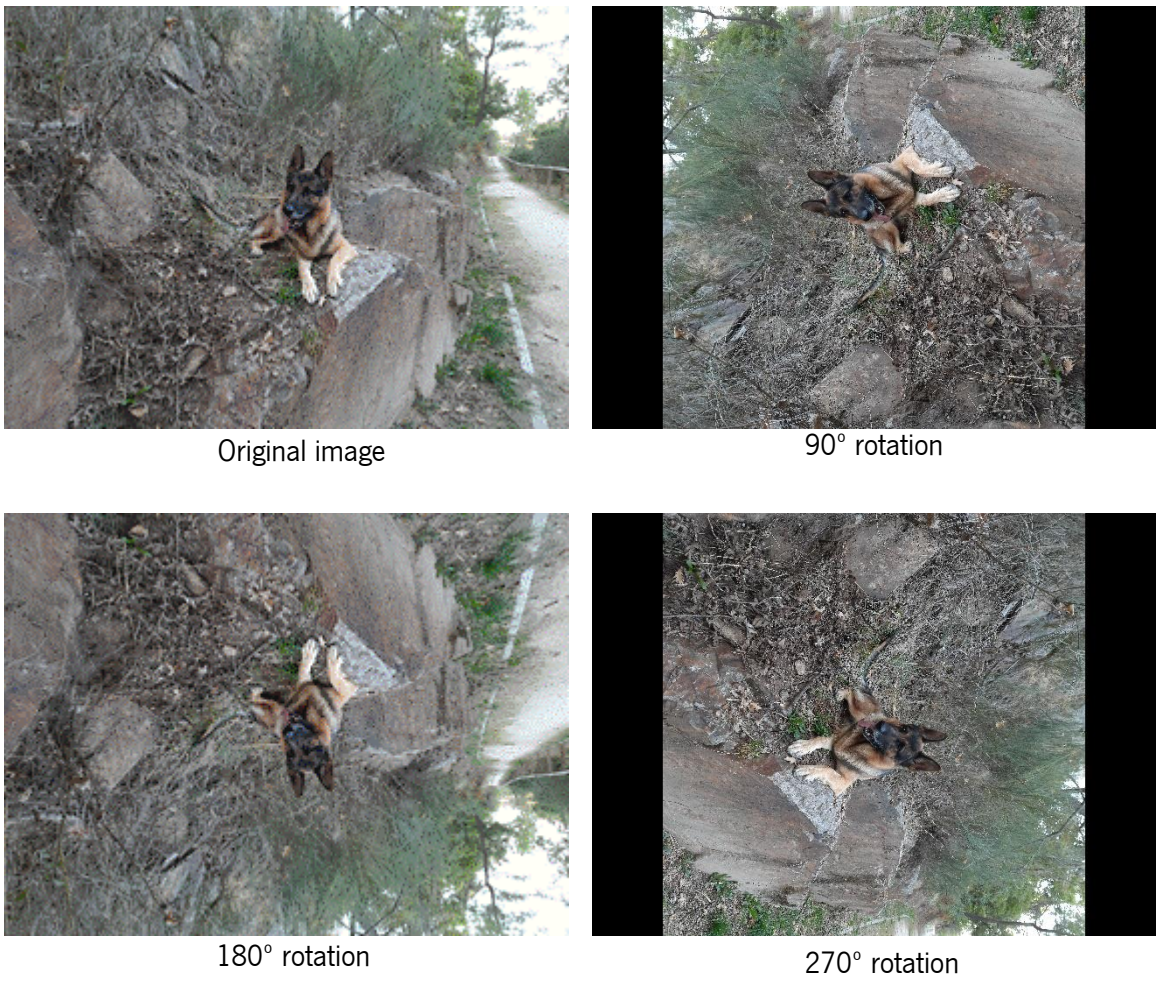

Figure 2.8 – Example of an image being rotated

Translation, translation just involves moving the image along the X or Y direction or both. This method of augmentation is very useful as most objects can be located almost anywhere in the image. This forces our feature extractor to look everywhere. An example of image translation can be seen in Figure 2.9.

Figure 2.9 – Example image translation

2.2 MACHINE LEARNING

After translating an image to numbers an image classification technique must be implemented in order to classify the images, there are a few techniques used like supervised and unsupervised Machine Learning and Deep Learning, which include multiple algorithms like Support Vector Machines, or SVM for short, Decision Trees, K Nearest Neighbor, or KNN for short, Artificial Neural Networks, or ANNs for short, and Convolution Neural Networks, CNNs for short [24], in the next sections these techniques and algorithms will be explained.

2.2.1 SUPERVISED MACHINE LEARNING

INTRODUCTION

In Supervised learning, the machine is trained using data that is well labeled, meaning that some data is already tagged with the correct answer. It can be compared to learning which takes place in the presence of a supervisor or a teacher. A supervised learning algorithm learns from labeled training data, to then predict the outcome for never seen data [25].

HOW IT WORKS

The first step to create a supervised learning algorithm is to create a set of labeled data, for example, to predict how long it will take to drive from home to work, this set of labeled data will include things like the weather conditions, time of the day, day of the week, if it is or not holiday season, if it is dark or not, among others, these details are called input, the output, in this case, will be how long the drive takes. In this specific case, a human instinctively knows that if it is raining the drive will take longer, however the

machine needs data. To create the model, for this example, the first thing to do is create a training dataset, this dataset contains the total commute time and the corresponding factors, the inputs, based on this training dataset the machine might see that there's a relationship between the time the drive takes and the amount of rain. So the machine ascertains that the more it rains the longer it will take to get to work, it also can make other connections that impact the time that the drive takes, such as the time that the drive starts [25].

EXAMPLES OF SUPERVISED MACHINE LEARNING

Regression

This technique predicts a single output value using training data. The above-mentioned example is an example of regression. The outputs of these techniques always have a probabilistic interpretation, and the algorithm can be regularized to avoid overfitting [25].

Logistic Regression

This method is used to estimate discrete values based on a given set of independent variables. It predicts the probability of occurrence of an event by fitting data to a logit function, as it predicts a probability, its output value is between 0 and 1. However this method may underperform when there are multiple or non-linear decision boundaries, as this method is not flexible, it does not capture more complex relationships [25].

Classification

This Algorithm groups the output inside a class. If the algorithm tries to label input into two different classes, it is called a binary classification, selecting between more than two classes is referred to as multiclass classification [25].

Support Vector Machine

Support Vector Machines (SVM) is a supervised Machine Learning algorithm used for both regression and classification problems. When used for classification purposes, it separates the classes using a linear boundary. It builds a hyperplane or a set of hyperplanes in a high dimensional space and good separation between the two classes is achieved by the hyperplane that has the largest distance to the nearest training data point of any class. The real power of this algorithm depends on the kernel function being used. The most commonly used kernels are the linear kernel, gaussian kernel, and polynomial kernel [25]. An example of SVM can be seen in Figure 2.10.

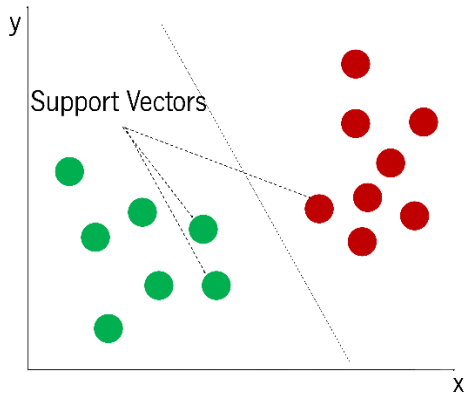

Figure 2.10 – Example of an SVM to distinguish between 2 different classes

Decision Trees

Decision Trees is also a supervised Machine Learning algorithm, which at its core is the tree data structure only, using a couple of if/else statements on the features selected. Decision Trees are based on a hierarchical rule-based method and they permit the acceptance and rejection of class labels at each intermediary level. This method consists of 3 parts partitioning the nodes, finding the terminal nodes, and allocation of the class label to the terminal node [25]. An example of a decision tree can be seen in Figure 2.11.

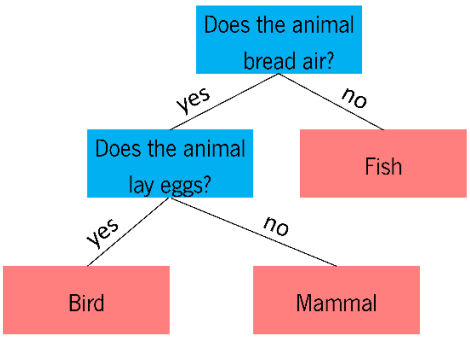

Figure 2.11 – Example of a Decision tree to classify animals

K-Nearest Neighbor

The k-Nearest Neighbor (KNN) is by far the simplest Machine Learning algorithm. This algorithm simply relies on the distance between feature vectors and classifies unknown data points by finding the most common class among the k-closest examples. In order to apply the k-nearest neighbor classification, a

distance metric or similarity function must be defined. Common choices include the Euclidean distance and Manhattan distance [25]. An example of KNN can be seen in Figure 2.12.

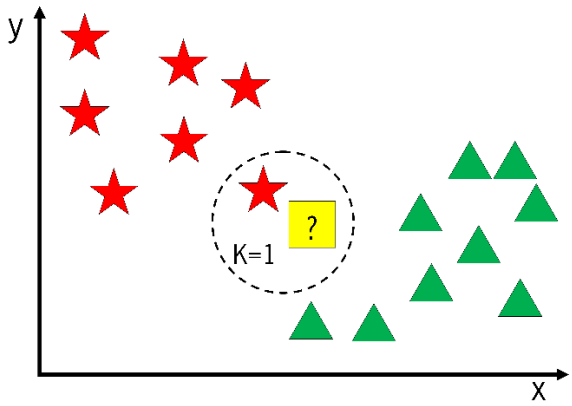

Figure 2.12 – Example of a KNN to distinguish between 2 different classes

CHALLENGES IN SUPERVISED MACHINE LEARNING

The main challenges faced in Supervised Machine Learning are irrelevant input features in the training dataset, which could give inaccurate results; the preparation and preprocessing of the data is always a challenge; accuracy suffers when impossible, unlikely, and incomplete values have been inputted as training data [25].

ADVANTAGES OF SUPERVISED LEARNING

The principal advantages of this type of learning are that it allows to collect data or produce a data output from the previous experience, and it has the capability of solving real-world computation problems [25].

DISADVANTAGES OF SUPERVISED LEARNING

The principal disadvantages of this type of learning are the need to select good examples from each class for the training of the classifier; classifying big data can be very hard; training for supervised learning needs a lot of computation time [25].

BEST PRACTICES FOR SUPERVISED LEARNING

The first thing to do in the construction of a Supervised Learning algorithm is to decide what kind of data is to be used as a training set, the next thing is to decide the structure of the algorithm and finally gather corresponding outputs. Only after this, the construction of the algorithm can begin [25].

2.2.2 UNSUPERVISED MACHINE LEARNING

INTRODUCTION

Unsupervised learning is a Machine Learning technique, where it is not needed to supervise the model. Instead, the model is allowed to work on its own to discover information. It mainly deals with unlabeled data. Unsupervised learning algorithms can perform more complex processing tasks compared to supervised learning, although, unsupervised learning can be more unpredictable [26].

WHY UNSUPERVISED LEARNING?

The prime reasons for using unsupervised learning are that it finds all kind of unknown patterns in data; it can find features which can be useful for categorization; it is easier to get unlabeled data than labeled data, which needs manual intervention [26].

TYPES OF UNSUPERVISED LEARNING

Unsupervised learning problems are grouped into two groups, clustering, and association.

Clustering

Clustering is an important concept when it comes to unsupervised learning, it mainly deals with finding a structure or pattern in a collection of uncategorized data. Clustering algorithms will process data and find natural clusters, groups, if they exist in the data. The algorithm can be modified to know how many clusters it should find [26].

Types of Clustering

Exclusive or Partitioning

In this clustering method, data is grouped in such a way that one data point can belong to one cluster only [26].

Agglomerative

In this clustering technique, every data point starts as a cluster, the iterative unions between the two nearest clusters reduce the number of clusters [26].

Overlapping

In this technique, fuzzy sets are used to cluster data, each point may belong to two or more clusters with separate degrees of membership [26].

Probabilistic

This technique uses probability distribution to create the clusters, for example, the key words man's shoe; women's shoe; women's glove; man's glove can be clustered into two categories shoe and glove or man and women [26].

Clustering Algorithms

Hierarchical Clustering

Hierarchical clustering is an algorithm that groups similar objects into the same cluster. The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other [26].

K-means Clustering

K-means is an iterative clustering algorithm that finds the highest value for every iteration, initially, the desired number of clusters is selected, in this clustering method, the data points need to be cluster into k groups. A larger k means smaller groups with more granularity in the same way a lower k means larger groups with less granularity.

It assigns a data point to one of the k groups, in K-means Clustering, each group is defined by creating a centroid for each group. The centroids capture the points closest to them and adds them to the cluster. The output of the algorithm is a group of labels [26].

K-nearest Neighbor

As stated before, this is the simplest Machine Learning algorithm, however, it is not specific to supervised learning problems, it can be also used in clustering problems [26].

Association

Association rules allow the establishment of associations amongst data objects inside large databases. This unsupervised technique is about discovering interesting relationships between variables in large databases [26].

APPLICATIONS OF UNSUPERVISED MACHINE LEARNING

Some applications of unsupervised Machine Learning are the automatic split of the dataset into groups base on their similarities; anomaly detection can discover unusual data points in the dataset, this is useful for finding fraudulent transactions; association mining identifies sets of items that often occur in the dataset [26].

DISADVANTAGES OF UNSUPERVISED LEARNING

Some of the disadvantages of unsupervised learning are the inability to get precise information regarding data sorting, and the output as data used in unsupervised learning is labeled and not known; less accuracy of the results, because the input data is not known and not labeled by people in advance, meaning the machine needs to do this itself [26].

SUPERVISED VS UNSUPERVISED LEARNING

In Table 2.1 the advantages and disadvantages of supervised and unsupervised Machine Learning are shown.

Table 2.1 – Supervised Machine Learning vs Unsupervised Machine Learning

Parameters	Supervised Machine Learning	Unsupervised Machine Learning
Process	In a supervised learning model, input and output variables will be a given.	In unsupervised learning model and input data will be a given
Input data	Algorithms are trained using labeled data	Algorithms are used against data which is not labeled
Computational Complexity	Supervised learning is a simpler method	Unsupervised learning is computationally complex
Use of data	Supervised learning model uses training data to learn a link between the input and the outputs	Unsupervised learning does not use output data

Accuracy of results	Highly accurate and trustworthy method	Less accurate and trustworthy method
Real-time learning	Learning method takes place offline	Learning method takes place in real-time
Number of classes	Number of classes is known	Number of classes is not known
Main drawback	Classifying big data can be a real challenge in Supervised Learning	You cannot get precise information regarding data sorting, and the output as data used in unsupervised learning is labeled and not known

2.2.3 DEEP LEARNING

INTRODUCTION

Deep Learning is a type of Machine Learning that trains a machine to perform human-like tasks, such as recognizing speech, identifying images, or making predictions. Instead of organizing data to run through predefined equations, Deep Learning sets up basic parameters about the data and trains the machine to learn on its own by recognizing patterns using many layers of processing, the first layer is called the input layer, the last layer is called the output layer, all layers in between are called hidden layers, the word deep means the network join neurons in more than two layers. Each hidden layer is composed of neurons, the neurons are connected to each other, the neuron will process and then propagate the input signal that it receives from the previous layer, the strength of the signal given to the neuron in the next layer depends on the weight, bias and activation function [27].

DEEP LEARNING PROCESS

A deep Neural Network provides state-of-the-art accuracy in many tasks, from object detection to speech recognition. They can learn automatically, without predefined knowledge explicitly coded by the programmers [28].

To grasp the idea of Deep Learning, imagine a family, with an infant and parents. The toddler points objects with his little finger and always says the word 'cat.' As its parents are concerned about his education, they keep telling him 'Yes, that is a cat' or 'No, that is not a cat.' The infant persists in pointing

objects but becomes more accurate with 'cats.' The little kid, deep down, does not know why he can say it is a cat or not. He has just learned how to hierarchies' complex features coming up with a cat by looking at the pet overall and continue to focus on details such as the tails or the nose before to make up his mind [28].

A Neural Network works quite the same way. Each layer represents a deeper level of knowledge, i.e., the hierarchy of knowledge. A Neural Network with four layers will learn more complex features than one with two layers [28].

The learning occurs in two phases, the first phase consists of applying a nonlinear transformation of the input and create a statistical model as output, the second phase aims at improving the model with a mathematical method known as derivative. The Neural Network repeats these two phases hundreds to thousands of times until it has reached a tolerable level of accuracy. The repeat of this two-phase is called an iteration [28].

CLASSIFICATION OF NEURAL NETWORKS

Neural Networks can either be shallow Neural Networks if the network has only one hidden layer or the can be deep Neural Networks if they have more than one hidden layer [28].

ACTIVATION FUNCTIONS

An activation function is applied as the last step in the chain of a neuron. It defines what output the neuron will present. Nonlinear behavior is commonly introduced through the use of activation functions. Some commonly used activation functions can be seen in Table 2.2.

Table 2.2 – Activation Functions used in Deep Learning

Activation function	*Formula*	*Remarks*
Identity	$f(x) = x$	
Binary step	$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$	
Rectified linear unit (ReLU)	$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$	Despite being easy to compute, this function introduces nonlinear behavior.

Sigmoid (logistic)	$f(x) = \frac{1}{1 + e^{-x}}$	
Softmax	$f_i(\vec{x}) = \frac{e^{x_i}}{\sum_{j=1}^{J} e^{x_j}}$ for $i = 1, \ldots, J$	Provides a normalized probability distribution of all possible classes, commonly used as the activation function for the output layer.

At the last layer, where the output of the Neural Network is presented, it is desired to have this output data displayed in the most useful way. In order to do so, the activation function of this last layer could provide a binary output where only the most probable class is set to 1 while all other classes are set to 0. More commonly, all outputs of all other output neurons are used to provide a normalized probability distribution, a Softmax function provides the latter [29].

LOSS FUNCTIONS

A loss function, also known as the error function, computes the difference between the desired output of the Neural Network and the predicted one. The resulted loss between two different loss functions for the exact same output can greatly vary. Selecting the most suitable loss function for a given problem can benefit the evaluation of the current state of the model considerably. Loss functions can be divided in a couple of categories, depending on the type of problem [30]:

- Regressive loss functions: for problems without a discrete number of possible outcomes.
- Classification loss functions: when searching for the highest probable class, given a discrete number of classes.
- Embedded loss functions: when searching for the similarity between two inputs.

Some common loss functions are presented in Table 2.3. For some specific situations, it may be beneficial to create a problem-specific loss function.

Table 2.3 – Loss Functions used in Deep Learning

Loss function	*Formula*
Mean Absolute Error (MAE) L1 loss	$L = L1 = \frac{1}{n}\sum_{i=1}^{n} abs(Y_i - \hat{Y}_i)$

Mean Squared Error (MSE) L2 Loss	$L = L2 = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$ n = number of classes Y = vector of n observed values $\hat{Y}$ = vector of n predictions
Root Mean Squared Error (RMSE)	$L = \sqrt{MSE}$
Cross-Entropy (binary: 2 classes)	$L = -(ylog(p) + (1-y)log(1-p))$ y = binary indicator if classification is correct p = predicted probability observation
Cross-Entropy (3+ classes)	$L = -\sum_{i=1}^{n} y_i \log(p_i)$ n = number of classes p = predicted probability observation for class i y = 1 if class i is correct classified, otherwise 0
Hinge loss	$L = \sum_{i=1}^{n} \max(0, 1 - y_i * p_i)$

OPTIMIZERS

Optimizers or optimization algorithms are designed to minimize (or sometimes maximize) the result of an objective function (error function). This is done through the altering of parameters that influence the error function. In the context of Neural Networks, optimizers alter the weights and bias of the neurons in an attempt to shape the network in such a way that it minimizes the loss (result from loss function), hence resulting in a more desirable outcome [31].

The Learning Rate (LR) defines how much change is made to the parameters during optimization and can be divided in two categories [32]:

- Constant learning rate: the learning rate is a hyperparameter assigned with a fixed value before optimization took place. This requires the user to be aware of a proper LR, which may be challenging. When the LR is relatively small, the optimization process will only cause small changes every step and take a relatively long time to get the desired results. However, when the LR is set relatively big, it may be difficult for the optimizer to find optimal values as it potentially keeps overshooting or undershooting. Another obstacle is that all parameters are updated with the same LR, eliminating parameter-specific needs [32].
- Adaptive learning rate: rather than a fixed LR for all parameters, adaptive LR optimization algorithms use methods to find the most suitable LR for every distinctive parameter. This eliminates the need to define a fixed LR and automatically defines the LR to the given specific situation [32].

A common constant LR optimization algorithm is Stochastic Gradient Descent (SGD). In SGD optimization all parameters are subtracted with a multiplication of the learning rate and the first derivative of the loss function with respect to those parameters [32].

Root Mean Square Propagation (RMSProp) is an adaptive LR optimization algorithm. For each of the parameters individually, the LR is altered by the magnitude of recent squared gradients for the respective parameter. Those recent gradients are exponentially decaying with a 'decay rate' hyperparameter. In the RMSProp algorithm, the concept of momentum is introduced. This refers to the use of knowledge from previous steps to find the best next step [32].

Adaptive Moment Estimation (Adam) is another adaptive LR optimization algorithm, an updated version of the RMSProp optimizer. Adam uses first-order as well as second-order moment, through storage of exponentially decaying average of past squared gradients (just like RMSProp) as well as an exponentially decaying average of past gradients [32].

Many other optimization algorithms exist, such as Adagrad, Adadelta, Adamax, Nadam, AMSGrad, among others [32].

TYPES OF DEEP LEARNING NETWORKS

Artificial Neural Networks, inspired by the properties of biological Neural Networks, Artificial Neural Networks are statistical learning algorithms and are used for a variety of tasks, from relatively simple classification tasks to computer vision and speech recognition. ANNs are implemented as a system of interconnected processing elements, called nodes, which are functionally analogous to biological

neurons. The connections between different nodes have numerical values, called weights, and by systematically altering these values, the network is eventually able to approximate the desired function. ANNs are divided into layers, made of neurons, there is an input layer where the data from the preprocessing goes, hidden layers. Such layers can be thought of as individual feature detectors, recognizing more and more complex patterns in the data as it is propagated throughout the network, for example, if the network is given a task to recognize a face, the first hidden layer might act as a line detector, the second hidden layer takes these lines as input and puts them together to form a nose, the third hidden layer takes the nose and matches it with an eye and so on until finally the whole face is constructed. This hierarchy enables the network to eventually recognize very complex objects and the output layer which is the classification [28]. An example of an ANN can be seen in Figure 2.13.

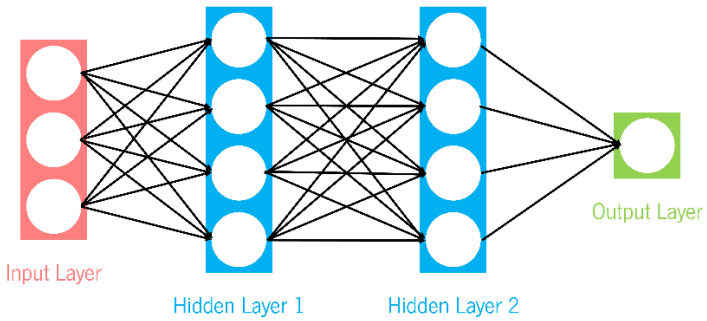

Figure 2.13 – Example of an ANN with 4 layers and 12 neurons

Feed-Forward Neural Network

The simplest type of Artificial Neural Network. With this type of architecture, information flows in only one direction, forward. It means, the information's flow starts at the input layer, goes to the hidden layers, and end at the output layer. The network does not have a loop, information stops at the output layer [28].

Recurrent Neural Networks (RNN)

RNN is a multi-layered Neural Network that can store information in context nodes, allowing it to learn data sequences and output a number or another sequence. In simple words, it is an artificial Neural Network whose connection between neurons include loops. RNNs are well suited for processing sequences of inputs [28].

For example, if the task is to predict the next word in the sentence "Do you want a _____?"

- The RNN neurons will receive a signal that points to the start of the sentence.
- The network receives the word "Do" as an input and produces a vector of the number. This vector is fed back to the neuron to provide a memory to the network. This stage helps the network to remember it received "Do" and it received it in the first position.
- The network will similarly proceed to the next words. It takes the word "you" and "want." The state of the neurons is updated upon receiving each word.
- The final stage occurs after receiving the word "a." The Neural Network will provide a probability for each English word that can be used to complete the sentence. A well-trained RNN probably assigns a high probability to "café," "drink," "burger," etc.

Common uses of RNN are:

- Help securities traders to generate analytic reports
- Detect abnormalities in the contract of financial statement
- Detect fraudulent credit-card transaction
- Provide a caption for images
- Power chatbots
- The standard uses of RNN occur when the practitioners are working with time-series data or sequences (e.g., audio recordings or text).

Convolutional Neural Networks (CNN)

Convolutional Neural Network is a special architecture of artificial Neural Networks. CNN's uses some of its features of the visual cortex and has therefore achieved state of the art results in computer vision tasks. Convolutional Neural Networks are comprised of two very simple elements, namely convolutional layers and pooling layers Although simple, there are near-infinite ways to arrange these layers for a given computer vision problem. The elements of a Convolutional Neural Network, such as convolutional and pooling layers, are relatively straightforward to understand. The challenging part of using Convolutional Neural Networks in practice is how to design model architectures that best use these simple elements [28]. An example of a CNN can be seen in Figure 2.14.

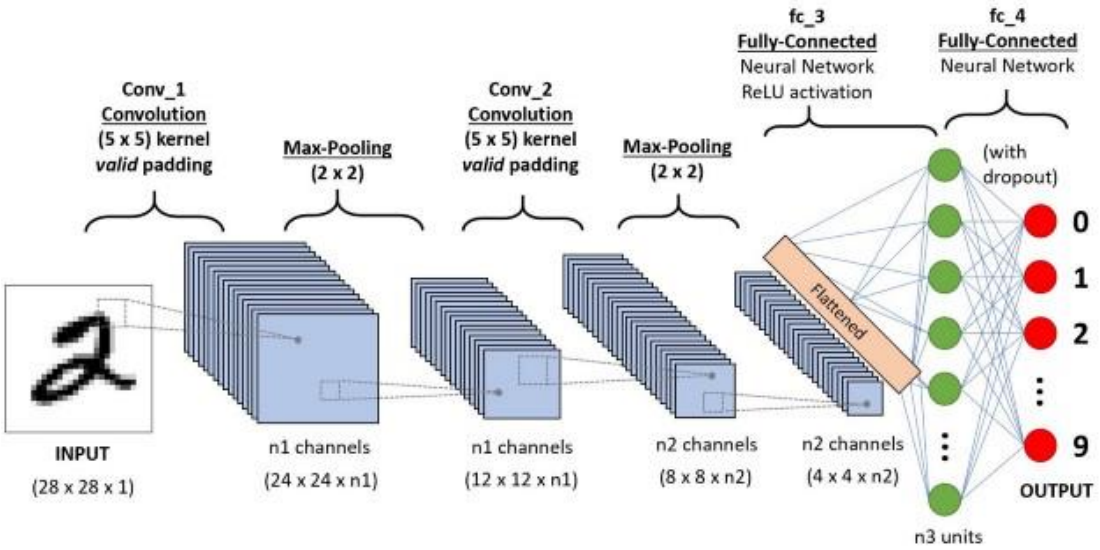

Figure 2.14 – Example of a CNN to identify handwritten characters, retrieved from [33]

Reinforcement Learning

Reinforcement Learning is defined as a Machine Learning method that is concerned with how software agents should take actions in an environment. Reinforcement Learning is a part of the Deep Learning method that helps to maximize some portion of the cumulative reward.

Typical Reinforcement Learning scenario

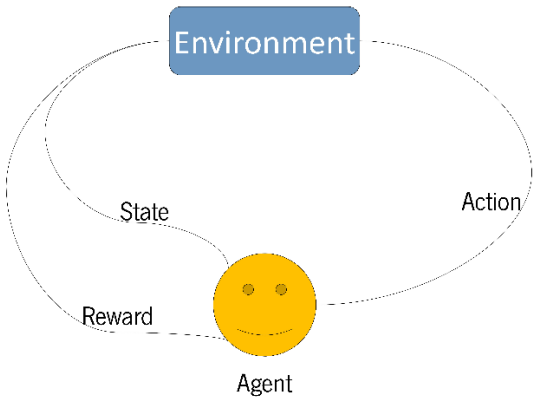

Figure 2.15 – Typical Reinforcement Learning scenario

Figure 2.15 demonstrates the typical Reinforcement Learning scenario. This scenario contains:

- Agent: It is an assumed entity that performs actions in an environment to gain some reward.
- Environment (e): A scenario that an agent has to face.

- Reward (R): An immediate return given to an agent when he or she performs specific action or task.
- State (s): State refers to the current situation returned by the environment.
- Policy (π): It is a strategy which applies by the agent to decide the next action based on the current state.
- Value (V): It is expected long-term return with discount, as compared to the short-term reward.
- Value Function: It specifies the value of a state that is the total amount of reward. It is an agent that should be expected beginning from that state.
- Model of the environment: This mimics the behavior of the environment. It helps you to make inferences to be made and also determine how the environment will behave.
- Model-based methods: It is a method for solving reinforcement learning problems that use model-based methods.
- Q value or action value (Q): Q value is quite similar to value. The only difference between the two is that it takes an additional parameter as a current action.

How does Reinforcement Learning work?

The agent performs an action in the environment, that action changes the state of the environment, the agent receives the new state and the reward of its action, with these parameters the agent can perform a better action until it reaches the objective [34].

Reinforcement Learning Algorithms

There are three approaches to implement a Reinforcement Learning algorithm value-based, policy-based, and model-based [34].

- In a value-based Reinforcement Learning method, you should try to maximize a value function V(s). In this method, the agent is expecting a long-term return of the current states under policy π.
- In a policy-based RL method, you try to come up with such a policy that the action performed in every state helps you to gain maximum reward in the future. Two types of policy-based methods are:
    - Deterministic: For any state, the same action is produced by the policy
    - Stochastic: Every action has a certain probability, which is determined by an equation

- In model-based RL method, it is necessary to create a virtual model for each environment. The agent learns to perform in that specific environment.

Types of Reinforcement Learning

There are two kinds of reinforcement learning methods positive and negative [34]

- Positive: It is defined as an event, that occurs because of specific behavior. It increases the strength and the frequency of the behavior and impacts positively on the action taken by the agent. This type of Reinforcement helps you to maximize performance and sustain change for a more extended period. However, too much reinforcement may lead to over-optimization of the state, which can affect the results.
- Negative: Negative Reinforcement is defined as the strengthening of behavior that occurs because of a negative condition that should have stopped or avoided. It helps you to define the minimum stand of performance. However, the drawback of this method is that it provides enough to meet up the minimum behavior.

Learning Models of Reinforcement

There are two important learning models in reinforcement learning, the Markov Decision Process and Q-Learning [34]

Markov Decision Process

A Markov Decision Process (MDP) provides a framework for decision making in which the outcome is partly under the control of a decision-maker but also partly random [34].

An MDP can be defined by a tuple of parameters:

- S: a set of states.
- A: a set of actions.
- $T(s'|s,a)$ is the transition function. This function expresses the probability for state s to become s' when acting a in state s, for every possible state s'. This implies that $\sum_{s'} T(s'|s,a) = 1$.
- $R\ (s,a,s') \rightarrow r_a(s)$: the reward function returns a scalar reward when going from state s to state s' through action a.
- $\pi(s) \rightarrow$ a: the policy π describes the best action to take according to this policy, for each state.

An MDP is Markovian, which means the transition function only depends on the current state to find the probability to go with any of the possible actions to any of the possible next states.

Value function approaches provide an estimation for the future rewards of a given policy. Therefore, they may be suitable to compare different policies [35]:

- $V^{\pi}(s) = E_{\pi}[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s]$: The state value function of a policy π will estimate the total reward for policy π, starting at state s. The symbol $\gamma \in [0, 1]$ denotes the discount factor, which allows an infinite sum to become finite when $\gamma < 1$ by exponentially decreasing the influence of future rewards. The optimum (state) value function $V^*(s)$ uses a policy π with the highest returned value, i.e. $V^*(s) = \max_{\pi} V^{\pi}(s)$. Any policy π that is capable of achieving the optimum value is called an optimum policy.
- $Q^{\pi}(s, a) = E_{\pi}[\sum_{t=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a]$: Similar to the state value function, the state-action value function (also known as Q-function) returns the expected reward for policy π, starting at state s and taking action a. Different than the state value function, the state-action value function must be supplied with an action a to take at state s.

Q-Learning

Q-learning is a reinforcement learning technique, using temporal-difference methods. Its goal is to find the best policy when being provided with a finite amount of states and actions and a reward for each time transition to a different state. The reward function does not need to be known, as long as a reward is given at every iteration.

When the Markov decision process is finite, Q-learning can always find an optimum policy when given infinite exploration time and a partially random behavior [36].

Q-learning can be implemented with a two-dimensional array, with all possible states mapping all possible actions, returning the action-value for each state. When initialized with zeros, Q-learning does not have a preference for what action to take in a given state as all possible actions lead to the same outcome, hence behaving randomly. At every iteration, the obtained states for each of the possible actions are observed, and (when not randomly decided) the action that results in the most optimum state (highest returned value) is selected. The value of the new state $Q_t(s_{t+1}, a)$, together with the received reward r_t are then used to update the value of the state-action pair, according to this formula:

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha * \left(r_t + \gamma * \max_a Q_t(s_{t+1}, a) - Q_t(s_t, a_t) \right)$$

The learning rate $\alpha \in]0, 1]$ expresses the trade-off between the momentary knowledge of the current state-action pair and the newly obtained knowledge. When this parameter would be 0 no learning would occur while setting it to 1 would replace all previously obtained knowledge regarding the handled state-action pair by the knowledge that is obtained at that iteration. The discount factor $\gamma \in [0, 1]$ expresses the importance of rewards that were given in previous iterations. When the discount factor would be 0, only the currently given reward is of importance, neglecting rewards that are multiple actions away from the current state. When the discount factor is 1, the source of a specific reward easily becomes difficult to track.

After a significant amount of iterations, all possible state-action pairs have been altered in such a way that they provide an approach to the optimum policy.

Partially random behavior can prevent Q-learning to be stuck at local optima, not being aware of better alternatives. This behavior can be added through the introduction of exploration.

Applications of Reinforcement Learning

- Robotics for industrial automation.
- Business strategy planning.
- Machine Learning and data processing.
- Create training systems that provide custom instruction and materials according to the requirement of students.
- Aircraft control and robot motion control.

Why use Reinforcement Learning?

- Find which situation needs an action.
- Discover which action yields the highest reward over the longer period.
- Reinforcement Learning also provides the learning agent with a reward function.
- It also allows it to figure out the best method for obtaining large rewards.

When not to use Reinforcement Learning?

Reinforcement learning cannot be applied in all situations. Some situations where RL should not be used are when there is enough data to solve the problem with a supervised learning method; RL is computing heavy and time-consuming, in particular when the action space is large.

Challenges of Reinforcement Learning

The major challenges of RL are:

- Feature/reward design which should be very involved.

- Parameters may affect the speed of learning.
- Realistic environments can have partial observability.
- Too much Reinforcement may lead to an overload of states which can diminish the results.
- Realistic environments can be non-stationary.

EXAMPLES OF DEEP LEARNING APPLICATIONS

AI in Finance: The financial technology sector has already started using AI to save time, reduce costs, and add value. Deep Learning is changing the lending industry by using more robust credit scoring. Credit decision-makers can use AI for robust credit lending applications to achieve faster, more accurate risk assessment, using machine intelligence to factor in the character and capacity of applicants [28].

Underwrite is a Fintech company providing AI solutions for credit companies. Underwrite.ai uses AI to detect which applicant is more likely to pay back a loan. Their approach radically outperforms traditional methods [28].

AI in Human Resources (HR): Under Armour, a sportswear company revolutionizes hiring and modernizes the candidate experience with the help of AI. In fact, Under Armour reduces hiring time for its retail stores by 35%. Under Armour faced a growing popularity interest back in 2012. They had, on average, 30000 resumes a month. Reading all of those applications and begin to start the screening and interview process was taking too long. The lengthy process to get people hired and onboarded impacted Under Armour's ability to have their retail stores fully staffed, ramped, and ready to operate [28].

At that time, Under Armour had all of the 'must-have' HR management systems in place such as transactional solutions for sourcing, applying, tracking and onboarding but those tools were not useful enough. Under Armour choose HireVue, an AI provider for HR solutions, for both on-demand and live interviews. The results were bluffing; they managed to decrease by 35% the time to fill vacant jobs and hired higher quality staff [28].

AI in Marketing: AI is a valuable tool for customer service management personalization challenges. Improved speech recognition in call-center management and call routing as a result of the application of AI techniques allows a more seamless experience for customers [28].

For example, deep-learning analysis of audio allows systems to assess a customer's emotional tone. If the customer is responding poorly to the AI chatbot, the system can be rerouted the conversation to real human operators that take over the issue [28].

Apart from the three examples above, AI is widely used in other sectors/industries [28].

WHY IS DEEP LEARNING IMPORTANT?

Deep Learning is a powerful tool to make predictions and actionable results. Deep learning excels in pattern discovery (unsupervised learning) and knowledge-based prediction. Big data is the fuel for Deep Learning. When both are combined, an organization can reap unprecedented results in terms of productivity, sales, management, and innovation.

Deep learning can outperform the traditional method. For instance, Deep Learning algorithms are 41% more accurate than Machine Learning algorithms in image classification, 27% more accurate in facial recognition, and 25% in voice recognition [28].

LIMITATIONS OF DEEP LEARNING

Data labeling

Most current AI models are trained through "supervised learning." It means that humans must label and categorize the underlying data, which can be a sizable and error-prone chore. For example, companies developing self-driving-car technologies are hiring hundreds of people to manually annotate hours of video feeds from prototype vehicles to help train these systems [28].

Obtain huge training datasets

It has been shown that simple Deep Learning techniques like CNN can, in some cases, imitate the knowledge of experts in medicine and other fields. The current wave of Machine Learning, however, requires training data sets that are not only labeled but also sufficiently broad and universal.

Deep-learning methods required thousands of observations for models to become relatively good at classification tasks and, in some cases, millions for them to perform at the level of humans. Without surprise, Deep Learning is famous in giant tech companies; they are using big data to accumulate petabytes of data. It allows them to create an impressive and highly accurate Deep Learning model [28].

Explain a problem

Large and complex models can be hard to explain, in human terms. For instance, why a particular decision was obtained. It is one reason that acceptance of some AI tools are slow in application areas where interpretability is useful or indeed required.

Furthermore, as the application of AI expands, regulatory requirements could also drive the need for more explainable AI models [28].

2.3 ZERO SHOT LEARNING

2.3.1 STATE OF THE ART

Any of the techniques previously presented are very efficient in predicting the classification of an image in a labeled dataset, for example, identify which breed a dog is, some can do it better than others, but all can do it with very good results. However, they can not identify an image as dog since the word "dog" itself is not a category, if the category "dog" is added to the dataset the model will have a hard time deciding if an image is a dog or a specific breed, although ANNs and CNNs can classify an image with multiple categories if they use a sigmoid function as the output layer, this does not solve the problem since neither classifier can predict classes they have not encountered during training, such as "puppy", they can not transfer semantic information about categories they have been trained on to unseen but similar categories [37].

Over the last few decades machines have become much smarter but without a properly labeled training dataset of seen classes, it cannot distinguish between two similar objects. In Machine Learning, this is considered as the problem of zero-shot learning (ZSL) [19]. On the other hand, humans are capable of identifying approximately 30000 basic object categories.

In the case of machines, the ZSL recognition relies on the existence of a labeled training set of seen classes and the knowledge about how each unseen class is semantically related to the seen classes [38].

Zero-shot Machine Learning is used to construct recognition models for unseen target classes that have not been labeled for training. It utilizes the class attributes as aside information and transfers information from source classes with labeled samples. ZSL is done in two stages, training, where the knowledge about the attributes is captured, and inference, the knowledge is then used to categorize instances among a new set of classes. Recently, there has been a surge in interest in automatic recognition of attributes, due to the availability of data containing meta-information [39].

Zero-shot learning approaches are designed to learn intermediate semantic layer, their attributes, and apply them at inference time to predict a new class of data [40], Zero-Shot Learning also relies on the existence of a labeled training set of seen classes and unseen classes. Both seen and unseen classes

are related in a high dimensional vector space, called semantic space, where the knowledge from seen classes can be transferred to unseen classes [38].

With the semantic space and a visual feature representation of image content, ZSL can be solved in two steps, a joint embedding space is learned where both the semantic vectors (prototypes) and the visual feature vectors can be projected to, next-nearest neighbor (NN) search is performed in this embedding space to match the projection of an image feature vector against that of an unseen class prototype [38].

In order to make a model capable of solving ZSL in an effective way, the key features (images and text) are categorized as vectors. This means sourcing the specific vectors beforehand for the project. Once collected, they are provided with a description that enables the algorithms to classify them accordingly. The training is done with respect to these vectors which leads to classification according to separate classes. The testing phase recognizes new inputs and again leads to newer classes, regardless of the train data [19]. An example of how a model can solve ZSL can be seen in Figure 2.16.

2.3.2 DEVISE ARCHITECTURE

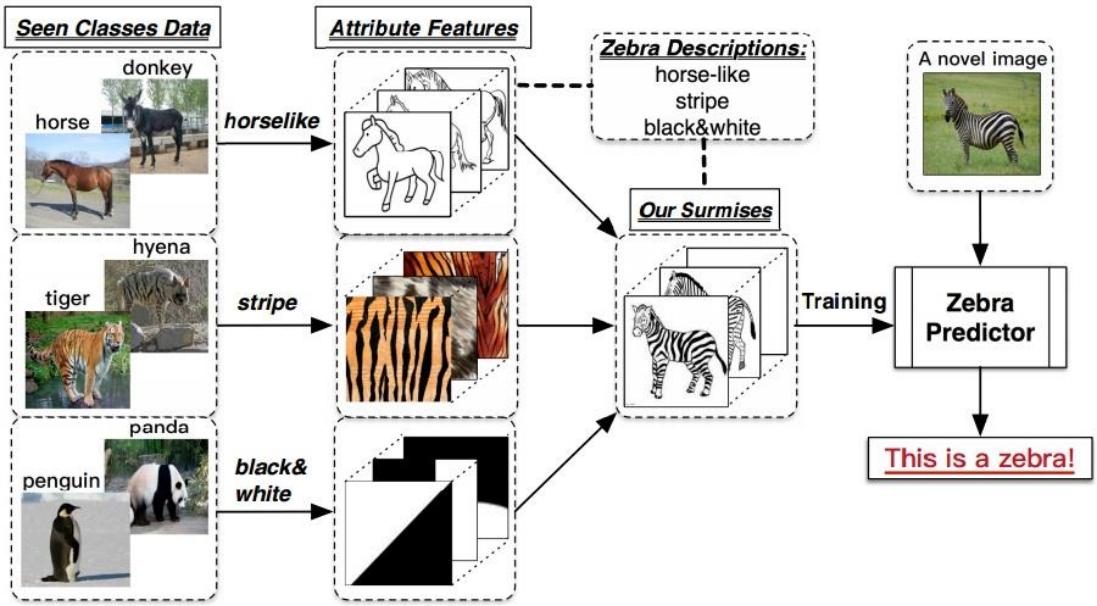

Figure 2.16 – Example of how a model solves ZSL, retrieved from [41]

For this specific problem a DeViSE architecture will be used [42], DeViSE stands for Deep Visual-Semantic Embedding, the authors present a novel image classification method which leverages semantic knowledge learned using language models, the method is able to make zero-shot predictions of tens of thousands of image labels not observed during training. Since modern visual recognition systems are often limited in their ability to scale to large numbers of object categories, in part due to the increasing

difficulty of acquiring sufficient training data in the form of labeled images as the number of object categories grows, a better solution is needed. There's where DeViSE comes in, since it uses source data both from image data, as other models do, but also from text data. The inner workings of this specific architecture will be discussed in a later chapter.

Once the algorithm is trained it is saved to be used in an application, in that application the trained algorithm will be presented with a medical image, the algorithm will then provide the top 5 most likely disorders present in such image. Figure 2.17 displays the program interface and Figure 2.18 displays the predictions made by the model for an X-Ray image.

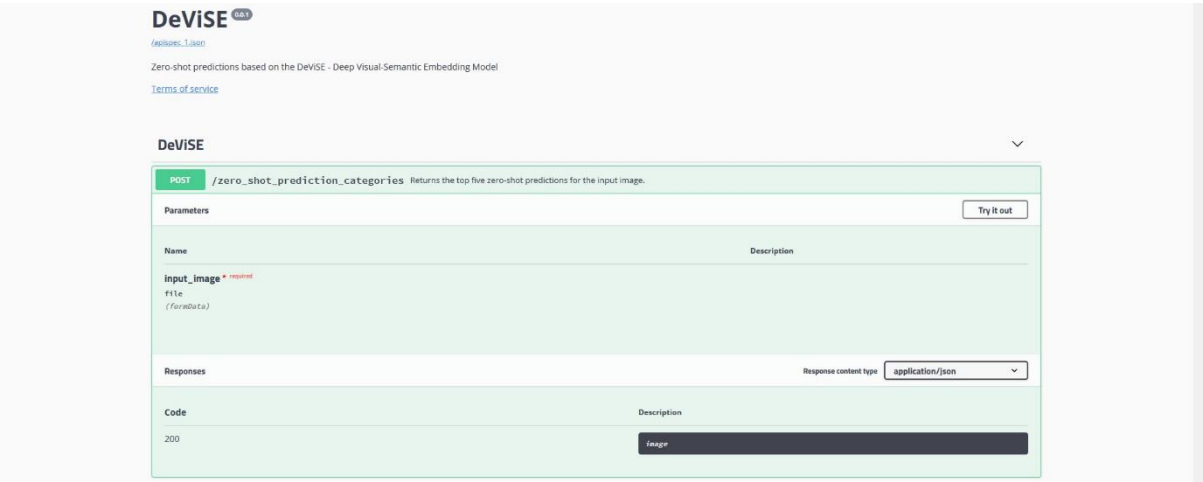

Figure 2.17 – Interface of the program to predict conditions in medical imaging

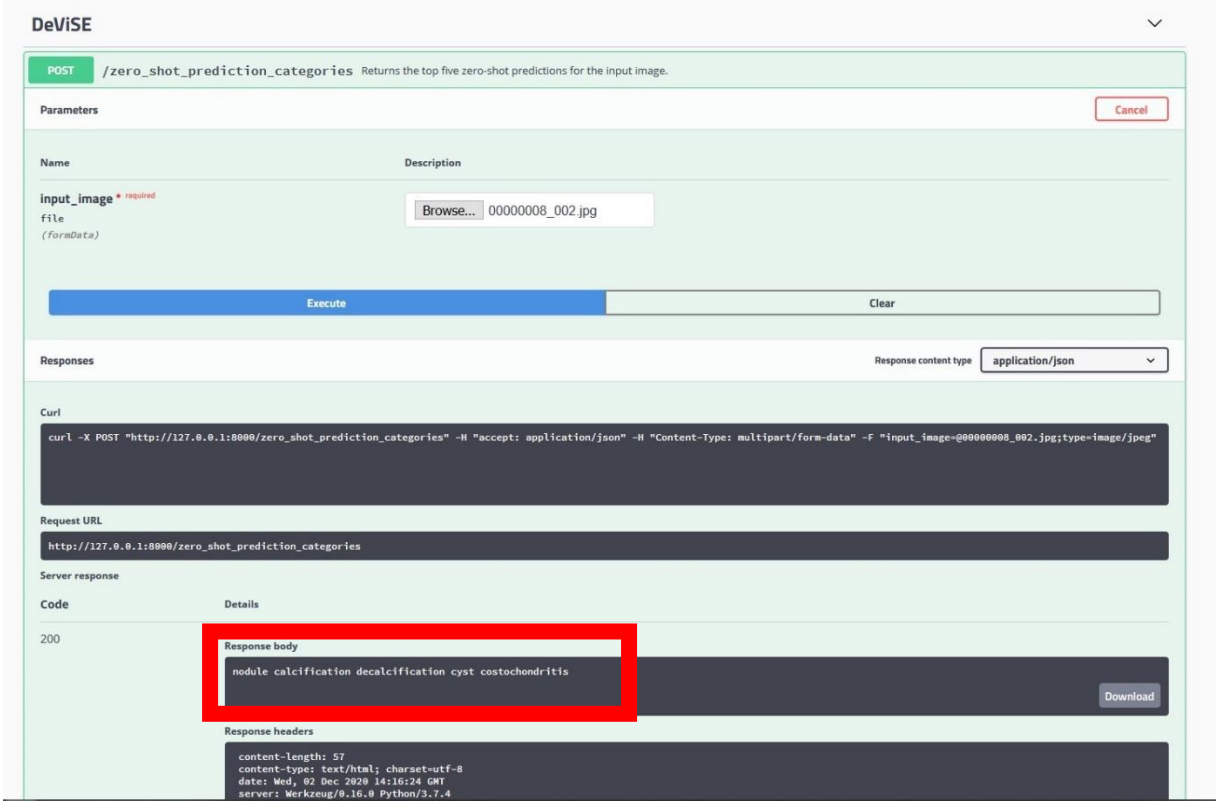

Figure 2.18 – Predictions made by the model for an X-Ray image

3 DeViSE Medical Imaging Case Study

The present chapter is about:

- Materials: the image and text data used to train the model for this specific application.
- Methods: the methods used to develop the solution to this problem.
- Results: the results obtain for the different medical imaging modalities with this solution.

3.1 MATERIALS

In order to develop this algorithm, as stated before, images and word vectors are needed, the images used to do so, are the ones in the ImageNet witch is an image database organized according to the WordNet hierarchy and a subset of X-Rays from the first X-Ray dataset, WordNet is a large lexical database of English. To obtain the word vectors it was used a pre-trained model on Wikipedia using fastText. In order to evaluate the algorithm performance, images of X-Rays and CT scans with several disorders from the academictorrents.com repositories were used.

3.1.1 IMAGE DATA

As stated before the image data consists of the labeled images of the ImageNet database, this database contains one thousand categories of images, with each category being represented by about 1280 images for a total of 1281167 images, the categories are very diverse and include things like “goldfish”, “hyena”, “lipstick”, “modem”, "monastery", "sunglasses", "banana", “hip”, etc.; and a subset of X-Rays of the first X-Ray dataset containing 8 categories, with each category being represented by 1000 images for a total of 8000 images.

For the validation, 3 datasets will be used two for X-Ray and another for CT.

The first X-Ray dataset contains 112120 labeled X-Ray images from human torsos.

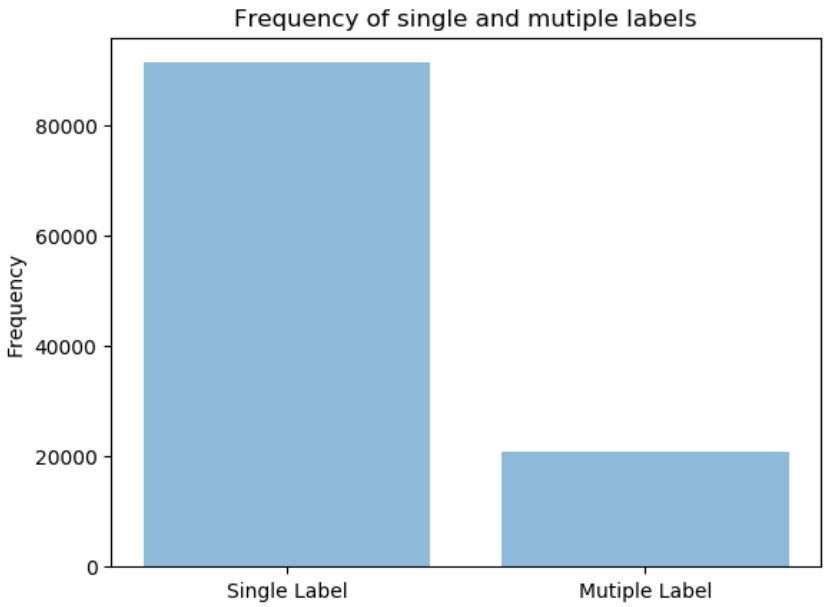

Figure 3.1 – Frequency of images with a single label and multiple labels

As seen in Figure 3.1 of the 112120 images 91385 images contain a single label, meaning that only one condition was found or that no condition was found, and the remaining 20735 contain more than one label meaning that more than one condition was found.

The conditions present in this dataset are cardiomegaly, emphysema, effusion, hernia, infiltration, mass, nodule, pneumothorax, pleural thickening, pneumonia, atelectasis, fibrosis, edema, and consolidation, which are labeled according, an extra-label “no finding” is used to label images were none of these conditions were found. The labels have the following distribution.

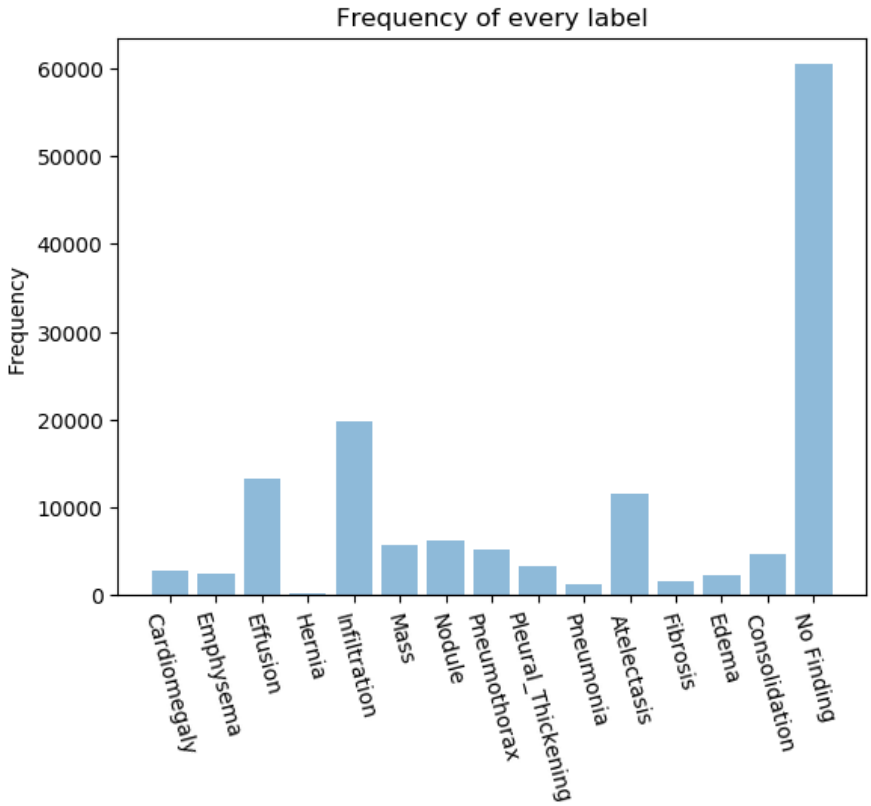

Figure 3.2 – Frequency of every label present in the X-Ray dataset

As seen in Figure 3.2 there are 2772 images labeled Cardiomegaly, 2516 labeled Emphysema, 13307 labeled Effusion, 227 labeled Hernia, 19870 labeled Infiltration, 5746 labeled Mass, 6323 labeled Nodule, 5298 labeled Pneumothorax, 3385 labeled Pleural_Thickening, 1353 labeled Pneumonia, 11535 labeled Atelectasis, 1686 labeled with Fibrosis, 2303 labeled with Edema, 4667 labeled with Consolidation and 60412 labeled with No Finding.

The second X-Ray dataset consists of 12089 images of different X-Rays, since this dataset is used to detect where pneumothorax is present the labels are -1 for no pneumothorax or a set of pixels that represent where in the image the pneumothorax is, since the objective of this work is only to find if a condition is present or not the labels were changed to 0 for no pneumothorax or 1 for pneumothorax and have the following distribution

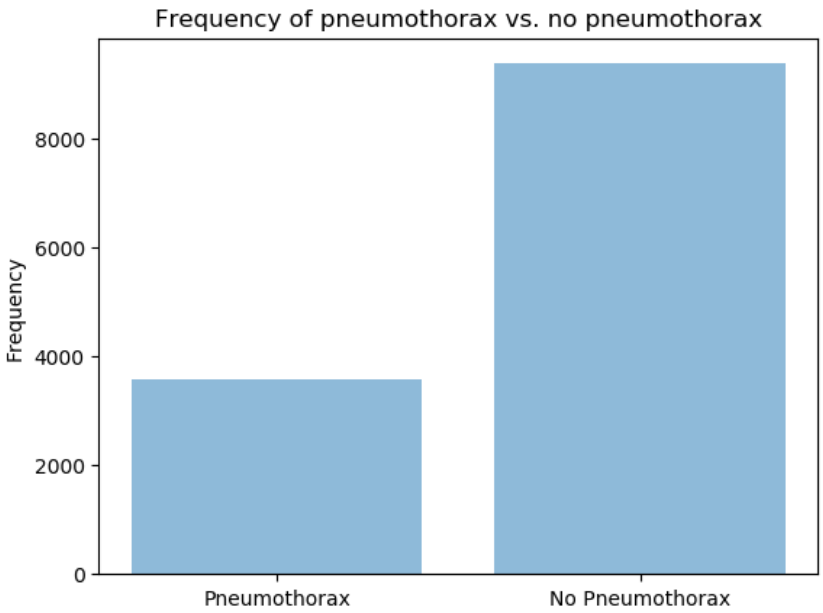

Figure 3.3 – Frequency of pneumothorax vs. no pneumothorax

As seen in Figure 3.3 pneumothorax has 3576 occurrences while no pneumothorax has 9378 occurrences.

The CT dataset consists of 1595 CT scans of human torsos each one, on average has 179 slices. The labels are either 0 for no lung cancer or 1 for lung cancer and have the following distribution

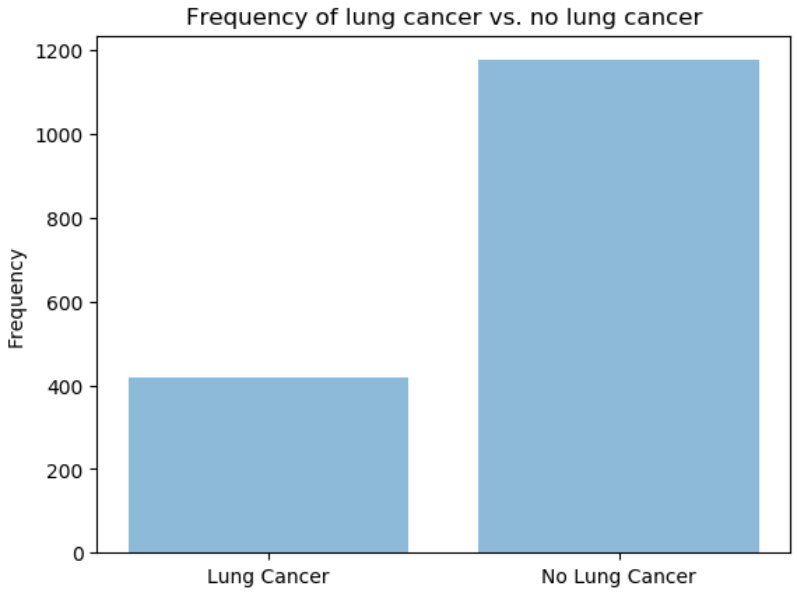

Figure 3.4 – Frequency of lung cancer vs. no lung cancer

As seen in Figure 3.4 419 CT scans are labeled with lung cancer, 1, while the remaining 1176 are labeled with no lung cancer, 0.

3.1.2 TEXT DATA

Text data consists of the word vectors obtained from the fastText library that according to its creators contains the vectors, in dimension 300, for every word in the English vocabulary, over 2.5 million words. The text data also consists of the lexical database of English known as WordNet. Nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms, named synsets, each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The resulting network of meaningfully related words and concepts. In the image database, each node of the hierarchy is depicted by, on average, about five hundred images.

3.2 METHODS

Figure 3.5 shows the overall flow diagram of how the solution was built.

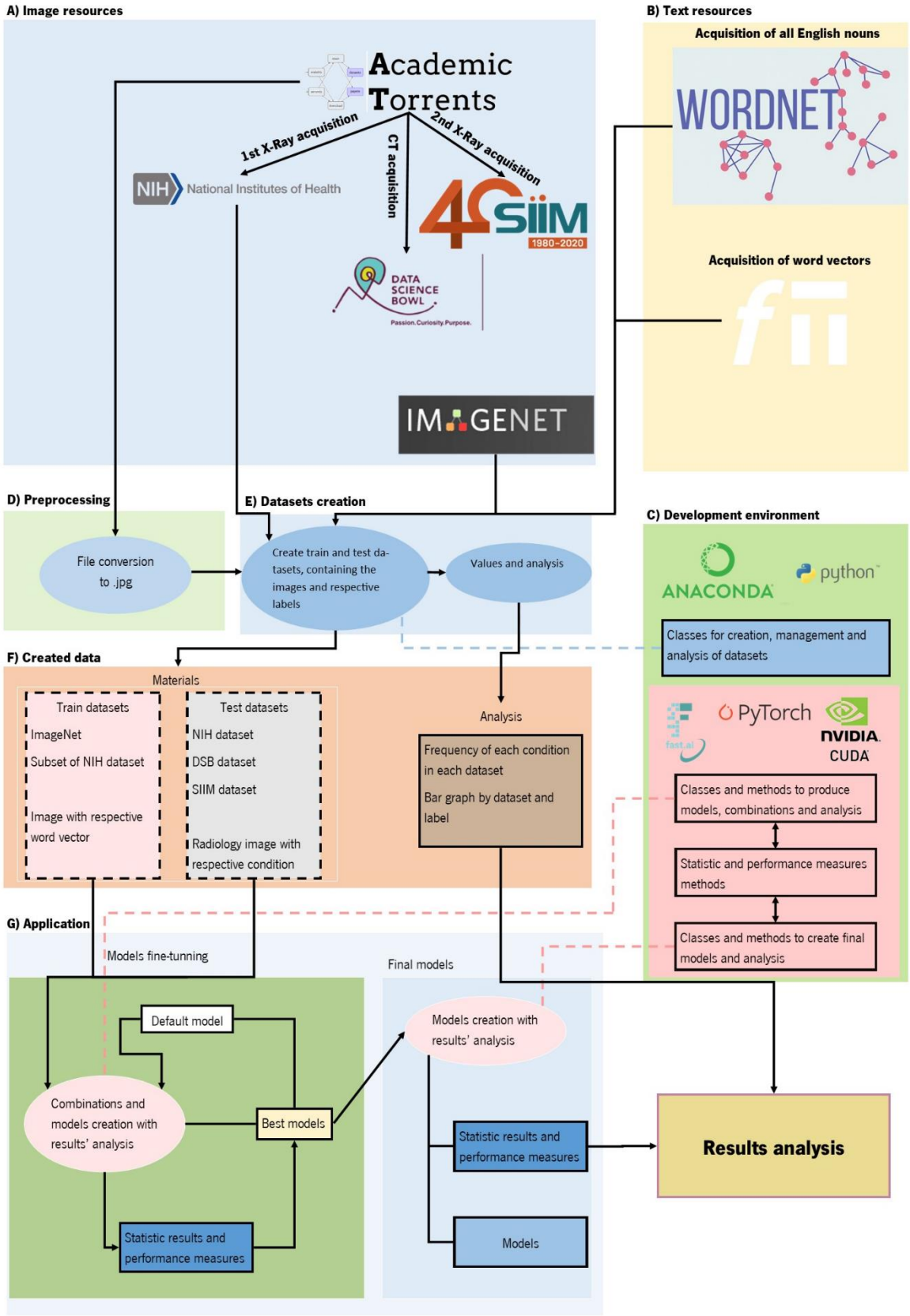

Figure 3.5 – Overall flow diagram

3.2.1 IMAGE RESOURCES

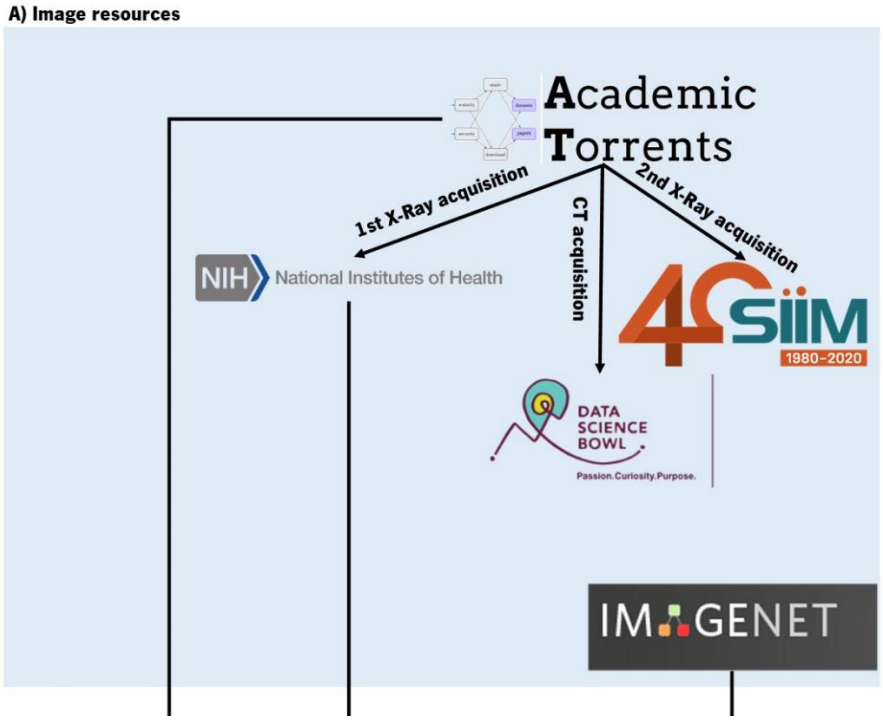

Figure 3.6 – Attainment of the image resources

As said before this type of algorithm needs both image and text data. The image data will be the ImageNet database, X-Ray images, provided by the National Institutes of Health and the Society for Imaging Informatics in Medicine and CT scans provided by the Data Science Bowl, all medical images are provided through the Academic Torrents website, Figure 3.6.

3.2.2 TEXT RESOURCES

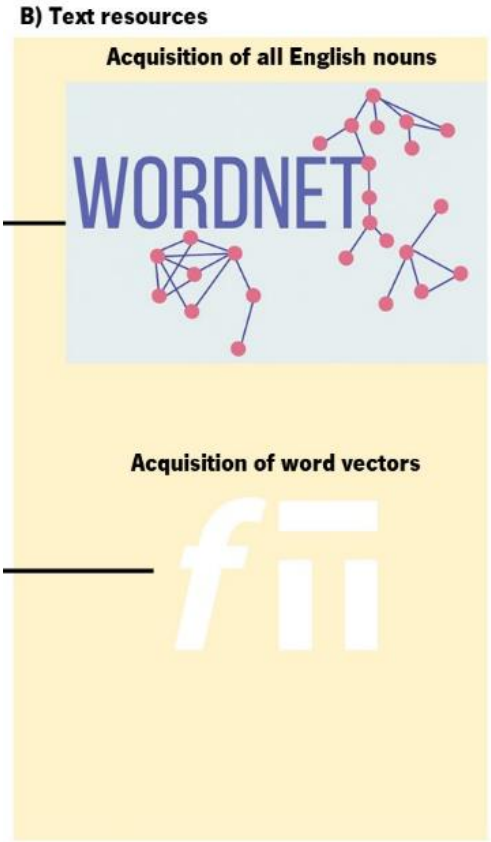

Figure 3.7 – Attainment of the text resources

The text data are the words present in the WordNet database, containing all the English nouns, and the English word vectors from the fastText library, containing high dimensional word vectors for all English words, Figure 3.7.

3.2.3 DEVELOPMENT ENVIRONMENT

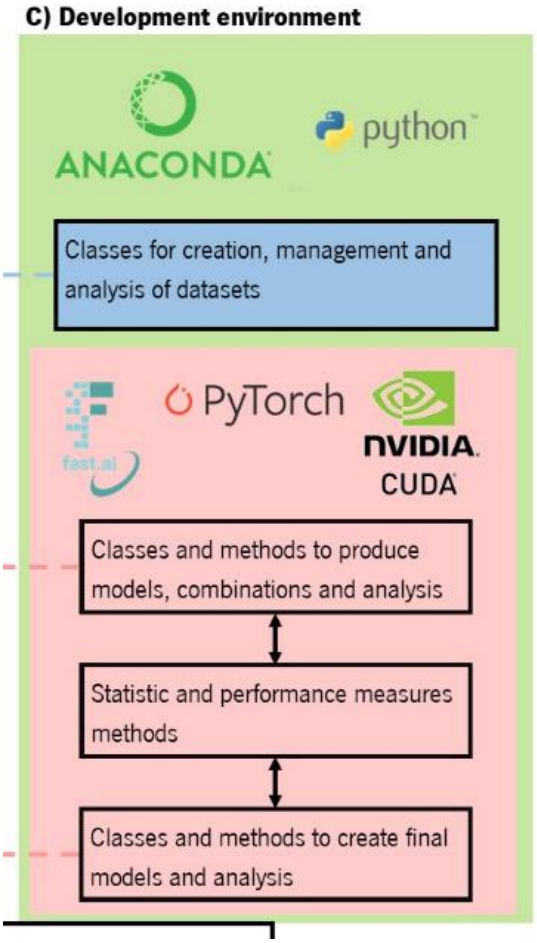

Figure 3.8 – Development environment

To create a model an environment is needed, to create this model the environment, Figure 3.8, is an Anaconda environment running Python, in this environment classes were created in order to create, manage and analyze datasets. Using PyTorch, fast.ai, a Deep Learning library for PyTorch which provides practitioners with high-level components that can quickly and easily provide state-of-the-art results in standard Deep Learning domains, and provides researchers with low-level components that can be mixed and matched to build new approaches [43], and Nvidia CUDA, a parallel computing platform and programming model for computing on graphical processing units [44]. These allow for the creation of classes and methods to produce models, combinations and analysis, statistical and performance measures methods to evaluate and improve the classes and methods, and the creation of the final model through the analysis of the created classes and methods.

3.2.4 PREPROCESSING

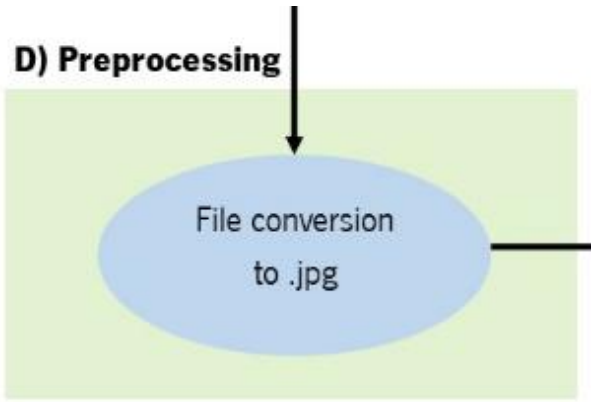

Figure 3.9 – Preprocessing used in the development of this model

There is not a lot of preprocessing needed for the creation of these model, in fact, the only preprocessing needed is to convert the medical images form dcm files into jpg files, Figure 3.9, since the pre-trained model used was trained with jpg files the model trained with these can not predict for dcm files.

3.2.5 DATASET CREATION

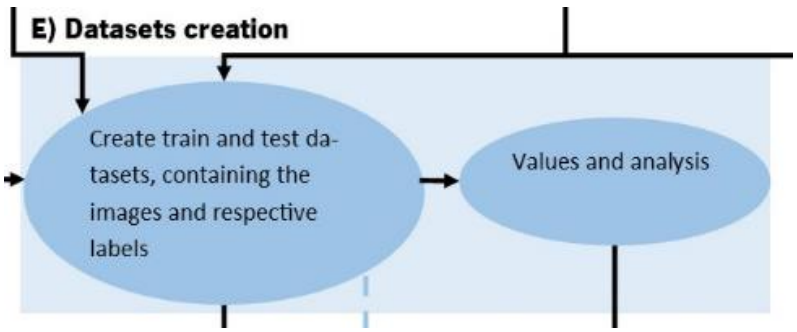

Figure 3.10 – Dataset Creation

To create the training dataset, Figure 3.10, the ImageNet classes are mapped to the word vectors, two lists are created, one with all the names of the classes in the ImageNet database, the other with all English nouns present in the WordNet database, and the id's of each mapped to the respective word vector. With these the training dataset can be formed, which will consist of the images from ImageNet and the word vectors correspondent to the WordNet id's, 99% of the dataset will be used for training while a randomly selected 1% will be used for validation. A second training dataset is also created the same way the first dataset is, however instead of using the ImageNet images it uses a subset of the first X-Ray dataset, consisting of 1000 single label images for each of the seven most frequent conditions and

also 1000 images with no conditions labeled nothing. These datasets are then evaluated by the classes created in the development environment.

3.2.6 CREATED DATA

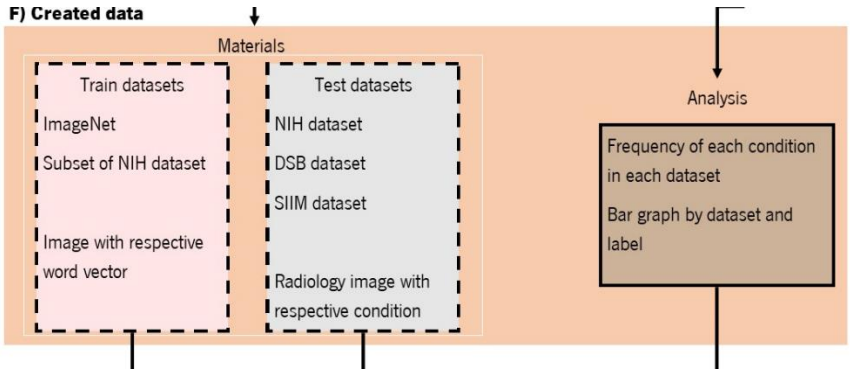

Figure 3.11 – Created data

The created data, Figure 3.11, consists of two training datasets, one using ImageNet images and the other a subset of the first X-Ray dataset, both with the respective word vectors for the labels, and three test datasets of radiology images, two for X-Rays images and another for CT scans, each labeled with the respective conditions. The analysis of the datasets will consist of the frequency of each label in each test dataset displayed as a bar graph and form this information later perform an analysis of the results.

3.2.7 APPLICATION

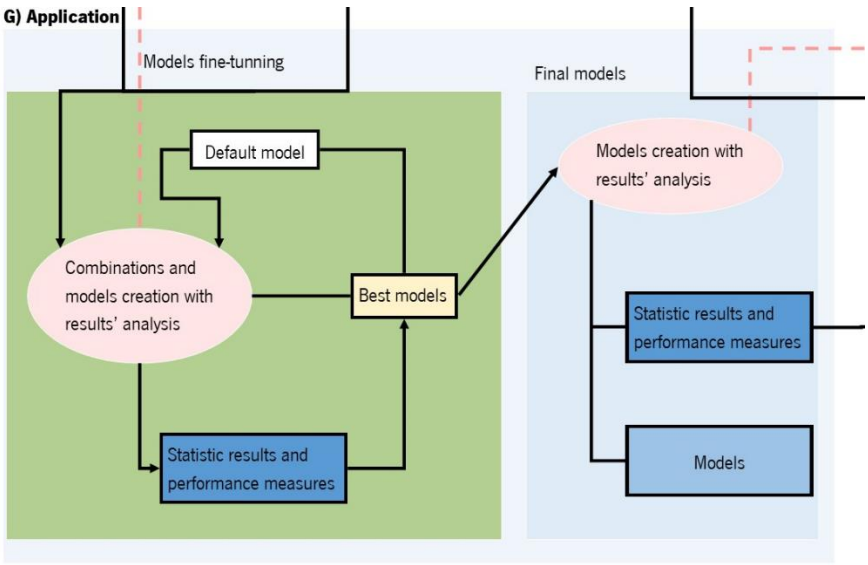

Figure 3.12 – Processes used to tune the models and obtain the final model

The model will be a pre-trained one, from the package torchvision in the PyTorch library, called a Resnet34, this model is, as the name suggests, a residual network of 34 layers, which will be tuned, this process is shown in Figure 3.12, the idea of a residual network is to eliminate the notorious vanishing gradient problem, as the gradient is back-propagated to earlier layers, repeated multiplication may make the gradient infinitely small, as a result, as the network goes deeper, its performance gets saturated or even starts degrading rapidly. They do this with an identity shortcut connection, that skips one or more layers [45]. A representation of an identity shortcut connection is displayed in Figure 3.13.

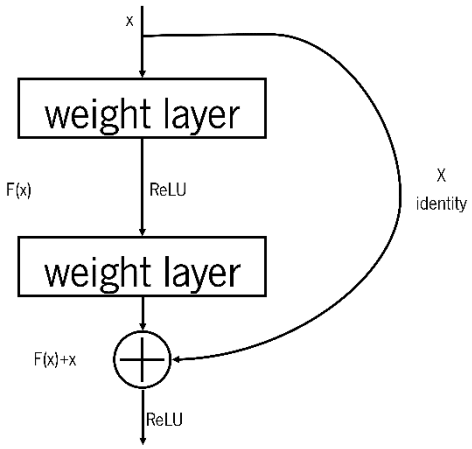

Figure 3.13 – Identity shortcut connection

For the training, the optimizer, which is responsible for update the network weights, is the Adam optimizer, this optimizer instead of adapting the parameter learning rates based on the average first moment (the mean), it also makes use of the average of the second moments of the gradients (the uncentered variance). Specifically, the algorithm calculates an exponential moving average of the gradient and the squared gradient, and the parameters beta1 and beta2 [46], which have been set to 0.9 and 0.99 respectively, control the decay rates of these moving averages. The initial value of the moving averages and beta1 and beta2 values close to 1.0 result in a bias of moment estimates towards zero. This bias is overcome by first calculating the biased estimates before then calculating bias-corrected estimates. As for the loss function, which is responsible for evaluating how the algorithm models the data, in other words how good is the algorithm, if the result of this function is close to zero it means the predictions deviate little from actual results, so the function used to do that will be the cosine similarity, this function calculates the similarity between two vectors, it is define as the cosine between the angle of the vectors, if that angle is 0, meaning the vectors are the same, the result of the function is 1, and if

the angle is π or 180 degrees, meaning the vectors are opposite to each other result of the function is 0, this is the loss function for each training example, but we need the loss function for the whole training, so the function is now the mean of the cosine similarity, however this is a problem, as stated before the loss function should return values close to 0 if the prediction is similar to the actual result, however this function returns values close to 1 if that happens, so to resolve this the true loss function is 1 – mean of the cosine similarity, between the value predicted by the algorithm and the true value.

The architecture of the network can be seen in Figure 3.14, this architecture is divided in 6 blocks, the first block contains a convolutional layer, that takes the input image, this layer has a size of 7x7, input size 224x224, 64 output feature maps, batch normalization and ReLU for activation function, this layer is followed by a MaxPool2d layer with input size 112x112 and 64 output feature maps, the second block has 6 layers all convolutional with size 3x3, input size 56x56, 64 output feature maps, with batch normalization and ReLU for activation function, the third block has 8 layers equal to the ones in the second block, the first with input size 56x56, the following with input size 28x28, all with 128 output feature maps, the fourth block has 12 layers equal to the ones in the second and third blocks, the first with input size 28x28, the following with 14x14, all with 256 output feature maps, the fifth block has 6 layers equal to the previous 3 blocks, the first with input size 14x14, the following layers with input size 7x7, all of the layers have 512 output feature maps, the last block has 3 layers the first is an AdaptativeAvgPool2d, with input size 7x7 and 512 output feature maps, the second layer is a linear one with input size 1x1, 512 output feature maps, batch normalization, dropout and ReLU as the activation function, the last layer is also a linear one, as an input size of 1x1, 300 output feature maps, batch normalization and dropout.

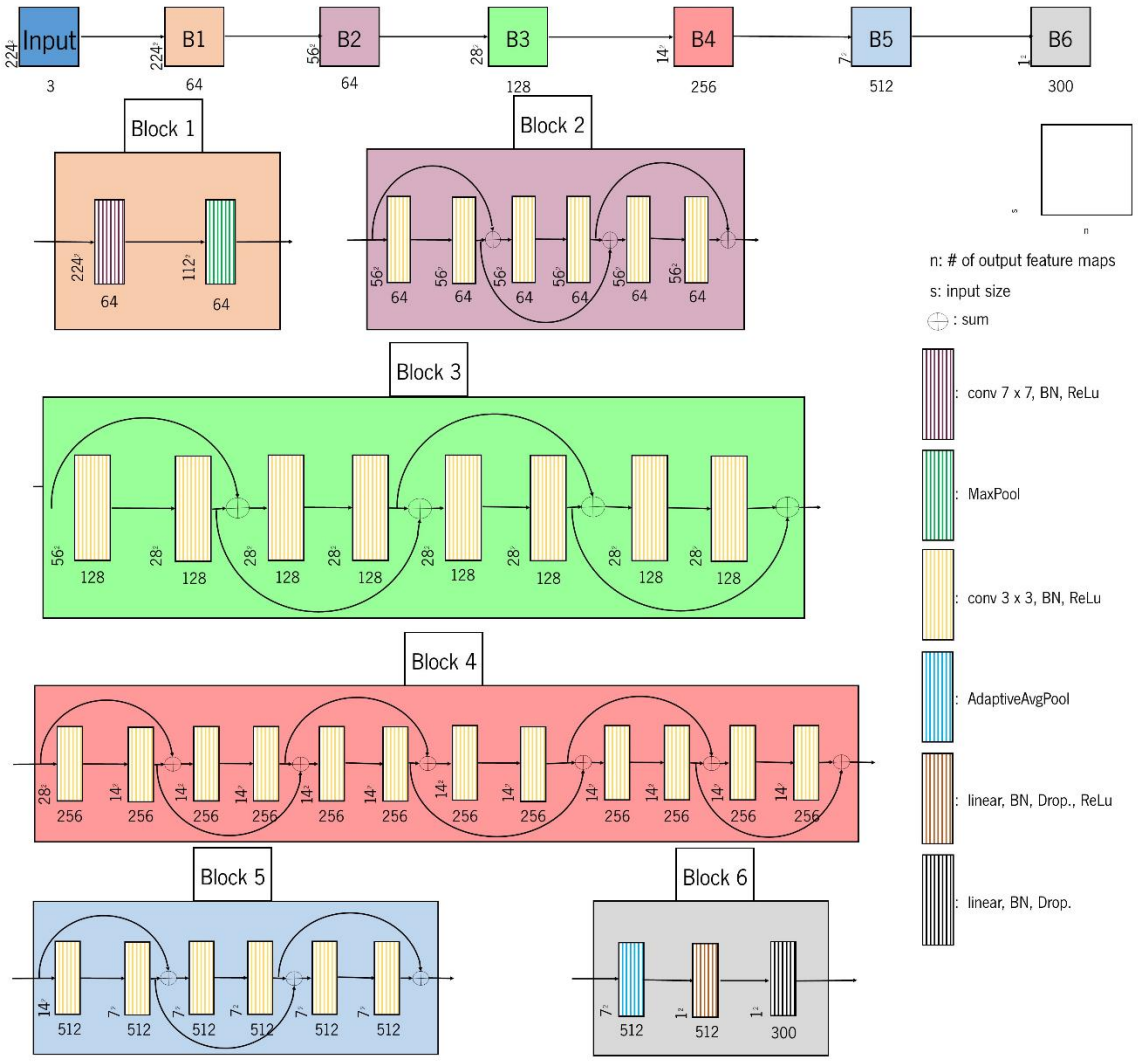

Figure 3.14 – Neural Network architecture

3.3 RESULTS

3.3.1 FIRST X-RAY DATASET

Experiment 1: With ImageNet

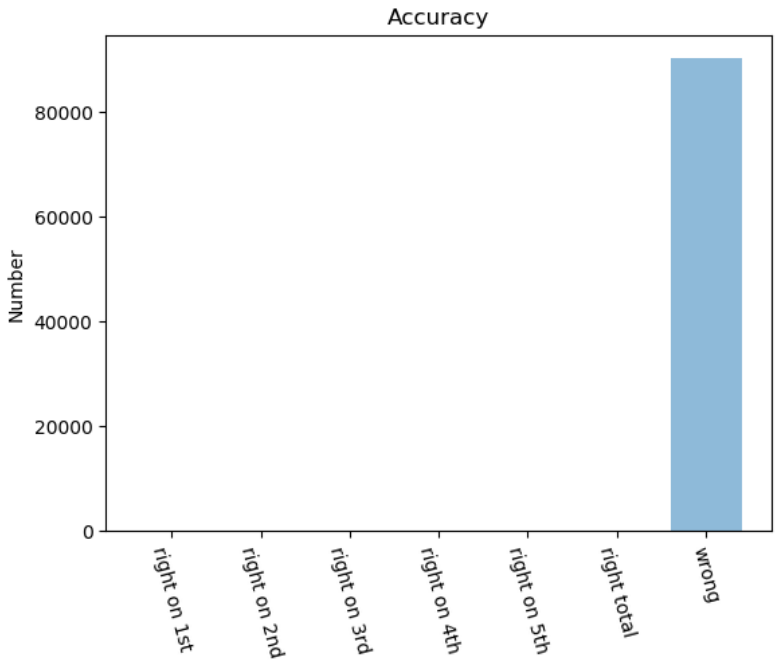

Figure 3.15 – Overall accuracy for Images with one label and without images labeled Pleural_Thickening

As seen in Figure 3.15, with this technique, the model is not capable of predicting conditions correctly. The label Pleural_Thickening was not included in this analyses because that label does not exist in the WordNet database, however since pleural thickening is an increase in the bulkiness of one or both of the pulmonary pleurae, and both thickening and pleura, singular for pleurae, are labels in WordNet, the 1127 images labeled with Pleural_Thickening were checked to see if on the five predictions either one or both of these labels appear, none of these labels appear a single time.

For images with more than one label, the result was the same.

Experiment 2: With subset of first X-Ray Dataset

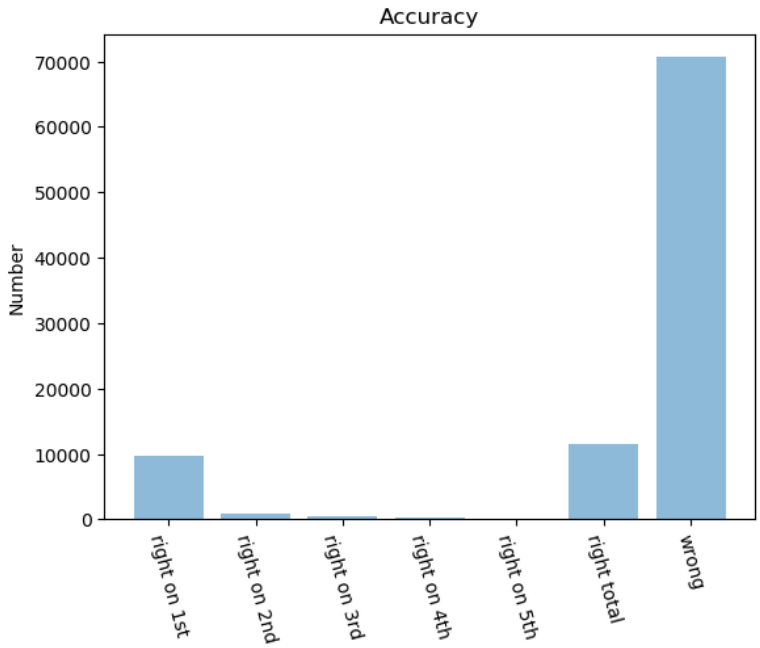

Figure 3.16 – Overall accuracy for Images with one label and without images labeled Pleural_Thickening

As seen in Figure 3.16, with this technique, the model is capable of predicting some conditions correctly, as seen in the image most correct predictions occur in the first prediction, 9719 were corrected in the first prediction follow by 946 correct predictions in the second prediction, 512 in the third prediction, 288 in the fourth prediction and 149 in the fifth prediction, for a total of 11614 correct predictions and 70644 wrong predictions meaning that for this scenario the algorithm has an accuracy rate of approximately 14.2% and of those about 83.7% are correct on the first prediction. The label Pleural_Thickening was not included in this analyses because that label does not exist in the WordNet database, however since pleural thickening is an increase in the bulkiness of one or both of the pulmonary pleurae, and both thickening and pleura, for pleurae, are labels in WordNet, the 1127 images labeled with Pleural_Thickening were checked to see if, on the five predictions either one or both of these labels appear, one these images had the prediction pleura in the first five predictions and 5 others had the label thickening in the first five predictions, none of the images had both labels. The accuracy for each individual label is seen next.

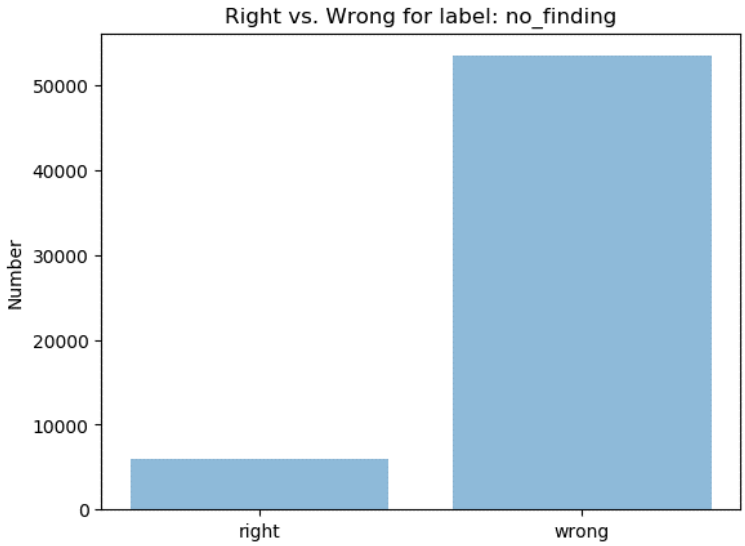

Figure 3.17 – Right vs. Wrong for label: no_finding

As seen in Figure 3.17 for the images labeled with no_finding which was used for the training, the model correctly predicted "nothing" for 5948 images while 53464 images were not labeled correctly, meaning for this case the model as an accuracy of approximately 10%

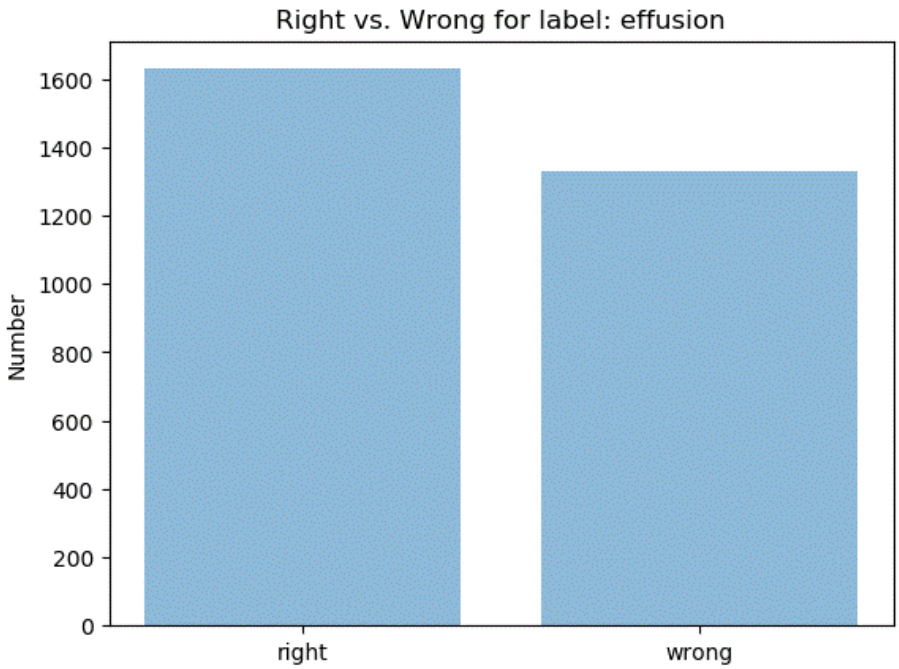

Figure 3.18 – Right vs. Wrong for label: effusion

As seen in Figure 3.18 for the images labeled with effusion which was used for the training, the model correctly predicted 1629 images while 1330 images were not labeled correctly, meaning for this case the model as an accuracy of approximately 55.1%

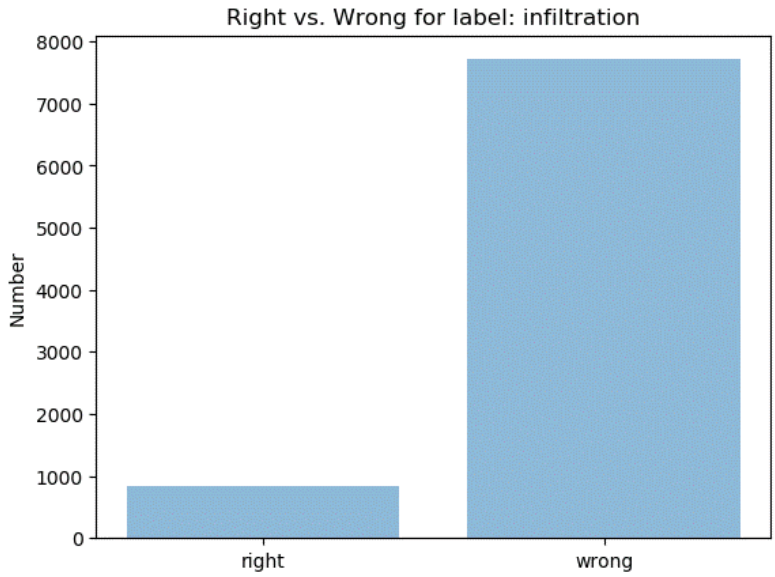

Figure 3.19 – Right vs. Wrong for label: infiltration

As seen in Figure 3.19 for the images labeled with infiltration which was used for the training, the model correctly predicted 841 images while 7710 images were not labeled correctly, meaning for this case the model as an accuracy of approximately 9.8%

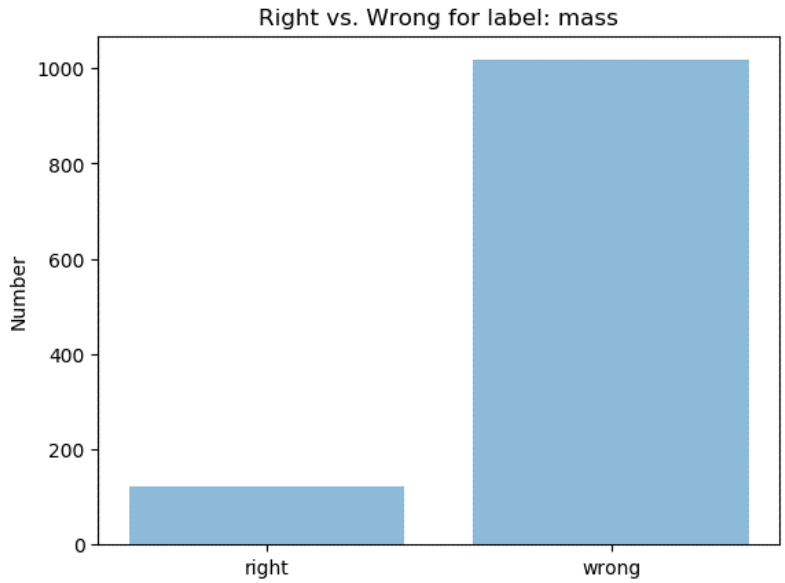

Figure 3.20 – Right vs. Wrong for label: mass

As seen in Figure 3.20 for the images labeled with mass which was used for the training, the model only correctly predicted 121 images while 1017 images were not labeled correctly, meaning for this case the model as an accuracy of approximately 10.6%

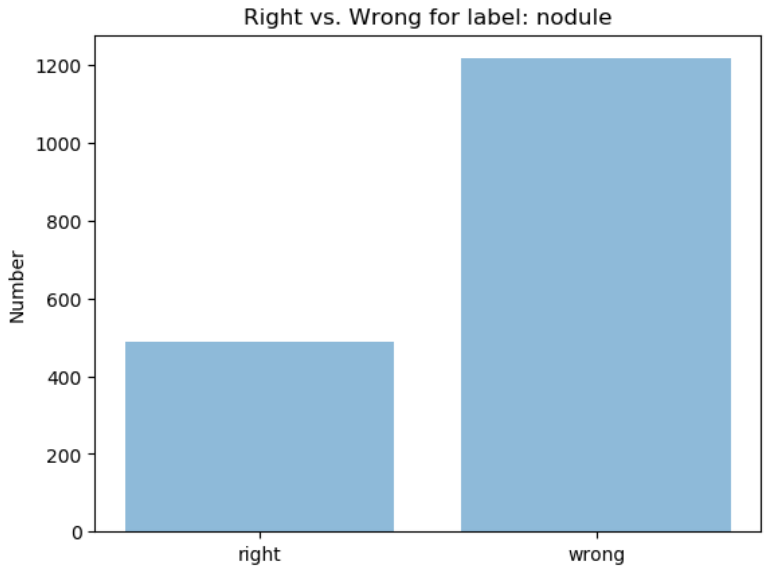

Figure 3.21 – Right vs. Wrong for label: mass

As seen in Figure 3.21 for the images labeled with nodule which was used for the training, the model correctly predicted 489 images while 1217 images were not labeled correctly, meaning for this case the model as an accuracy of approximately 28.7%

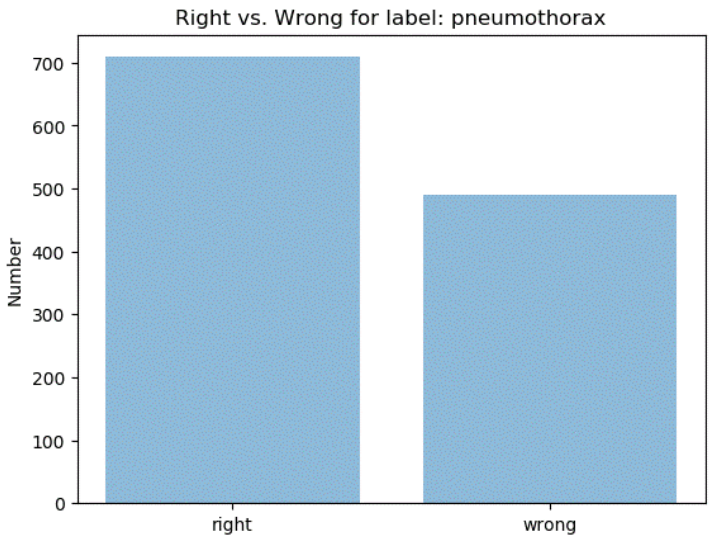

Figure 3.22 – Right vs. Wrong for label: pneumothorax

As seen in Figure 3.22 for the images labeled with pneumothorax which was used for the training, the model correctly predicted 709 images while 409 images were not labeled correctly, meaning for this case the model as an accuracy of approximately 59.1%

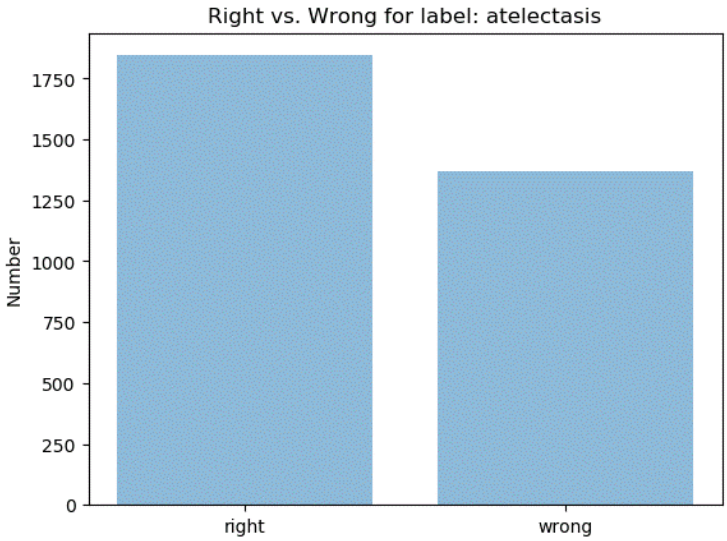

Figure 3.23 – Right vs. Wrong for label: atelectasis

As seen in Figure 3.23 for the images labeled with atelectasis which was used for the training, the model correctly predicted 1845 images while 1367 images were not labeled correctly, meaning for this case the model as an accuracy of approximately 57.4%

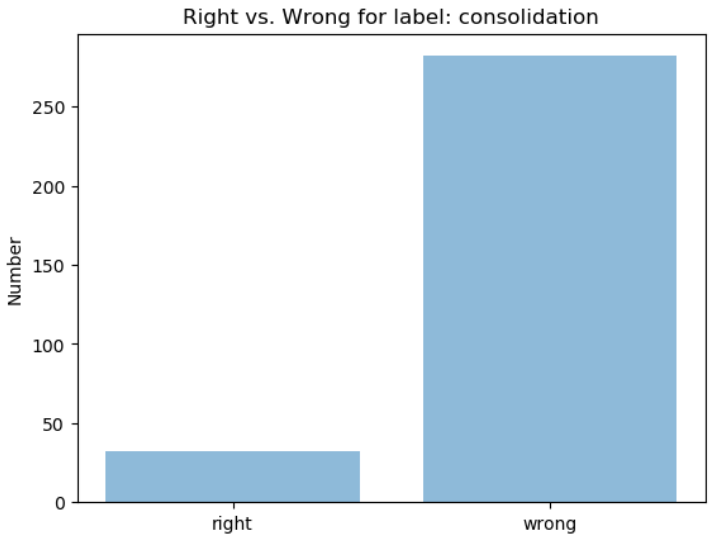

Figure 3.24 – Right vs. Wrong for label: consolidation

As seen in Figure 3.24 for the images labeled with consolidation, which was used for the training, the model correctly predicted 32 images while 282 images were not labeled correctly, meaning for this case the model as an accuracy of approximately 10.2%

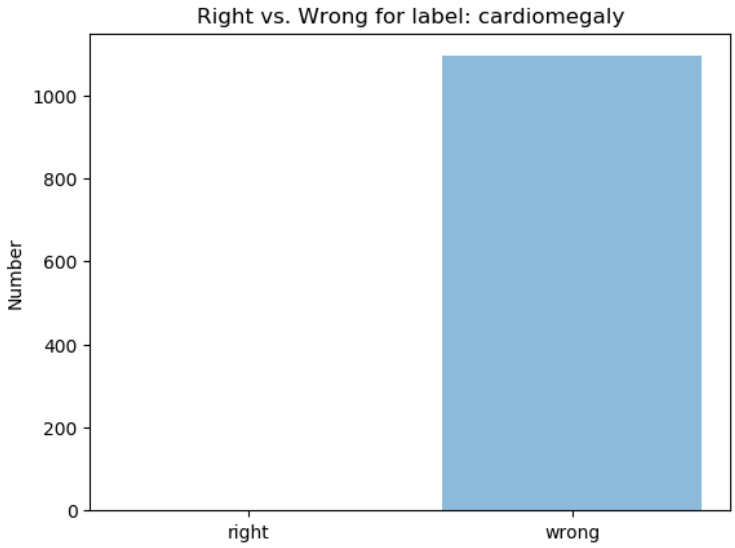

Figure 3.25 – Right vs. Wrong for label: cardiomegaly

As seen in Figure 3.25 for the images labeled with cardiomegaly, which was not used for the training, the model is not able to correctly predict a single image, all 1094 images were labeled incorrectly. All other labels that were not used in the training have the same results, with the label edema there's a slight difference because although the images labeled with edema none was predicted with edema in the top five categories some other images were predicted with edema in the top five categories.

For images with multiple labels, there are 20735 images with more than one label for a total of 47757 labels, for this experiment the results are as follows.

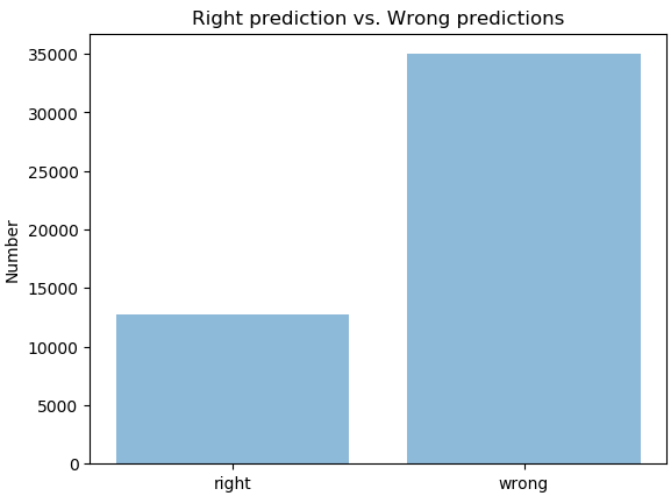

Figure 3.26 – Right vs. Wrong predictions for images with multiple labels except for label pleural_thickening

As seen in Figure 3.26 the model correctly predicts 12776 labels while 34981 labels are incorrectly predicted, meaning overall an accuracy of about 26.8%, for the label pleural_thickening, the same

approach used for the images with a single label was used here, in this case of the 2258 images labeled as such, 14 were predicted as thickening, none was predicted has either pleura or with both labels.

For specific conditions, the results were as follows.

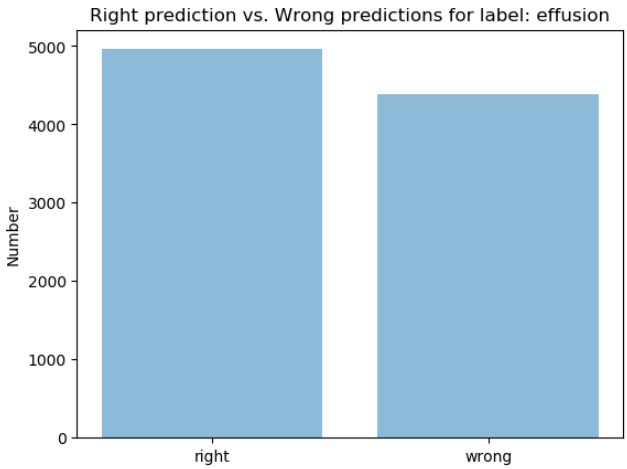

Figure 3.27 – Right vs. Wrong predictions for label: effusion

As seen in Figure 3.27, for the images labeled with effusion, which was a train category, there were 4959 images predicted correctly while 4389 were predicted wrongly, this means an accuracy of about 53%.

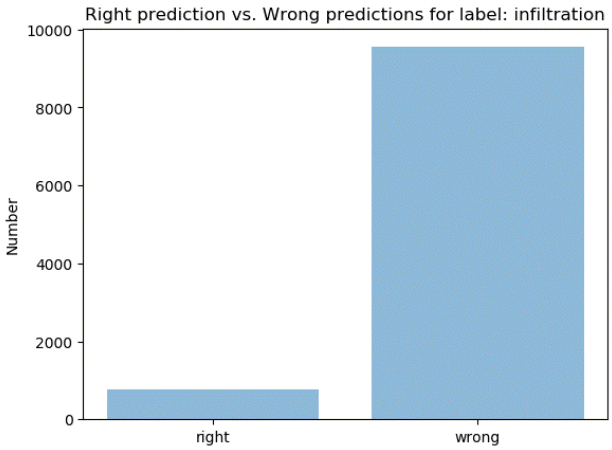

Figure 3.28 – Right vs. Wrong predictions for label: infiltration

As seen in Figure 3.28, for the images labeled with infiltration, which was a train category, there were 770 images predicted correctly while 9549 were predicted wrongly, this means an accuracy of about 7.46%.

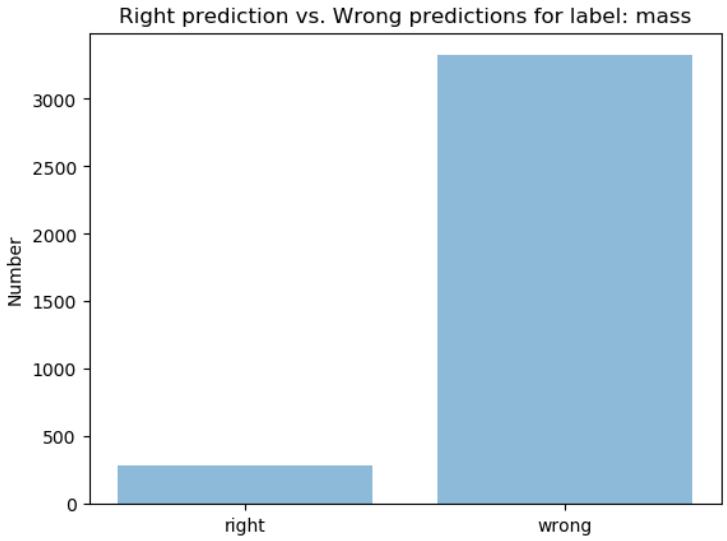

Figure 3.29 – Right vs. Wrong prediction for label: mass

As seen in Figure 3.29, for the images labeled with mass, which was a train category, there were 286 images predicted correctly while 3322 were predicted wrongly, this means an accuracy of about 7.93%.

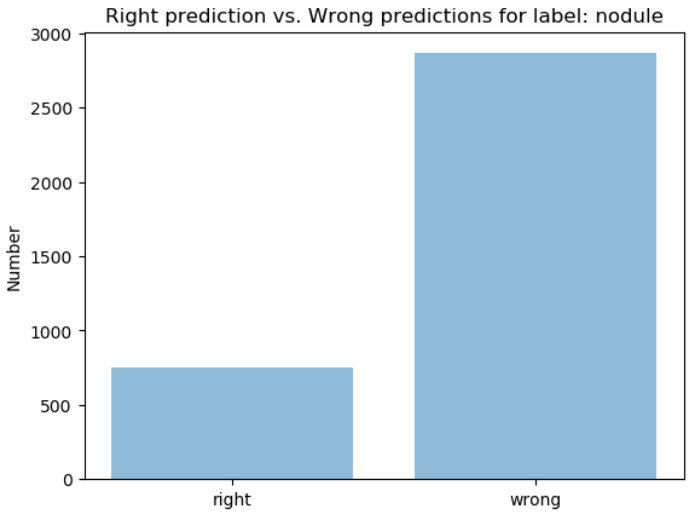

Figure 3.30 – Right vs. Wrong predictions for label: nodule

As seen in Figure 3.30, for the images labeled with nodule, which was a train category, there were 753 images predicted correctly while 2865 were predicted wrongly, this means an accuracy of about 20.8%.

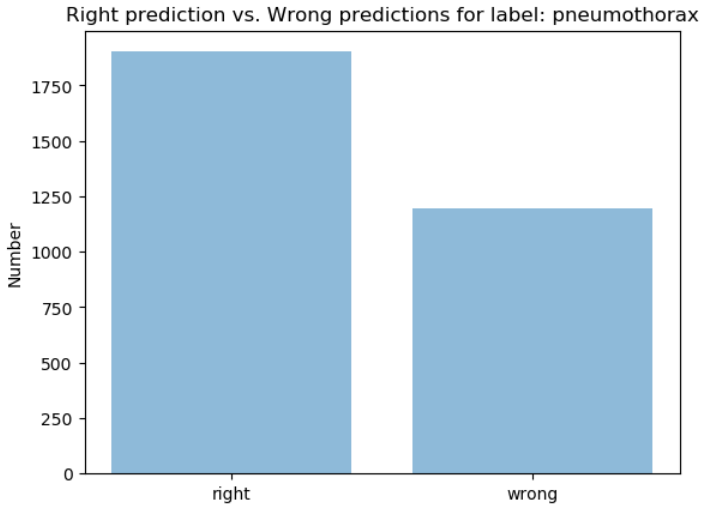

Figure 3.31 – Right vs. Wrong predictions for label: pneumothorax

As seen in Figure 3.31, for the images labeled with pneumothorax, which was a train category, there were 1902 images predicted correctly while 1197 were predicted wrongly, this means an accuracy of about 61.4%.

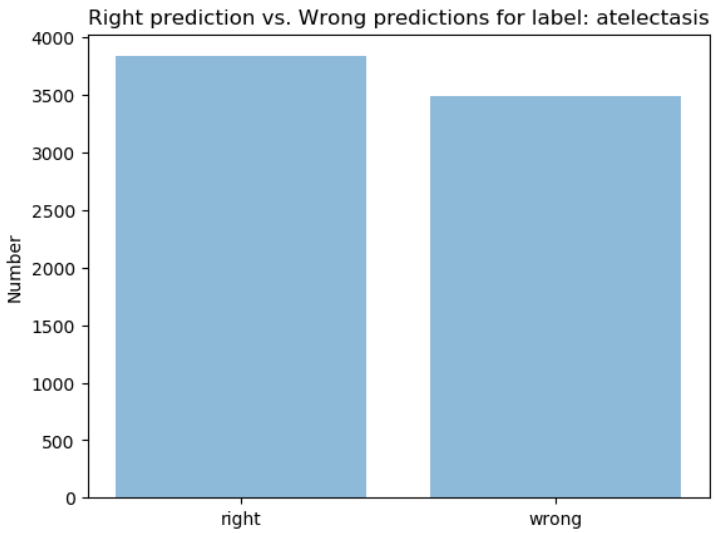

Figure 3.32 – Right vs. Wrong predictions for label: atelectasis

As seen in Figure 3.32, for the images labeled with atelectasis, which was a train category, there were 3836 images predicted correctly while 3487 were predicted wrongly, this means an accuracy of about 52.4%.

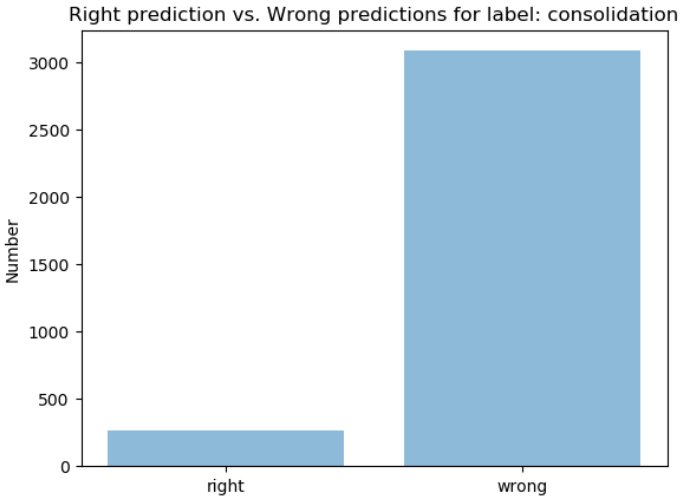

Figure 3.33 – Right vs. Wrong predictions for label: consolidation

As seen in Figure 3.33, for the images labeled with consolidation, which was a train category, there were 286 images predicted correctly while 3086 were predicted wrongly, this means an accuracy of about 8.48%.

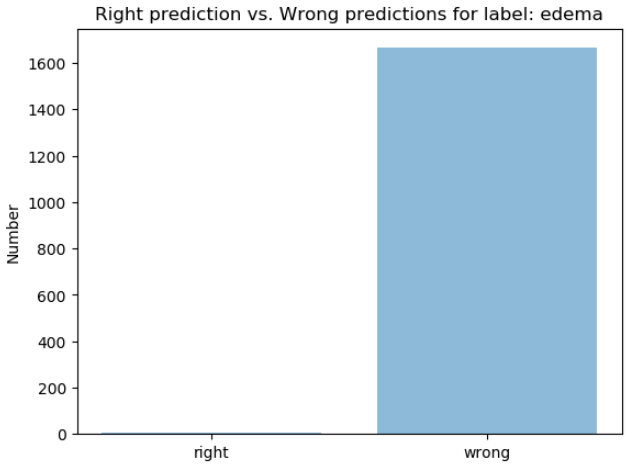

Figure 3.34 – Right vs. Wrong predictions for label: edema

As seen in Figure 3.34, for the images labeled with edema, which was not a train category, there were 4 images predicted correctly while 1665 were predicted wrongly, this means an accuracy of about 0.24%. All other labels that were not part of the training dataset have the same results as the instances of single label images.

3.3.2 SECOND X-RAY DATASET

Experiment 1: With ImageNet

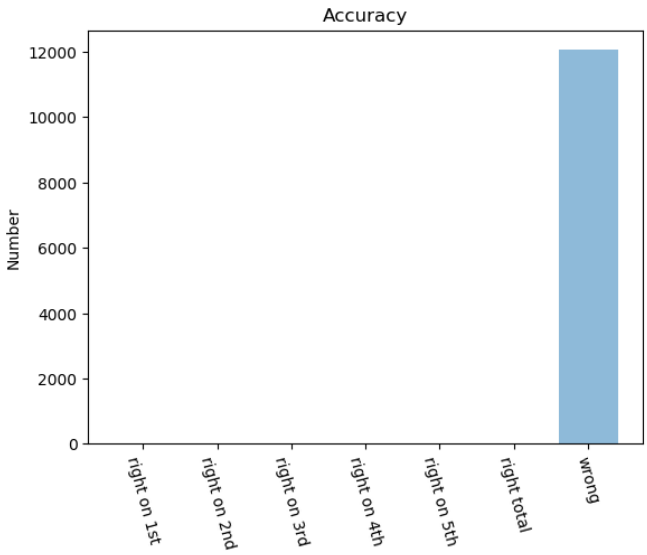

Figure 3.35 – Overall accuracy for images of the second X-Ray

As seen in Figure 3.35, with this technique, the model also is not capable of predicting instances correctly, all 12047 X-Rays images were not predicted correctly.

Experiment 2: With Subset of X-Ray Dataset

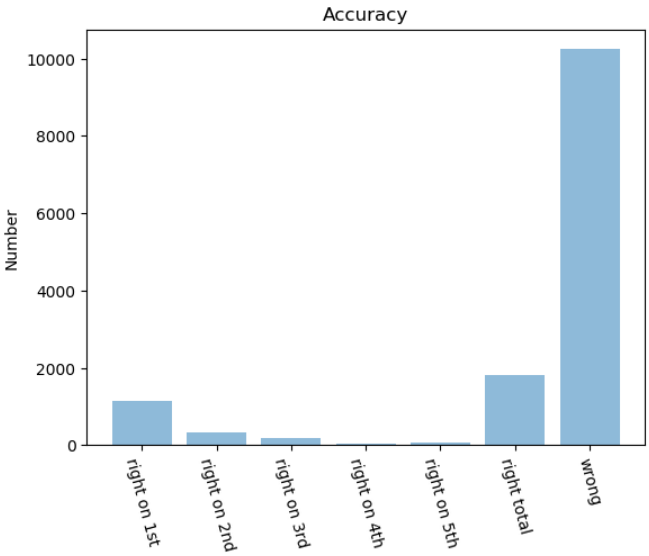

Figure 3.36 – Overall accuracy made by the model images of the second X-Ray

As seen in Figure 3.36, with this technique, the model is capable of predicting some instances correctly, as seen in the image most correct predictions occur in the first prediction, 1150 were correct in the first prediction follow by 347 correct predictions in the second prediction, 189 in the third prediction, 53 in the fourth prediction and 67 in the fifth prediction, for a total of 1806 correct predictions and 10241 wrong predictions meaning that for this scenario the algorithm has an accuracy rate of approximately 15.9% and of those about 63.7% are correct on the first prediction.

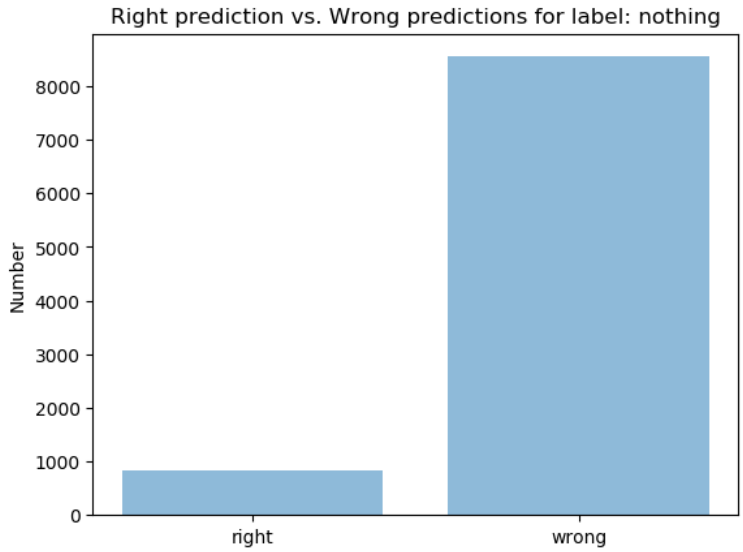

Figure 3.37 – Right vs. Wrong predictions for label: nothing

As seen in Figure 3.37, for the images labeled with nothing, which was a train category, there were 828 images predicted correctly while 8550 were predicted wrongly, this means an accuracy of about 8.83%.

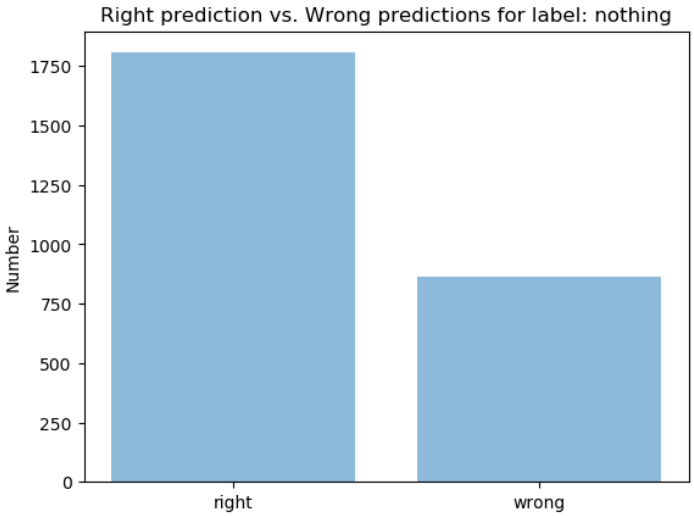

Figure 3.38 – Right vs. Wrong predictions for label: pneumothorax

As seen in Figure 3.38, for the images labeled with pneumothorax, which was a train category, there were 1806 images predicted correctly while 863 were predicted wrongly, this means an accuracy of about 67.7%.

3.3.3 CT DATASET

Experiment 1: With ImageNet

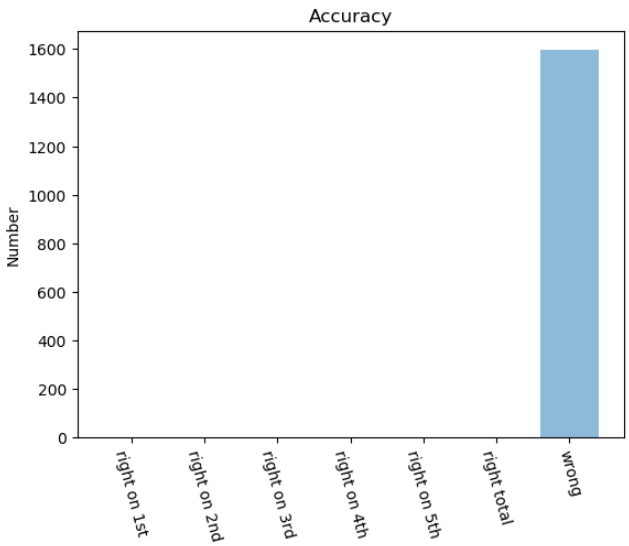

Figure 3.39 – Overall accuracy for CT scans

As seen in Figure 3.39, with this technique, the model is not capable of predicting instances correctly, all 1595 CT scans were not predicted correctly.

E 2: With Subset of X-Ray Dataset

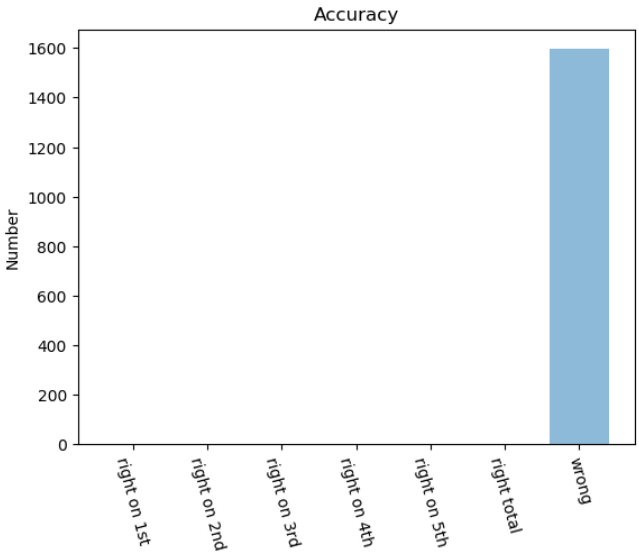

Figure 3.40 – Overall accuracy of the predictions made by the model for the CT scans

As seen in Figure 3.40, with this technique, the model also is not capable of predicting instances correctly, all 1595 CT scans were not predicted correctly.

3.3.4 SYNTHESIS

The Table 3.1 shows the synthesis of the results by modalities and experiments

Table 3.1 – Synthesis of the results

Experiences Modalities	Experiment 1: ImageNet Dataset	Experiment 2: Subset of X-Ray dataset
First X-Ray	Unable to correctly predict the condition(s) of a single image	Correctly predicted 11614 conditions for the 82258 images with a single label. Correctly predicted 12776 labels out of the 47757 labels present in images with multiple labels

Second X-Ray	Unable to correctly predict the condition of a single image	Correctly predicted the condition of 1806 images out of the 12407
CT scan	Unable to correctly predict the condition of a single image	Unable to correctly predict the condition of a single image

4 Conclusions

This work had as its objective the construction of a Machine Learning model capable of predicting medical conditions present in medical image exams, regardless if the model was trained for that specific condition or not, for that it uses labeled images, both generic images and radiology specific images, as is trained dataset as well as word vectors to determine how semantically close the words are to each other, the model was then tested in 3 different radiology datasets, two of X-Ray and one of CT.

4.1 VISUAL SEMANTIC EMBEDDING FOR MEDICAL IMAGING

The idea of this work was to create an algorithm capable of correctly predicting medical conditions present in a radiology image, more specifically X-Rays and CT scans, regardless if those conditions are or not part of a training dataset, to accomplish that labeled images were used, some more generic images, ImageNet database, and some specific to radiology and medical conditions, subset of NIH X-Rays of 8000 images for seven conditions, 1000 each plus 1000 for healthy patients labeled as "nothing", and word vectors, from the fastText library, that indicate how semantically close each word is to every other in a high dimensional vector space, from the results is observable that in the experiment with the ImageNet database, generic images, the model is not capable of predicting any of the conditions tested for any of the datasets, in the second experience, with a subset of the NIH X-Ray dataset, specific radiology images, in this case the results are not perfect or even good, although the algorithm can predict with some reliability the conditions that it was train with, from the seven conditions, present in the X-Ray dataset, that it was not train with it was only capable of predicting two and with a very low success rate, for the second X-Ray dataset it is able to predict the condition present in those images, however that was a train category; and for the condition present in the CT scan, lung cancer, it is not able to predicted, this is due to the lack of training data and the word vectors not relating the words in a pure medical sense, making the extraction of attribute features very limited, never the less this shows that it is possible to use a Machine Learning algorithm to correctly predict medical conditions that it was not train with, this algorithm can, obviously, be improved to do so the steps mentioned in the next subsection, Future Work, should be followed.

4.2 FUTURE WORK

As seen in the results this model can, although rarely, predict medical conditions which it was not trained with, even for conditions that the model was trained with the results can improve significantly, to achieve better performance for the train categories as well as for the categories not present in the training dataset,

two major steps need to be performed, the first being the creation of an updated dataset for several medical conditions, as well as images with no conditions, with each condition being represented with several images, about 1000 images per condition should be fine, this way the model has more information to make the connections to conditions that it was not trained with more precision, the second step is to create the word vectors for all known medical conditions, using only medical terminology, in other words, in medical terms alone how close each word is to every other word, these word vectors also should include a label specific to be used in images without any conditions. With these steps performed the model will be capable of extracting more attribute features and doing so more precisely, making the model perform better.

REFERENCES

[1] D. L. Hill, P. G. Batchelor, M. Holden and D. J. Hawkes, "Medical Image Registration," *Physics in Medicine & Biology,* vol. 46, pp. 1-45, 2001.

[2] NHS England, "Diagnostic Imaging Dataset," November 2018. [Online]. Available: https://www.england.nhs.uk/statistics/statistical-work-areas/diagnostic-imaging-dataset/. [Accessed 17 April 2020].

[3] J. Benseler, "A Pocket Guide to Medical Imaging," in *The Radiology Handbook*, Ohio, Ohio University Press, 2006.

[4] World Health Organization, "Density of physicians (total number per 1000 population, latest available year)," 2019. [Online]. Available: https://www.who.int/gho/health_workforce/physicians_density/en/. [Accessed 19 April 2020].

[5] D. Ballard and J. Sklansky, "Tumor detection in radiographs," *Computers and Biomedical Research,* vol. 6, pp. 299-321, August 1973.

[6] B. Yaniv, I. Diamant, L. Wolf, S. Lieberman, E. Konen and H. Greenspan, "Chest pathology detection using deep learning with non-medical training," in *IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, New York, 2015.

[7] P. Lakhani and B. Sundaram, "Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks," *Radiology,* vol. 284, 2017.

[8] I. Pan, S. Agarwal and D. Merck, "Generalizable Inter-Institutional Classification of Abnormal Chest Radiographs Using Efficient Convolutional Neural Networks," *Journal of Digital Imaging,* pp. 1-9, 2019.

[9] J. Kleesiek, A. Biller, G. Urban, U. Kothe, M. Bendszus and F. Hamprecht, "Ilastik for multi-modal brain tumor segmentation," *Proceedings MICCAI BraTS (Brain Tumor Segmentation Challenge),* pp. 12-17, 2014.

[10] D. Zikic, B. Glocker, E. Konukoglu, A. Criminisi, C. Demiralp, J. Shotton, O. M. Thomas, T. Das, R. Jena and S. J. Price, "Decision forests for tissue-specific segmentation of high-grade gliomas

in multi-channel MR," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2012, pp. 369-376.

[11] S. Pereira, A. Pinto, A. Victor and C. A. Silva, "Brain tumor segmentation using convolutional neural networks in MRI images," *IEEE transactions on medical imaging,* vol. 35, pp. 1240-1251, 2016.

[12] F. Isensee, P. Kickingereder, W. Wolfgang, M. Bendszus and K. H. Maier-Hein, "Brain tumor segmentation and radiomics survival prediction: Contribution to the brats 2017 challenge," in *International MICCAI Brainlesion Workshop*, Springer, 2017, pp. 287-297.

[13] National Cancer Institute (NIH), "NCI Dictionary of Cancer Terms," [Online]. Available: https://www.cancer.gov/publications/dictionaries/cancer-terms/def/soft-tissue?redirect=true. [Accessed 23 July 2020].

[14] Z. Akkus, B. Erickson, A. Galimzianova, A. Hoogi and D. Rubin, "Deep Learning for Brain MRI Segmentation: State of the Art and Future Directions," *J Digit Imaging.,* vol. 30, pp. 449-459, Aug 2017.

[15] B. Lin, K. Michael, S. Kalra and H. Tizhoosh, "Skin lesion segmentation: U-nets versus clustering," in *2017 IEEE Symposium Series on Computacional Intelligence (SSCI)*, 2017.

[16] A. Setio, A. Traverso, T. Bel, M. Berens, C. v. d. Bogaard, P. Cerello, H. Chen, Q. Dou, M. Fantacci, B. Geurts and et al., "Validation, comparison, and combinaion of algorithms for automatic detection of pulmonary nodules in computed tomography images: the luna16 challenge," *Medical image analysis,* vol. 42, pp. 1-13, 2017.

[17] D. Ting, C. Cheung and et. al, "Development and validation of a deep learning system for diabetic retinopathy and related eye diseases using retinal images from multiethnic populations with diabetes," *JAMA,* vol. 318, p. 2211–2223, 2017.

[18] A. Esteva, B. Kuprel, R. Novoa, J. Ko, S. Swetter, H. Blau and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature,* vol. 542, pp. 115-118, 2017.

[19] S. Sinha, "What is Zero Shot Learning," 2018.

[20] S. T. March and V. C. Storey, "Design science in the information systems discipline: an introduction to the special issue on design science research," *MIS quarterly,* vol. 32, no. 4, pp. 725-730, 2008.

[21] K. Peffers, T. Tuunanen, M. A. Rothenberger and S. Chatterjee, "A design science research methodology for information systems research," *Journal of management information systems,* vol. 24, no. 3, pp. 45-77, 2007.

[22] M. G. M.-H. G. Roobottom CA, "Radiation-reduction strategies in cardiac computed tomographic angiography," *Radiation-reduction strategies in cardiac computed tomographic angiography,* November 2010.

[23] Physics Central, "MRI Magic," [Online]. Available: https://www.physicscentral.com/explore/action/mri.cfm. [Accessed 12 March 2020].

[24] T. Jain, "Basics of Image Classification Techniques in Machine Learning.".

[25] Guru99, "Supervised Machine Learning: What is, Algorithms, Example," [Online]. Available: https://www.guru99.com/supervised-machine-learning.html. [Accessed 17 July 2020].

[26] Guru99, "Unsupervised Machine Learning: What is, Algorithms, Example," [Online]. Available: https://www.guru99.com/unsupervised-machine-learning.html. [Accessed 20 July 2020].

[27] SAS, "Deep Learning. What it is and why it matters," [Online]. Available: https://www.sas.com/en_us/insights/analytics/deep-learning.html. [Accessed 20 July 2020].

[28] Guru99, "Deep Learning Tutorial for Beginners: Neural Network Classification," [Online]. Available: https://www.guru99.com/deep-learning-tutorial.html. [Accessed 20 July 2020].

[29] D. Radečić, "Softmax Activation Function Explained".

[30] A. Agrawal, *Loss Functions and Optimization Algorithms. Demystified.,* 29 September 2017.

[31] SAS, "Optimization Algorithms," [Online]. Available: https://documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=v_006&docsetId=casdlpg&docsetTarget=p1i430hb7ri1fnn1511qu227wtbw.htm&locale=en. [Accessed 19 May 2019].

[32] S. Lau, *Learning Rate Schedules and Adaptive Learning Rate Methods for Deep Learning,* 29 July 2017.

[33] S. Noor, "The Convolution Parameters Calculation," [Online]. Available: https://medium.com/@shahzaibnoor40/the-convolution-parameters-calculation-b2394da8dd59. [Accessed 9 June 2020].

[34] Guru99, "Reinforcement Learning: What is, Algorithms, Applications, Example," [Online]. Available: https://www.guru99.com/reinforcement-learning-tutorial.html. [Accessed 20 July 2020].

[35] J. P. I. M. G.-W. E. Rueckert, "Adapting Brain Signals With Reinforcement Learning Stratagies for Brain Computer Interfaces," 2017.

[36] A. R. P. Potash, "Convergence of Q-learning: a simple proof," 2016.

[37] F. M. Graetz, "DeViSE Zero-shot learning".

[38] L. Zhang, T. Xiang and S. Gong, "Queen Mary University of London". *Learning a Deep Embedding Model for Zero-Shot Learning..*

[39] Y. Fu, T. Xiang, Y. Jiang, X. Xue, L. Sigal and S. Gong, "Recent Advances in Zero-shot Recognition".

[40] M. Buche, S. H. and F. Jurie. *Improving Semantic Embedding Consistency by Metric Learning for Zero-Shot Classification..*

[41] P. C. o. T. Suns, "Getting started with Zero-Shot Learning," [Online]. Available: https://zhuanlan.zhihu.com/p/34656727. [Accessed 12 June 2020].

[42] A. Fome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato and T. Mikolov, "DeViSE: A Deep Visual-Semantic Embedding Model".

[43] fast.ai, "About fastai," [Online]. Available: https://docs.fast.ai/#About-fastai. [Accessed 23 June 2020].

[44] NVidai, [Online]. Available: https://developer.nvidia.com/cuda-zone. [Accessed 23 June 2020].

[45] V. Fung, "An Overview of ResNet and its Variants," 15 Junho 2017.

[46] B. J., *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning,* 3 July 2017.