Deep Learning Methods for TB Diagnosis

Mouhamed Lamine Ngom (mouhamed@aims.ac.za)

African Institute for Mathematical Sciences (AIMS)

Supervised by: Dr. Bubacarr Bah
AIMS South Africa and Stellenbosch University
Co-supervised by: Dr. Habiboulaye AMADOU BOUBACAR
Air Liquide, Paris, France

14 May 2020

Submitted in partial fulfillment of a structured masters degree at AIMS South Africa

Abstract

Tuberculosis is one of the top ten causes of death worldwide. Every year several people die including children because of this disease. This study aims to provide a diagnostic method to give a small contribution to the fight against Tuberculosis and thus save millions of lives. The study attempted to solve this problem by classifying chest X-ray images into Tuberculosis and non-Tuberculosis classes.

During the study, various techniques and deep learning methods were implemented.Different Convolutional Neural Networks (CNNs) models have been established and existent CNNs architecture, posed. Then, the implementation of Attention Mechanism in CNNs models and architecture. Finally, presentation of an algorithm which minimizes false negatives and improve prediction by combining all of this methods and techniques.

All the models were trained and tested on two Tuberculosis chest X-ray images datasets and most of them got performance above 95% in training and above 90% in testing.

Declaration

I, the undersigned, hereby declare that the work contained in this research project is my original work, and that any work done by others or by myself previously has been acknowledged and referenced accordingly.

Mouhamed Lamine Ngom, 14 May 2020

Contents

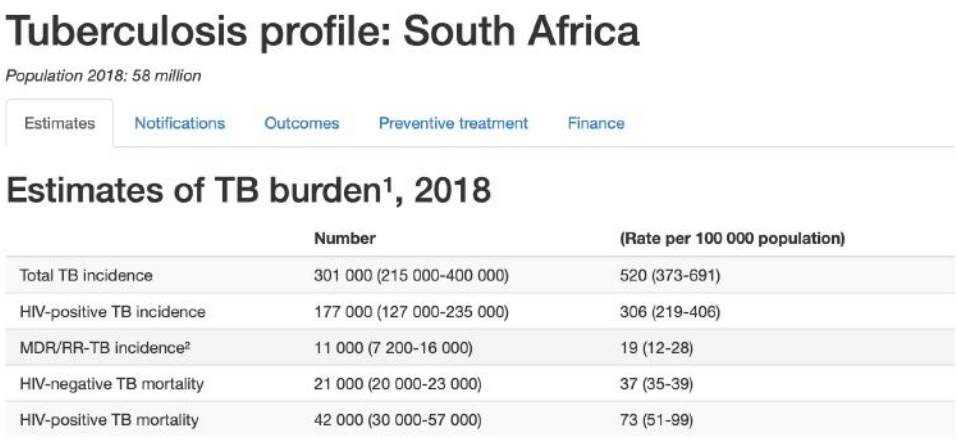
1. Introduction

Tuberculosis is an infectious and contagious disease that often affects the lungs Maladie Infectieuse. Caused by the bacteria Mycobacterium Tuberculosis, when a person with pulmonary tuberculosis coughs, sneezes or spits, he projects tuberculosis bacilli into the air. Just inhale a few to get the disease.

About a quarter of the world's population has latent tuberculosis, which means that these people are not yet sick and cannot also transmit the disease to someone else, although it is true that they are infected with the tubercle bacillus. However, the risk that they will develop the disease during their lifetime is 5 to 15% according to the World Health Organization (WHO) Tuberculosis.

There is a much greater risk than that for people with compromised immune systems, such as people living with HIV, those who are malnourished, diabetics and tobacco users.

In 2018, 10 million people contracted tuberculosis and 1.5 million of them died from it, including 251,000 people living with HIV. Tuberculosis is the number one killer of HIV-positive people, according to the WHO Tuberculosis. In fact, the WHO also declared that in 2018 1.1 million children had tuberculosis and that 251,000 died from it, including children with tuberculosis associated with HIV.



Tuberculosis profile: South Africa

Population 2018: 58 million

Estimates | Notifications | Outcomes | Preventive treatment | Finance

Estimates of TB burden[1], 2018

	Number	(Rate per 100 000 population)
Total TB incidence	301 000 (215 000-400 000)	520 (373-691)
HIV-positive TB incidence	177 000 (127 000-235 000)	306 (219-406)
MDR/RR-TB incidence[2]	11 000 (7 200-16 000)	19 (12-28)
HIV-negative TB mortality	21 000 (20 000-23 000)	37 (35-39)
HIV-positive TB mortality	42 000 (30 000-57 000)	73 (51-99)

Figure 1.1: Tuberculosis profile in South Africa
Source: https://worldhealthorg.shinyapps.io/tb_profiles/?_inputs_&lan=%22EN%22&iso2=%22ZA%22&main_tabs=%22est_tab%22

Globally we see that the continents most affected by tuberculosis are Africa and Asia see Figure 1.2 with South Africa and India at the head of the countries most affected in the world. Here Tuberculosis spread by countries is a link where you can visualize the Tuberculosis profile of a country.

The Figure **??** is the data containing the current profile of South Africa about Tuberculosis.

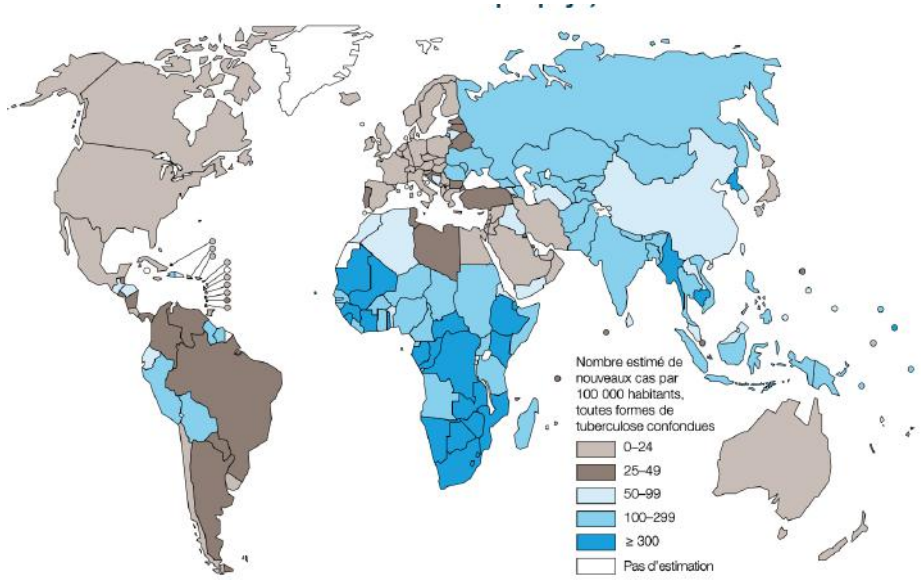


Figure 1.2: Tuberculosis World Wide
Source: http://www.stoptb.org/assets/documents/resources/publications/acsm/303100-worldtbday-fr_02b-email.pdf

1.1 Background

This part describes how some Deep Learning Methods, like Convolutional Neural Networks (CNNs) are used in image classification to solve Medical issues. However this part focus more on how attention can be useful to improve these methods.

Rajpurkar et al. (2018), is a 121-layer convolutional neural network trained on Ronald M. Summers (2017), currently the largest publicly available chest X-ray dataset. It contains over 100,000 frontal-view X-ray images with 14 diseases.

It is statically shown that the algorithm performs better compare to 4 radiologists in the average. In fact Rajpurkar et al. (2018) has a F1 score of 0.435 , higher than the radiologist average of 0.387.

While the training they did data augmentation and variate some hyper-parameters like learning rate (0.001 and decay of 10 after each epoch). The last fully-connected layer of the Dense convolutional network is replaced by a single output which they apply sigmoid to get the probabilities of having each of 14 diseases.

Noted that prediction of a disease base on X-ray images could need in some special case to have a lateral view and also information relevant to the medical antecedent of the patient. Since the frontal view was presented only to the radiologists, it would be very challenging to make a prediction based just on that for some special diseases.

For some kind of diseases, the information that allow you to make a decision is either distributed in the whole image or locate in specific parts on the image. Hence it will be helpful to focus on those informations while doing a prediction. That's what the attention mechanism aims to do.

In the paper Jetley et al. (2018) they prove that adding attention to Visual Geometry Group (VGG16) increase the accuracy of 7% on CIFAR-100. The architecture proposed consists to extract 3 chosen

features l_1, l_2, l_3 from different convolutional layer. Then they take a global feature G as a representation of the global image and correspond to the feature of the fully connected layer FC1-512 see Figure 1.3. Then they compute the compatibility score between each local feature l_i and the global feature G in order to know which of the feature give more attention. Finally they apply a sum weight of features and connected it to the dense output layer to make a prediction. Besides, they show that attention helps to identify and use the effective spatial support of visual information used by Convolutional Neural Networks (CNNs) to make classification decisions.

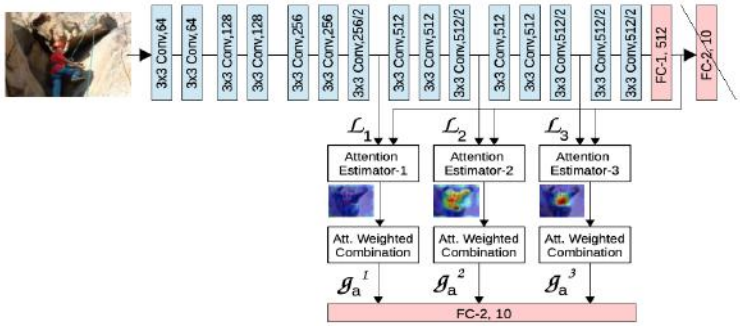


Figure 1.3: Tuberculosis World Wide
Source: https://arxiv.org/pdf/1804.02391.pdf

A similar architecture is underline in the paper Guan et al. (2018) where it is used an AG-CNN which has three branches, a global branch, a local branch, and a fusion branch mostly like the architecture proposed in the paper Jetley et al. (2018). In the fusion branch, it considers the part of images that contains lesions which are detected by the local branch and also the whole image which is represented by the global branch, to make decision. Hence it helps to avoid noise contained in the non-lesion area and also reduced the misalignment.

In some other algorithms like HongyuWang and Xia (2018), they used two branches. A classification branch for label prediction

which are a ResNet152 (152 learnable layers) and an attention branch for abnormality detection? The classification branch consists of a convolutional neural network follow by a max-pooling before to end with three (3) triple-layer residual blocks. Instead of using the Softmax layer of ResNet, they remove it and full-connected the last layer with 14-neurons representing (14 Thoracic Diseases) with sigmoid as the activation function for each of them. The output give a vector y_{cbp} (c.b.p as classification branch prediction).

In the attention branch, we have 6 convolutional layer. For the first 3 convolutional, each of them is followed by a Relu. Then, they employed the gradient-weighted class activation mapping (Grad-CAM) in the output of the third convolutional layer to estimate the class discriminative localization map for each class and normalize it, the whole is given as input to the 4th layer. They applied Relu in the 4th and 5th layers and sigmoid in the last. Which gives a vector named y_{abd} (y attention branch for abnormality detection). Finally, they combine the y_{cbp} and y_{abd} to do the prediction. They compare they are the model without attention to the one with attention and it clearly shows that the one with attention is performing better.

Despite the use of attention is improving the model performance however for some kind of diseases we

can be misled by noise or loss of information while doing the prediction. Indeed, for some diseases like Nodule which is present just in a part of the image, the global branch can be largely affected, if there is noise in the non-disease area. As the local branch can also lose some pieces of information by just focusing on a part of the image, for diseases like Pneumonia where its information is distributed in the whole image. Another way to think is while doing prediction, take the probability of a patient having a disease A, knowing its medical antecedent as a prior to adjusting the network prediction.

1.2 Objectives and goals of the project

The diagnosis and treatment of tuberculosis saved 58 million people between 2000 and 2018 according to the World Health Organization (WHO) Tuberculosis. We would like to contribute to this to save the lives of many children and adults too. This is how we implemented a method using Artificial Intelligence to provide a simple, effective, and accessible diagnostic method for everyone. So even people who do not have advanced technologies, or who have a shortage of experienced doctors, will be able to access a remote diagnostic platform. So we will save a lot of lives.

To achieve that result we set the following goals:

1. Explore the state of the art works including both academic literature and advanced industrial solutions using Artificial Intelligence for pulmonary diseases diagnosis.

2. Leverage open data and collect relevant X-ray images from tuberculosis (TB) versus healthy patients for Machine Learning perspectives.

3. Develop effective Deep Learning architectures and benchmark the algorithms for comparative analysis: Dense, Convolutional, Attention.

1.3 Project structure

Nowadays deep learning methods are widely used in many areas like image recognition, translation, image captioning, image classification, and more However, it contributes a lot in medical sciences and shows its improvement with image classification.

Our work is as described above to use Deep Learning Methods (Convolutional Neural Networks (CNNs), Attention, CNNs architectures) for Tuberculosis diagnosis.

After finishing the introduction, we are going to talk in Chapter 2 about Methodology and Terms. This chapter covers the basic understanding of Convolutional Neural Networks(CNNs) and its architectures and also an introduction of different types of Attention mechanism and the mathematics behind it. Then in Chapter 3, we are going to present our CNNs models and also talk about the dataset used to train them. Finally

Chapter 4 is mainly about the results that we got by using different training processes.

2. Methodology and terms

This chapter is about to present Convolutional Neural Networks and its architecture

This chapter covers a brief explanation of Convolutional Neural Networks (CNNs) and its architecture, an introduction to Attention Mechanism and enumerates a few types of Attention Mechanism.

2.1 Convolutional Neural Networks

CNNs take history in biological processes. In fact in 1968 , Hubel (1968) studies have shown that the visual cortex of animals contains complex arrangements of cells, responsible for detecting light in the sub-regions of the overlapping visual field, called receptive fields. They underline two different cells

One is called simple cells which respond to characteristic peaks like high contrast, big intensity and so one, inside the receptor field. The other called complex cells which have more ampler receptor fields and are locally invariant to the exact position of the motif. These two cells behave like a local filter for the input space.

Convolutional Neural Networks (CNNs) are a regularized version of fully connected networks Multilayer perceptron. We have a fully connected network if each neuron in the current layer is connected to all neurons in the previous one. The idea behind CNNs is to learn filters (features) from images dataset to classify received inputs.

CNNs, compared to other image classification algorithms, learn the filters which are hand engineered in traditional algorithms. Therefore this is a major advantage because it make them to be independent from prior knowledge and human effort in feature design.

Another advantage is the shared weight mechanism that we will describe later in the convolutional part. It makes the network faster and also induces translation invariance in CNNs.In general, CNNs are composed of a number of the convolutional layer in which part we apply filters to obtain features maps, a number of pooling layer or subsampling to avoid high dimensionality, refer to Figure 2.1. The pooling layer is most of the time a max pooling which consist to take the maximum number in the receptive field. However, an average pooling can be used which consist to take the average of numbers in the receptive field. Some other researchers define their own pooling layer. We can do one or more convolutional layers follow by a pooling layer and repeat it as long as we want depending on how would attack the problem. After that we make a dense fully connected layer or more (generally 1) before to connected with the final output as shown in Figure 2.1.

1. Convolutional Layer

   In the convolutional layer, we apply Filters/Kernels on the image or inputs. Those filters are represented as matrixes or tensors depending on the number channels or depth of the image (2 in the case of a greyScale image, 3 in the case of Red Green Blue (RGB) or colored image).

   Remember for CNNs we want to learn filters that activate when it detects some special feature.

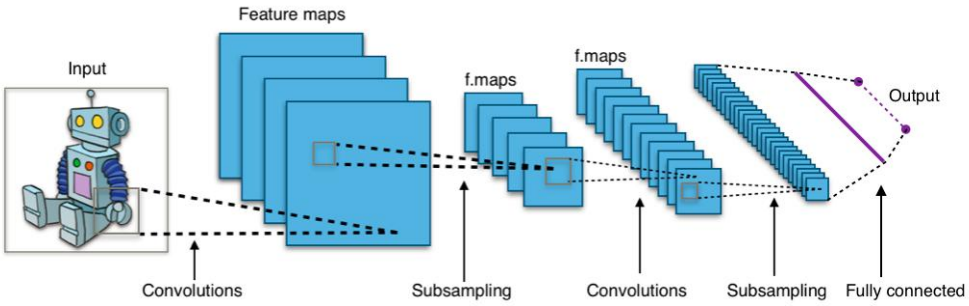


Figure 2.1: Typical CNN architecture
Source: https://en.wikipedia.org/wiki/File:Typical_cnn.png

In order to do have that, each filter is convolved across the heigh and width of the input volume. We compute the dot product of the entries of the input and this filter which gives a feature map or activation map. Theses activation maps obtained by applying many filters are stacking and represented the output volume of the layer.

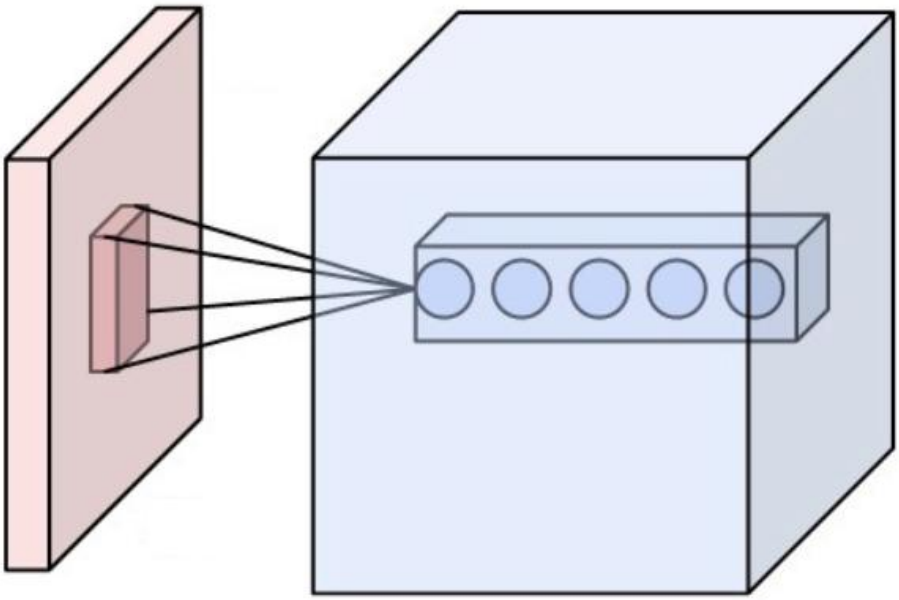


Figure 2.2: Neurons of a convolutional layer (blue), connected to their receptive field (red)
Source: https://en.wikipedia.org/wiki/File:Conv_layer.png

The surface where the filter is applied is called receptive fields describe in Figure 2.2 and the result of the convolve is called a neuron. the neurons obtained after sliced the filter in the input in one depth, share the same weight and bias. This is what we called the sharing weight or parameter sharing mechanism. Therefore it contributes to control the number of free parameters and to the translation invariance of the CNN architectures.

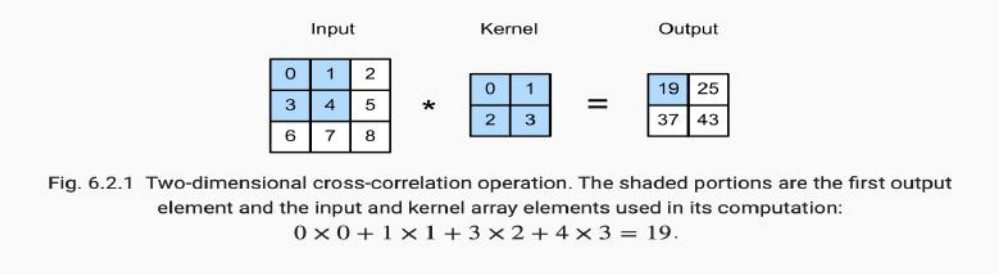


Fig. 6.2.1 Two-dimensional cross-correlation operation. The shaded portions are the first output element and the input and kernel array elements used in its computation:
$0 \times 0 + 1 \times 1 + 3 \times 2 + 4 \times 3 = 19.$

Figure 2.3: Filter/kernel convolving the input entries to give a feature map
Source: https://d2l.ai/chapter_convolutional-neural-networks/conv-layer.html

2. Pooling Layer

   Pooling layers help to rationalize the underlying computation. They combine the the output of neurons clusters at one layer into a single neuron is the next layer. Hence reduce the dimension of the input (data). There are local pooling which combines small clusters typically 2×2 and global pooling which acts on all the neurons in the convolutional neural network. A pooling layer can be a maximum pooling (max pooling) or average pooling.

   A maximum pooling take the maximum value in the cluster as the neuron for the next layer while average pooling, takes the average of neurons value in the cluster as the neuron for the next layer.

   Noted that it is not only this two methods of pooling that exist, it arrives that some people define their own method or function for pooling.

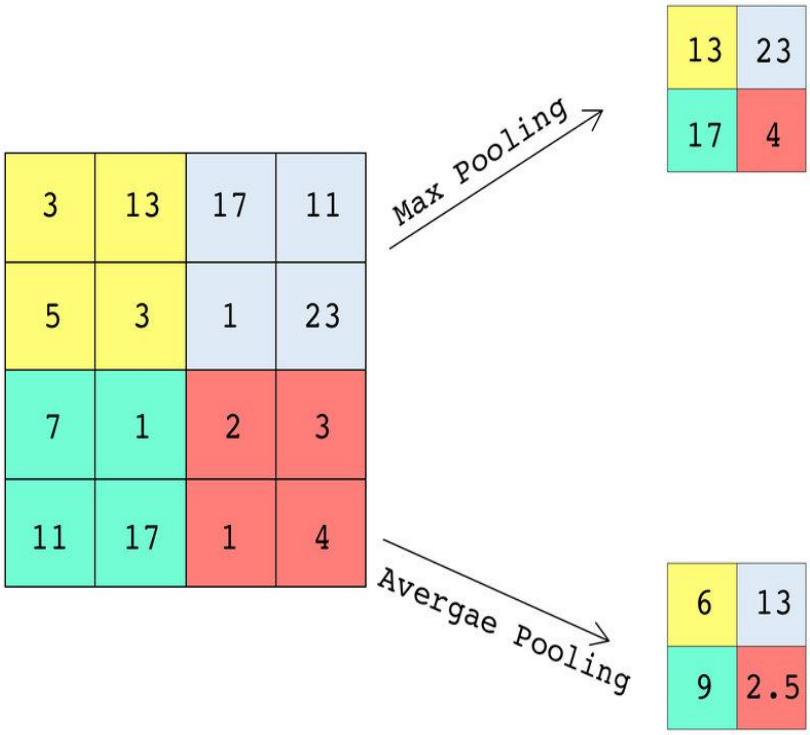


Figure 2.4: Max Pooling and average pooling
Source: https://www.researchgate.net/figure/Example-of-max-pooling-and-average-pooling-operations

3. Fully Connected layer

   In the fully connected layer we just flatter the final output into a single long vector of values , each representing a probability that a certain feature belongs to a label. We can add many dense fully connected as we want before to connect it with the final output layer in order to classify the image (Image Classification). This part is named as the classification part of CNNs architecture.

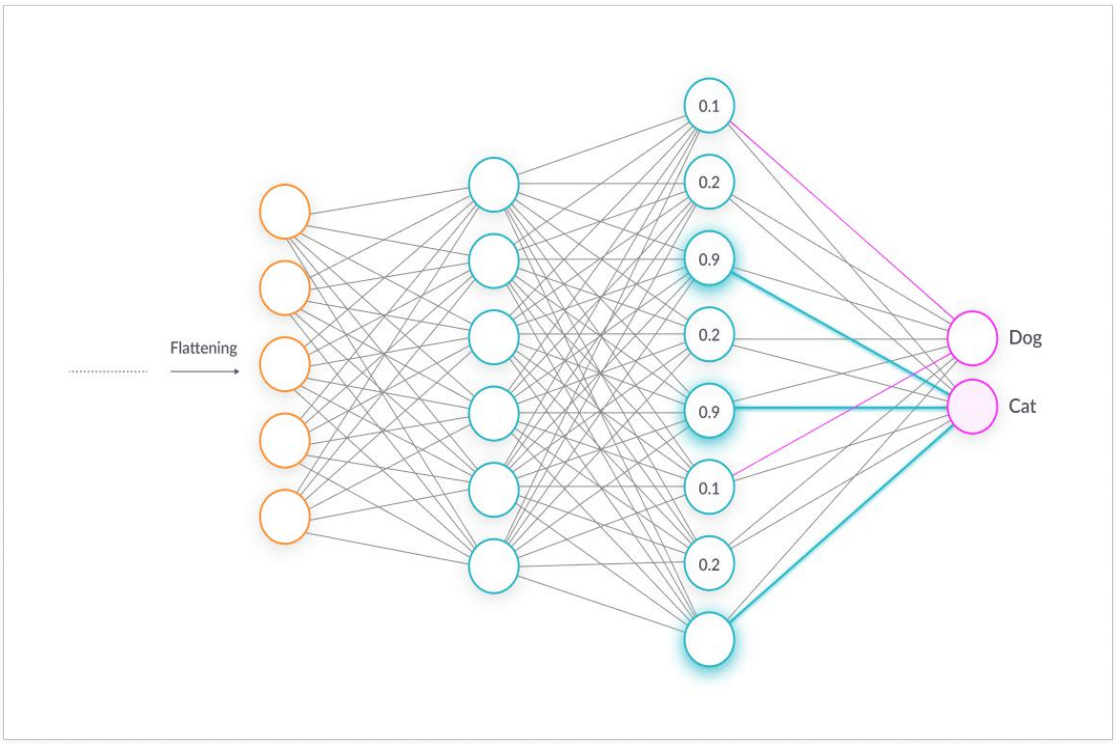


Figure 2.5: Fully Connected layer CNN
Source: https://missinglink.ai/guides/convolutional-neural-networks/fully-connected-layers-convolutional-neural-networks-complete-guide/

2.1.1 Convolutional neural Networks Architectures.

Convolutional neural networks are widely used in deep learning image classification, image recognition times series etc. What makes it so popular is some architectures that people created that show the beauty of CNNs we are going to see just two of them that we used in our project to solve the Tb prediction problem.

- Visual Geometry Group 16 (VGG16)

  VGG16 is a very deep convolutional neural network proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper Simonyan and Zisserman (2015). In ImageNet which is a dataset containing more than 14 million images belonging in 1000 classes, the VGG achieves a test accuracy of 92.7% which makes it be part of the top 5 test accuracy. It makes improvements over other architectures by replacing large kernelsize filters. Note that it was training for a couple of weeks (3 to 4 weeks) in a NVIDIA Titan Black GPU's.

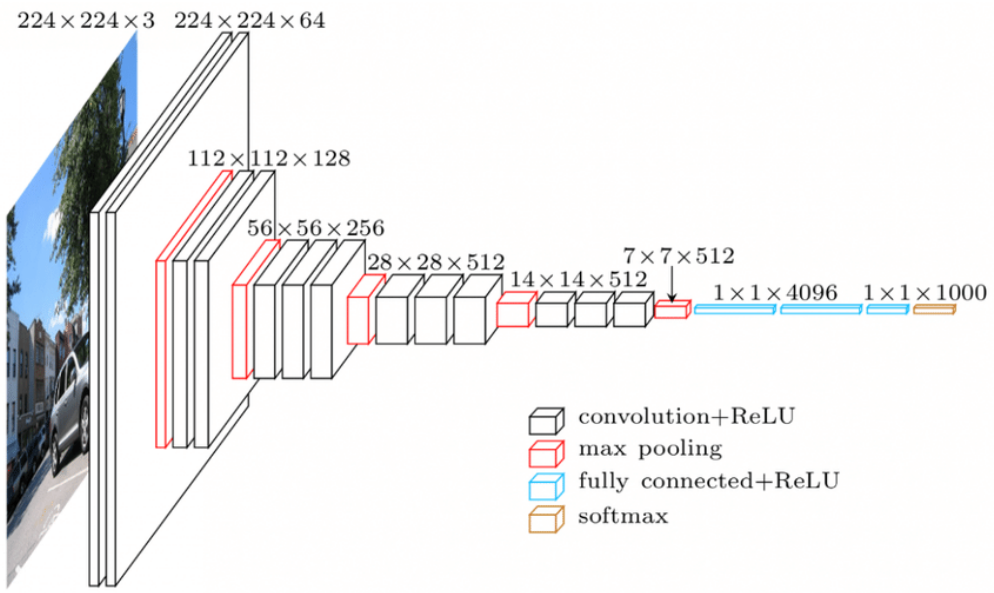


Figure 2.6: VGG 16 architecture
Source: https://www.researchgate.net/figure/VGG-16-neural-network-architecture_fig1_327070011

During the training the VGG16 takes as input an image of 224 × 224 with a depth of 3 Figure 2.6. The image is followed by 2 convolutional layers with a filters of size 3 × 3 of stride 1 (used 64 of them) followed a max-pooling of size 2 × 2 of stride 2. The same scenario is repeated again 2 convolutional layers (same size filters but 128 of them) followed by a max-pooling. Then, comes another 3 convolutional layer (with 256 filters) followed by a max-pooling. Again repeated other 2 , 3 convolutional layer followed by a max-pooling, both have the same number of filter 512. Then comes 3 fully connected layers with 4096, 4096, and 1000 neurons respectively. The last one representing the output layer of 1000 classes (with activation function Softmax) Figure 2.7.

Noted the for each convolutional layer and pooling layer Relu is used as activation function for non-linearity. VGG16 has over 138 million parameters.

Figure 2.7: Fully Connected layer CNN
Source: https://qph.fs.quoracdn.net/main-qimg-e657c195fc2696c7d5fc0b1e3682fde6

2.2 Attention

Attention mechanism, is, for a neural network to focus more on some parts of an image rather than others. This is possible by doing the principle of Gap off and on which consists to activate some pixels of the image and deactivate others. This gives as result an image with some parts blur Figure 2.9 (the area where the network will give less attention) and other crop Figure 2.8 or without blur effects Figure 2.9 (areas, where the network will focus more, will be considered to make a prediction or classification. This attention mechanism is quite good and the case that the image contains some noises or when in some image classification we are looking for some pieces of information which are regrouped just in some part of the image. While some are saying that attention was born for translation Weng (2018) because of its capacity to memorize long sequences which is so important for when we want to translate long sentences. Others also used in image recognition (Image captioning, image classification) and obtain good results in terms of enhancement of accuracyJetley et al. (2018); Górriz et al. (2019).

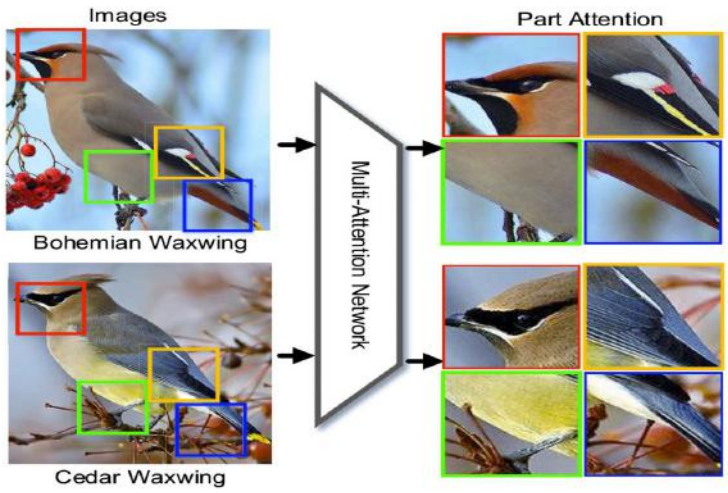


Figure 2.8: Multi attention mechanism
Source: https://www.microsoft.com/enus/research/publication/learningmultiattentionconvolutional-neuralnetworkfinegrainedimagerecognition/

Figure 2.9: Soft attention mechanism
Source: https://machinelearningmastery.com/howtocaption-photoswithdeeplearning/

There exist many types of attention mechanism. Mostly known is soft attention and hard attention. In the first while we are giving more focus in some parts of the image or sequence we are also considering other parts but with less focus. While in hard attention as its name says we just focus on some relevant parts where the information seems to be and hide totally other parts of the image. They are also other type of attention approaches but we are going to see the maths behind attention and also see some types of attention.

2.2.1 The maths behind attention mechanism.

The attention mechanism aims to compute the alignment score or compatibility between two sources a and b. In order to know how the source a is related to the source b. To have a clear idea of what it does mean let translate it into mathematics language.

Let us denote the set of features map on a given convolutional layer l

$$F^l = \left\{ f_1^l, f_2^l, ..., f_n^l \right\}, \qquad l \in \{1, 2, ...m\}. \tag{2.2.1}$$

Set G representing the global image
Remember we are interesting on how a feature f_i^l,belonging to a layer l and the spatial i is related to the global image G.
For that we need to compute the compatibility score between this feature map and the global image G. Denote c_i^l the compatibility score to the feature map f_i^l to G.

$$c_i^l = \langle f_i^l, G \rangle, \tag{2.2.2}$$

which is the product between the feature and the global image. Set

$$C^l = \left\{ c_1^l, c_2^l, ..., c_n^l \right\}, \tag{2.2.3}$$

the set of all compatibility scores of a given layer l (of its feature maps) or denote also as $C(\hat{F}^l, g)$ where $\hat{F}^s$ is the image of F^l under a linear mapping of the f_i^l to the dimensionality of G
We are going to normalized the set of compatibility score by applying Softmax on it

$$A^l = \left\{ a_1^l, a_2^l, ..., a_n^l \right\}, \tag{2.2.4}$$

with

$$a_i^l = \frac{\exp(c_i^l)}{\sum_{j=1}^n \exp(c_j^l)}, \qquad i \in \{1, 2, ...n\}. \tag{2.2.5}$$

Finally we are going to compute

$$G_a^l = \sum_{i=1}^n a_i^l . f_i^l. \tag{2.2.6}$$

G_a^l replaces G as a global image descriptor. And the case where we are just based on it to do a prediction

The Attention is representing by A which contains the set probability scores of each layer.

In the case where $m = 1$ (just one layer), the attention-incorporating global vector G_a is mapped onto a Tdimensional vector which is passed to a Softmax function to obtain the class probabilities $\left\{p_1, p_2, ..., p_t\right\}$ (t is the number of classes).

In the case where we do not have a single layer (m $>$ 1) We can proceed in two different ways :
Either we Concatenate the global vectors into a single vector $G_a = \left[G_a^1, G_a^2, ..., G_a^m\right]$. Then take G_a as input for the linear classification describe above. Or using m different linear classifiers and averaging the output class probabilities. Which means
We will take each G_a^i, passed it through a Softmax to obtain the following output class probabilities: $\left\{p_1^i, p_2^i, ..., p_t^i\right\}$. And then take the average of this probabilities to obtain the final class probabilities $P = \left\{p_1, p_2, ..., p_t\right\}$ with $p_j = \dfrac{\sum_{i=1}^{m} p_j^i}{m}$

2.2.2 Types of attention mechanism.

In this section we are going to see briefly some type of attention mechanism.

- Self Attention

  An attention mechanism is called Self-attention or intra-attention when it when its compute a representation of a single sequence q by relating different positions of q. This mechanism is used a lot in machine reading, abstractive summarization, or image description generation.

  In the paper Cheng and Dong (2018), they used self-attention for machine-reading Figure 2.10 describes their Self-attention mechanism that shows how a current word of the sequence is related or correlated to the previous ones.

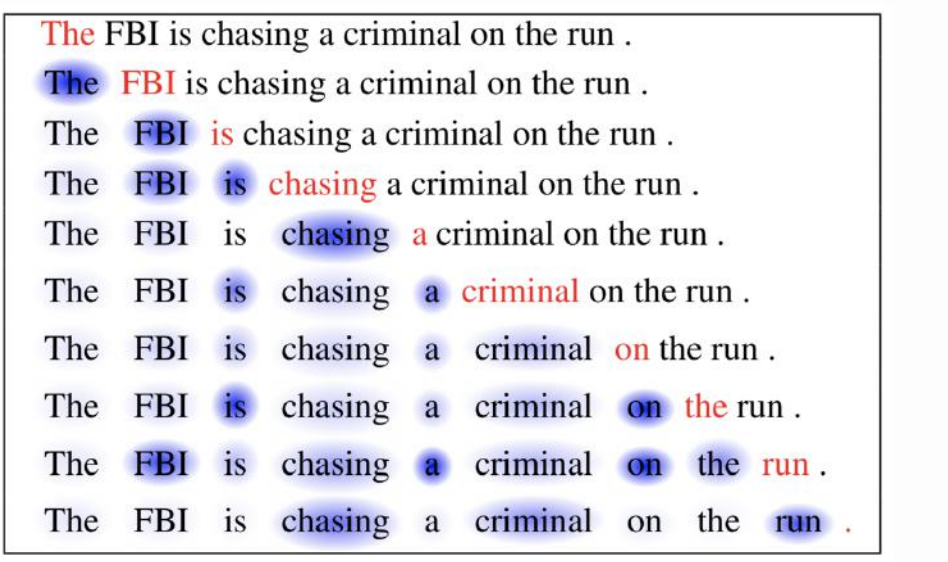


Figure 2.10: Self-attention mechanism
Source: https://arxiv.org/pdf/1601.06733.pdf

- Soft and Hard attention

  As we describe above the soft attention take information in the whole image while hard attention

does not. In fact, let $f_1, f_2, ..., f_n$ be the features from a convolutional layer. For soft attention, we compute the score s_i of each f_i to see how much attention do we have to give it.

$s_i = \tanh(W_c \times C + W_x \times f_i)$ with C the context and $W_c, W - x$ the weight of the context C, the feature f_i respectively.

Once we have the scores we normalized them by passing them through a Softmax activation function.

$$\alpha_i = \frac{\exp(s_i)}{\sum_{j=1}^{n} \exp(s_j)}$$

Then after that we take the average weight as Z of the features as output of the attention representing the new image with some parts becoming dark (low score value) and another stay the same or overlay (high value of score) see Figure 2.11.

$$Z = \sum_{i}^{n} \alpha_i \times f_i$$

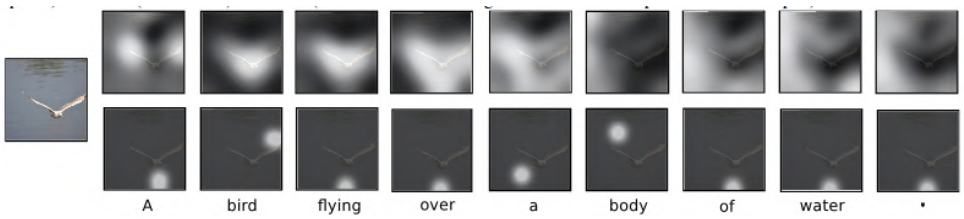


Figure 2.11: Soft vs Hard attention
Source: https://arxiv.org/pdf/1502.03044.pdf

For the hard attention instead of taking the sum of weighted of all features, it just takes one feature randomly. As result the network just focus in one part of the image see Figure 2.11. Hence the Z in hard attention is equivalent to

$$Z \equiv \alpha_i \times f_i$$

with f_i the feature choose randomly. That why we say that soft attention is deterministic and differentiable while hard attention is stochastic and non differentiable hence to update its weights we called complex methods most of them taken from Reenforcement learning methods.

3. Models formulation

This chapter covers different Convolutional Neural Networks models created and the implementation of attention mechanism in one of them.

3.1 Datasets

During our research project we use two datasets Chest-ray (images) provided by S et al. (Accessed April 2020b). The datasets are available to the research community and both sets contain normal as well as abnormal x-rays, with the latter containing manifestations of tuberculosis. Below are the complete information about the two datasets.

- The Montgomery County X-ray Set is a small dataset containing 138 posterior-anterior x-ray images. It is constituted of 58 x-rays of abnormal person with manifestations of tuberculosis and 80 x-rays normal. All images are deidentified and available in DICOM format. The set covers a wide range of abnormalities, including effusions and miliary patterns. The data set includes radiology readings available as text file.

- The Shenzhen Hospital X-ray Set / China data set, is a dataset presenting 340 normal x-ray images and 275 abnormal x-rays showing various manifestations of tuberculosis. Collected by Shenzhen No.3 Hospital in Shenzhen in China and images are in a JPEG format.

3.2 Convolutional Neural Networks models

We created different model which are in the number of 6: 3 Convolutional neural network without attention mechanism, VGG16 with attention and without attention and Finally we implement attention mechanism in one of our CNN models pre-trainned. We are going to see how the models performs by using different training process.

3.2.1 Presentation of our different model.

- DSG20a

  This CNN model with over 34 million parameters. The architecture is built as follows 3 convolutional layers followed by a max-pooling and a dropout (to avoid overfitting during the training) and this repeated 3 times before to flatten. We set up two activations function depending on the shape of the input that we will pass the model and also see which activation function will give more non-linearity during the training. Here is Figure 3.1 . Either we connect the output with Softmax activation function which will be applied in 2 neurons in this case, each of them presenting the probability of one class : TB (The class of people who have the tuberculosis disease) or Non TB (normal people, do not have Tuberculosis). Or we use Sigmoid but in this case, it will be connected with 1 neuron, and as output is a number between 0 and 1. Then we consider if the number is greater than 0.5 means that the person has TB and if not the person do not have TB. We will see in the result more clearly about the use of the two activation functions.

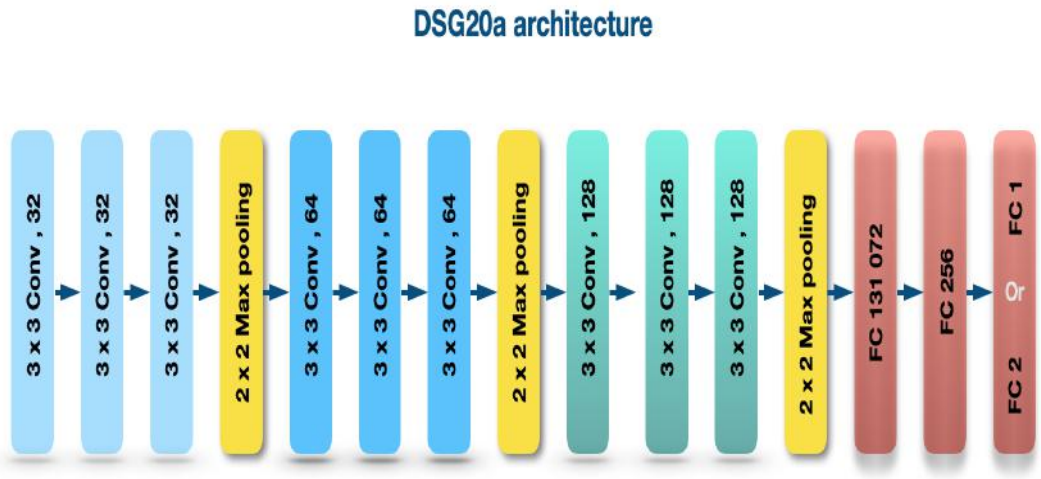


Figure 3.1: DSG20a model architecture

- DSG20b

  This model has a number of parameters for around 6 million. It has an architecture similar to the previous one just here instead of repeating sequence 3 convolutional layers, followed by max-pooling, followed by a drop-out 3 times we just repeat it 2 times. And use sigmoid as activation function as described in Figure 3.2.

Figure 3.2: DSG20b model architecture

- DSG20

  The DSG20c CNN model is the one that has less parameter but with a number of convolutional layers and max-pooling more important as to show the Figure 3.3. The reason why it has fewer parameters is the fact of its number of max-pooling and also the filters we didn't use a big number of filters and some times we use a filter of dimension 2×2

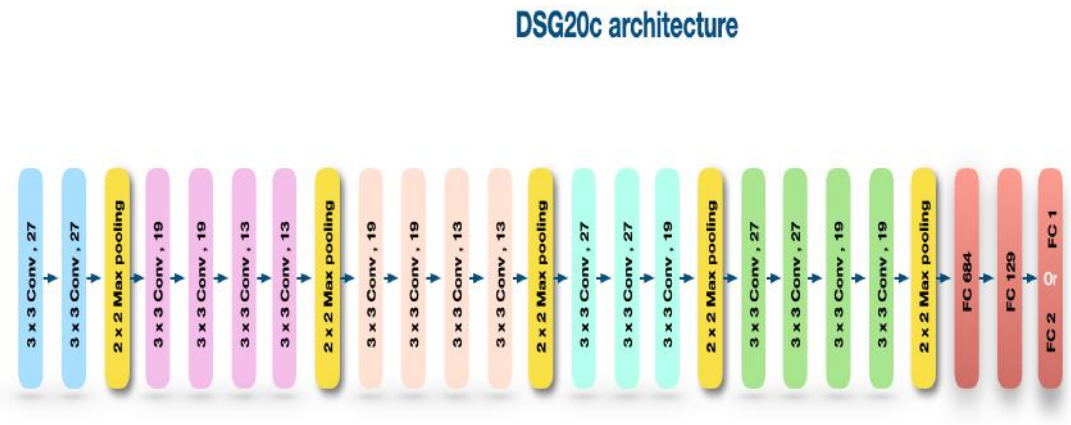


Figure 3.3: DSG20c model architecture

3.3 Implement of Attention in out pre trained DSG20b model

In this part we implement attention in one of our CNN models (DSG20b). The attention acts like describe the section 2.2. The goal is to push the network focusing in relevant part of the image which suppose to contain the information that we are looking for. We will first describe the attention mechanism that we setup and then after see the attention performances during the training on the S et al. (Accessed April 2020a). I took inspiration in Mader (Accessed May 2020) codes, an implementation of attention mechanism with VGG16.

3.3.1 Attention mechanism implementation.

DSG20b pre-trained features are taken and normalized. After that, an attention tensor with kernel size 1×1 was created (to act in the pixels) for all its convolutional layer. Then the attention tensor is multiply to the normalized pre trained features and give as result the mask for the features. To account for missing values from the attention model, the attention tensor and the mask are passed as input to a function rescale Global Average Pooling (GAP) which apply a GAP in both inputs and return a tensor T as result of the mask rescaled by the attention tensor see Figure 3.4. Finally the tensor T is fully connected with 128 neurons followed by 2 neurons (for the classification) with Softmax as activation function.

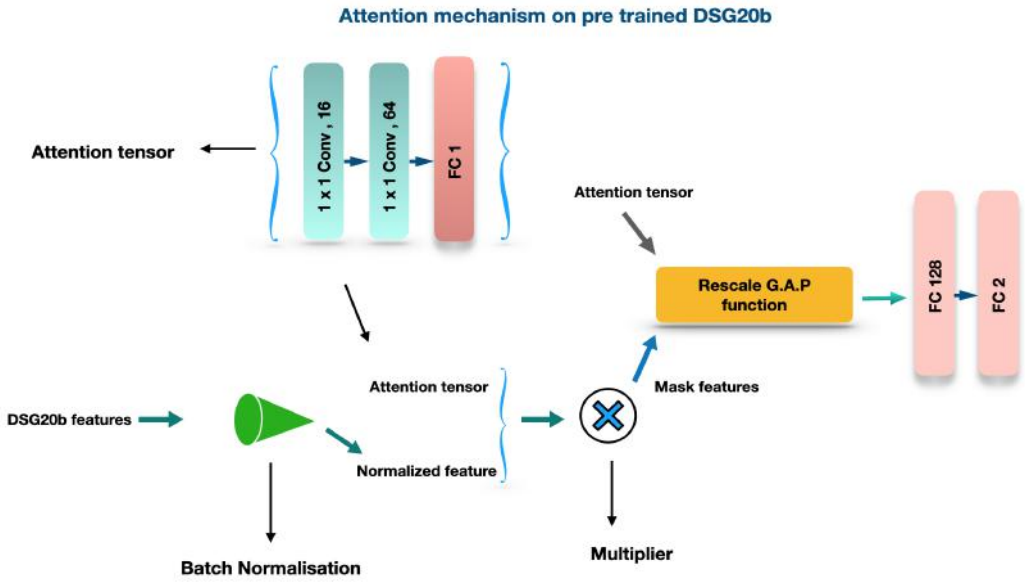


Figure 3.4: DSG20b with attention mechanism

3.4 DSG20u (united)

In this section we will introduce a concept which is to combine the top best models while doing a prediction. For that we will define some concepts like probability score ,score error and TB certainty threshold.

3.4.1 Terms definition.

- Probability score
  Since almost all models are doing good in terms of training accuracy and testing accuracy however it is not enough to judge which model is better than the other one. Hence we define the probability score which will give us a clear look at it.

  Let N_i be the number of images belonging to the label i (TB or Non TB). Let also denote P_k^i the predict probability for an image k to be classify to the label i (TB or Non TB). We define the probability score S^i as follows :

  $$S^i = \frac{\sum_{k=1}^{N_i} P_k^i}{N_i}$$

  Since

  $$P_k^i \in [0,1] \quad \Longrightarrow \quad S^i \in [0,1]$$

  The probability score helps us to know how well the model predicted all N images for both classes. A model is doing perfectly in the whole datasets if its probability score is equal to 1 for both classes.

- Score error

Since the model can not predict perfectly certainty all images, it is doing some errors while the prediction. We define the score error from the probability score directly. Let E^i be the score error then

$$E^i = 1 - S^i, \quad E^i \in [0, 1]$$

The score helps us to know the general error that is doing while predicting images. It can also be calculated as follows

$$E^i = \frac{\sum_{k=1}^{N_i}(1 - P_k^i)}{N_i}$$

- TB certainty threshold
  When we are doing prediction or selecting the model we are interesting on which model is doing less false negative and more false positive. For instance, it is better than the model predicts for a person who does not have TB as someone who has TB than predicted a person who does have TB as someone who does not. This last can be a serious matter. So we want to avoid as much as possible the second case (false negative). So while we are doing a prediction especially for Non TB we want to " be sure or certain" of our decision. Hence we define the certainty threshold to help us to adjust predictions and then minimize as much as a possible false negative. The certainty values is between [0,1], TB certainty threshold is set to be equal to 0.3. If the certainty value is less than the TB certainty threshold then we are not going to predict the image as Non TB and If the certainty is greater than TB certainty threshold, then we are "sure" to predict the image as a Non TB image. The following section describes how the algorithms works.

3.4.2 Algorithm description.

Let DSG20a , DSG20b , DSG20c and DSG20b-attention our selected models (M = 4) . We are going to compute the probability score of each of them. Set $P_m^i, m \in 1, M$ the probability given by the model m for the label i , S_m^i and E_m^i the probability score of the model m for the label i with its score error respectively.We compute the final probability of the label i denoted P^i as follows

$$P^i = \frac{\sum_{m=1}^{M} S_m^i \times P_m^i - E_m^i}{M}.$$

Algorithm: The function that make the decision will take as input a list containing the prediction (as probabilities) of each model for a given label and their probability scores in the same order.

$$models\_preds = \left\{(P_1^{ntb}, P_1^{tb}), ..., (P_m^{ntb}, P_m^{tb})\right\},$$
$$model\_scores = \left\{(S_1^{ntb}, S_1^{tb}), ..., (S_m^{ntb}, S_m^{tb})\right\}.$$

Algorithm 1: DSG20u algorithm for making prediction

```
                                                    // Compute the final probability for TB
1  P^tb = (∑_{m=1}^{M} S_m^tb × P_m^tb − E_m^tb) / M
                                                    // Compute the final probability for Non TB
2  P^ntb = (∑_{m=1}^{M} S_m^ntb × P_m^ntb − E_m^ntb) / M
                                                    // Compute the certainty
3  certainty = abs(P^ntb − P^tb)
                                                    // Compare the final probabilities and make a decision
4  if P^ntb > P^tb then
       // We want to be sure that the person does not have TB
5      if certainty > TB_CERTAINTY_THRESHOLD then
           // Here we are "certain" that the person does not have TB
6          return 0
7      else
           // We are not "certain" that the person does not have TB
8          return 1
9  else
10     return 1
```

4. Results

Different training process have been shown different results. This chapter covers a discussion about the results obtained.

4.1 Training process

4.1.1 Train on the Montgomery Dataset.

All models are trained in the S et al. (Accessed April 2020c) dataset. Since the dataset is very small (just 138 images). It becomes necessary to do a data augmentation which multiply by 10 the dataset size. Below are results obtained.

After raining this models by 500 epochs and a steps per epoch of 3 except DSG20 with attention which was trained with 315 epochs, we realized they were doing quite good as describe in Figure 4.1. As you can, the models DSG20c , VGG , VGG with attention have a training accuracy of 1.O0 but a testing accuracy of 0.80 , 0.70 , 0.80 respectively. While the DSG20a and DSG20b have a training accuracy of 0.80 but a testing accuracy which is more important 0.90 and 1.00 respectively. Almost the same is for DSG20 with attention which has a training accuracy low 0.70 but a high testing accuracy of 0.90. The low value in the training could be improves by increasing its number of epochs. Some models during the training can overfit (high accuracy in the earlier epochs , validation accuracy not following the training accuracy) and miss some predictions during the testing like the VGG model.

Models trained in the Montgomery dataset with augmentation		
Models	Train Accuracy	Test Accuracy
DSG20a	0.80	0.90
DSG20b	0.80	1.00
DSG20c	1.00	0.80
DSG20 with att.	0.70	0.90
VGG	1.00	0.70
VGG with att.	1.00	0.80

Figure 4.1: Train models on the Montgomery Dataset

4.1.2 Train on the China Dataset.

All models are trained using the this dataset S et al. (Accessed April 2020a) and we use a different approach this time. Instead of doing just data augmentation, models are trained with data augmentation and without it. The Figure 4.2 highlight the results obtained.
We can see that most of the models are doing better without data augmentation than with data augmentation in both the training and testing accuracy except DSG20a. The goal of this process was to see If it was necessary to do data augmentation or not. Since we have a data set which is balanced (326 TB , 336 Non TB) and it was not too small. So we decide to make comparison in order to have a detail look on it. We can also see that DSG20 with attention is doing good in

training accuracy and testing accuracy.

Models	Train without data augmentation		Train with data augmentation	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
DSG20a	0.70	0.80	0.81	0.71
DSG20b	1.00	0.90	0.84	0.93
DSG20c	1.00	0.90	0.87	0.93
DSG20 with att.	1.00	0.90	—	—
VGG	1.00	0.90	0.968	0.75
VGG with att.	1.00	0.90	0.93	0.90

Figure 4.2: Train models on the China Dataset

4.1.3 Train on the China Dataset and use the Montgomery dataset as testing data.

This time models are trained using the S et al. (Accessed April 2020a) and tested in the S et al. (Accessed April 2020c) dataset. Below is a table summarizing the results obtained.
Here we can notice that except the DSG20a which has training accuracy and testing accuracy 1.00 and 0.40 respectively, all the others models are doing pretty good in during the training and testing in general mostly without data augmentation, see Figure 4.3. This should make think that the models are good since they were training in one dataset and test in a completely different one.

Models trained in China Dataset and test in Montgomery dataset				
Models	Train without data aug		Train with data aug	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
DSG20a	1.00	0.40	1.00	0.40
DSG20b	1.00	0.90	1.00	0.90
DSG20c	1.00	1.00	0.80	1.00
DSG20 with att.	0.90	1.00	0.80	1.00
VGG	1.00	1.00	0.90	0.80
VGG with att.	1.00	1.00	0.80	0.94

Figure 4.3: Train models on the China Dataset and use Montgomery dataset as testing data

4.1.4 Train on the combine dataset.

In this part we combine both dataset and train all models with. In one hand we use data augmentation in the other we did not. See Figure:4.4 which summarize the results obtained for all the models.
As we can notice, models are doing good both with data augmentation and without it, except the DSG20b which had a training accuracy very poor (0.5) during the training with data augmentation.

All the figures remains in global the same they show that models trained without data augmentation are better than with data augmentation in general. And also that the models are good since they show their performance when, trained in one dataset and tested to a completely different one. The implementation of attention in the models shows also good result. Since almost all models are doing good, The use of attention could not be seen truly here. When it is arrived to

Models trained in combined datasets				
Models	Train without data aug		Train with data aug	
	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
DSG20a	1.00	0.90	0.90	1.00
DSG20b	1.00	1.00	0.50	1.00
DSG20c	0.90	1.00	1.00	0.90
DSG20 with att.	1.00	0.90	——	—-
VGG	1.00	0.90	0.97	0.75
VGG with att.	0.90	1.00	0.80	1.00

Figure 4.4: Train models on both datasets

make a choice between models, it can be really difficult. Hence we got the idea to combine the models while doing a prediction this is the aim of DSG20u describes in section

4.1.5 Comparison between DSG20u and other CNN models.

This part describes how the DSG20u algorithm improves predictions. In fact we selected the top 4 best models,DSG20b,DSG20c,VGG and DSG20 with attention that have the highest probability score in both TB and Non TB see Figure 4.5. Then We take a sample test of size 161 randomly in the combine dataset (with 79 TB images and 82 Non TB) and passed it to each models. The Figure 4.5 shows the results.

We can see that DSG20u has the highest percent in the prediction of TB and Non TB making it to have the highest accuracy of 0.985. We can deduced from that, the algorithm which is as result of combine models could help to enhance the precision while the prediction.

DSG20u compare to other CNNs models			
Models	Non TB well predicted (%)	TB well predicted (%)	Accuracy
DSG20u	0.97	1.00	0.98
DSG20b	0.90	0.93	0.92
DSG20c	0.94	0.87	0.90
VGG	0.96	0.77	0.86
DSG20 with att.	0.90	0.92	0.91

Figure 4.5: Comparison between DSG20u and other CNNs models

Noted that in general where the DSG20u algorithm fails during the prediction is where all the models fails together which means most the models are good most the algorithms performs better. That what explains the results discussed above.

5. Conclusion

Deep Learning Methods have been used a lot for Tuberculosis diagnosis specially Convolutional Neural Networks (CNNs). This later combine with Attention Mechanism improves the the model prediction.

The study has been showed good results in the prediction of TB from chest X-ray images, for almost all models that have been exposed. Most of the CNNs models reached an accuracy above 95% in the training and above 90% in the testing. The new proposed model which is an ensemble method that makes a weighted combination of the best performing models with a way to choose the weights to improve prediction power by reducing false negatives, that is increase recall.

A future work will be to test our models and algorithms in a different dataset to see how they will perform or figure out how to use prior information like medical antecedent, while making Tuberculosis prediction.

Acknowledgements

In the name of Allah, the Most Gracious and the Most Merciful.

Alhamdulillah, all praise to almighty Allah for the strengths and His blessing in completing this thesis.

I would like to thank AIMS and it's funders for supporting this thesis which has been kept on track and been seen through to completion with the monitoring, support, and encouragement of numerous people including my Supervisor Dr. Bubacarr Bah from University of Stellenbosh and AIMS South Africa, my Co-Supervisor Dr. Habiboulaye AMADOU BOUBACAR from Air Liquide, Paris, France, who gave me the golden opportunity to do this wonderful project on the topic Deep Learning Methods for TB Diagnosis, which also helped me in doing a lot of Research and I discovered many things I am really thankful to them. My tutor Reem ELMAHDI.

I would like to acknowledge Jan Groenewald for his IT support and helps to solve errors that occurred along this project.

I would like to thanks also Dr. Simukai UTETE for all its advices, support and referee that she wrote for me.

I would especially like to thank Professor Barry Green, and all the AIMS community and staff.

I pay my gratitude to Noluvuyo Hobana for giving us extra courses in the English Beginner Class that we still miss. She helps us to improve our level and also provides necessary resource and remarks along this research work.

Nobody has been more important to me in the pursuit of this project than the members of my family. I would like to thank my parents Cheikh Ahmed Tidiane Ngom and Indé Seck, whose love and guidance are with me in whatever I pursue. They are the ultimate role models. Most importantly, I wish to thank my best friend Ansoumane Talla for his sincerity and loyalty that he always shows. He is more than a brother for me.

I would like to thank also all the persons that did pray for me during this time and give me their support.

References

Cheng, J. and Dong, L. Long short-term memory-networks for machine reading. *https://arxiv.org/pdf/1601.06733.pdf*, 2018. URL https://arxiv.org/pdf/1601.06733.pdf.

Guan, Q., Huang, Y., Zhong, Z., Zheng, Z., Zheng, L., and Yang, Y. Diagnose like a radiologist: Attention guided convolutional neural network for thorax disease classification. *https://arxiv.org/abs/1801.09927*, pages 1–8, 2018.

Górriz, M., Antony, J., McGuinness, K., i Nieto, X. G., and O'Connor, N. Assessing knee oa severity with cnn attention-based end-to-end architectures. *https://www.researchgate.net/publication/332380878*, pages 1–8, 2019.

Górriz, M., Antony, J., McGuinness, K., i Nieto, X. G., and O'Connor, N. Github repo of attention with cnn. Github repo of attention with CNN, https://github.com/marc-gorriz/KneeOA-CNNAttention, Accessed May 2020.

HongyuWang and Xia, Y. Chestnet: A deep neural network for classification of thoracic diseases on chest radiography. *https://arxiv.org/pdf/1807.03058.pdf*, pages 1–8, 2018.

Hubel, editor. *Receptive fields and functional architecture of monkey striate cortex*. The Journal of Physiology, 1968.

Jetley, S., Lord, N. A., Lee, N., and Torr, P. H. S. Learn to pay attention. *https://arxiv.org/pdf/1804.02391.pdf*, pages 1–8, 2018.

Mader, K. S. Pretrained-vgg16 w/attention for tuberculosis. Kaggle, https://www.kaggle.com/kmader/pretrained-vgg16-w-attention-for-tuberculosis, Accessed May 2020.

Maladie Infectieuse. Infectious disease. Wikipedia, https://fr.wikipedia.org/wiki/Maladie_infectieuse, Accessed May 2020.

Multilayer perceptron. Multilayer perceptron. Wikipedia, https://en.wikipedia.org/wiki/Multilayer_perceptron, Accessed May 2020.

Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Ball, R. L., Langlotz, C., Shpanskaya, K., Lungren, M. P., and Ng, A. Y. Chexnet:radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv.org*, pages 1–8, 2018.

Ronald M. Summers, N. I. o. H. N. Nih clinical center provides one of the largest publicly available chest x-ray datasets to scientific community. *https://www.nih.gov/news-events/news-releases/nih-clinical-center-provides-one-largest-publicly-available-chest-x-ray-datasets-scientific-community*, pages 1–8, 2017.

S, J., S, C., SK, A., Y, W., P, L., and GR, T. Two public chest xray datasets for computeraided screening of pulmonary diseases. link to Download the dataset, https://openi.nlm.nih.gov/imgs/collections/ChinaSet_AllFiles.zip, Accessed April 2020a.

S, J., S, C., SK, A., Y, W., P, L., and GR, T. Two public chest xray datasets for computeraided screening of pulmonary diseases. Webots, https://lhncbc.nlm.nih.gov/publication/pub9356, Accessed April 2020b.

S, J., S, C., SK, A., Y, W., P, L., and GR, T. Two public chest xray datasets for computeraided screening of pulmonary diseases. Download link, https://openi.nlm.nih.gov/imgs/collections/NLM-MontgomeryCXRSet.zip, Accessed April 2020c.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for largescale image recognition. *arXiv.org*, 27:1–8, 2015.

South Africa and India. Pour finir avec la tuberculose. Rapport de plaidoyer, http://www.stoptb.org/assets/documents/resources/publications/acsm/303100-worldtbday-fr_02b-email.pdf, Accessed May 2020.

Tuberculosis. Tuberculosis. World Health Organization, https://fr.wikipedia.org/wiki/Maladie_infectieuse, Accessed May 2020.

Tuberculosis spread by countries. Tuberculosis spread in south africa. World Health Organization, https://worldhealthorg.shinyapps.io/tb_profiles/?_inputs_&lan=%22EN%22&iso2=%22ZA%22&main_tabs=%22est_tab%22, Accessed May 2020.

Weng, L. Attention? attention! *lilianweng.github.io/lil-log*, 2018. URL http://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html.