UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY TECH

Escuela de Ciencias Biológicas e Ingeniería

TÍTULO: Breast Cancer Identification Software With CNN

Trabajo de integración curricular presentado como requisito para la obtención del título de

Ingeniera Biomédica

Autor:

Pereira Carrillo Jackeline Pamela

Tutor

PhD. Almeida Galárraga Diego Alfonzo

Co-tutor

PhD. Chang Tortolero Oscar

Urcuquí, Febrero 2021

Urcuquí, 28 de mayo de 2021

SECRETARÍA GENERAL
(Vicerrectorado Académico/Cancillería)
ESCUELA DE CIENCIAS BIOLÓGICAS E INGENIERÍA
CARRERA DE BIOMEDICINA
ACTA DE DEFENSA No. UITEY-BIO-2021-00006-AD

A los 28 días del mes de mayo de 2021, a las 10:40 horas, de manera virtual mediante videoconferencia, y ante el Tribunal Calificador, integrado por los docentes:

Presidente Tribunal de Defensa	Dra. SALUM , GRACIELA MARISA , Ph.D.
Miembro No Tutor	Dr. ALEXIS FRANK , Ph.D.
Tutor	Dr. ALMEIDA GALARRAGA, DIEGO ALFONSO , Ph.D.

El(la) señor(ita) estudiante **PEREIRA CARRILLO, JACKELINE PAMELA**, con cédula de identidad No. **1722723786**, de la **ESCUELA DE CIENCIAS BIOLÓGICAS E INGENIERÍA**, de la Carrera de **BIOMEDICINA**, aprobada por el Consejo de Educación Superior (CES), mediante Resolución **RPC-SO-43-No.496-2014**, realiza a través de videoconferencia, la sustentación de su trabajo de titulación denominado: **BREAST CANCER IDENTIFICATION SOFWARE WITH CNN** , previa a la obtención del título de **INGENIERO/A BIOMÉDICO/A**.

El citado trabajo de titulación, fue debidamente aprobado por el(los) docente(s):

Tutor	Dr. ALMEIDA GALARRAGA, DIEGO ALFONSO , Ph.D.

Y recibió las observaciones de los otros miembros del Tribunal Calificador, las mismas que han sido incorporadas por el(la) estudiante.

Previamente cumplidos los requisitos legales y reglamentarios, el trabajo de titulación fue sustentado por el(la) estudiante y examinado por los miembros del Tribunal Calificador. Escuchada la sustentación del trabajo de titulación a través de videoconferencia, que integró la exposición de el(la) estudiante sobre el contenido de la misma y las preguntas formuladas por los miembros del Tribunal, se califica la sustentación del trabajo de titulación con las siguientes calificaciones:

Tipo	Docente	Calificación
Tutor	Dr. ALMEIDA GALARRAGA, DIEGO ALFONSO , Ph.D.	10,0
Presidente Tribunal De Defensa	Dra. SALUM , GRACIELA MARISA , Ph.D.	10,0
Miembro Tribunal De Defensa	Dr. ALEXIS FRANK , Ph.D.	10,0

Lo que da un promedio de: **10 (Diez punto Cero)**, sobre 10 (diez), equivalente a: **APROBADO**

Para constancia de lo actuado, firman los miembros del Tribunal Calificador, el/la estudiante y el/la secretario ad-hoc.

Certifico que *en cumplimiento del Decreto Ejecutivo 1017 de 16 de marzo de 2020, la defensa de trabajo de titulación (o examen de grado modalidad teórico práctica) se realizó vía virtual, por lo que las firmas de los miembros del Tribunal de Defensa de Grado, constan en forma digital.*

PEREIRA CARRILLO, JACKELINE PAMELA
Estudiante

GRACIEL
A MARISA
SALUM

Firmado digitalmente por GRACIELA MARISA SALUM
Fecha: 2021.05.28 11:55:23 -05'00'

Dra. SALUM , GRACIELA MARISA , Ph.D.
Presidente Tribunal de Defensa

DIEGO
ALFONSO
ALMEIDA
GALARRAGA

Firmado digitalmente
por DIEGO ALFONSO
ALMEIDA GALARRAGA
Fecha: 2021.05.28
12:12:34 -05'00'

Dr. ALMEIDA GALARRAGA, DIEGO ALFONSO , Ph.D.
Tutor

Firmado electrónicamente por:
FRANK ALEXIS
.

Dr. ALEXIS FRANK , Ph.D.
Miembro No Tutor

KARLA
ESTEFANIA
ALARCON FELIX

Firmado digitalmente por
KARLA ESTEFANIA
ALARCON FELIX
Fecha: 2021.05.28 11:40:01
-05'00'

ALARCON FELIX, KARLA ESTEFANIA
Secretario Ad-hoc

AUTORÍA

Yo, **Jackeline Pamela Pereira Carrillo**, con cédula de identidad 1722723788, declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, definiciones y conceptualizaciones expuestas en el presente trabajo; así cómo, los procedimientos y herramientas utilizadas en la investigación, son de absoluta responsabilidad de la autora del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos de la Universidad de Investigación de Tecnología Experimental Yachay.

Urcuquí, Mayo, 2021.

Jackeline Pamela Pereira Carrillo
CI: 1722723788

AUTORIZACIÓN DE PUBLICACIÓN

Yo, **Jackeline Pamela Pereira Carrillo,** con cédula de identidad 1722723788, cedo a la Universidad de Investigación de Tecnología Experimental Yachay, los derechos de publicación de la presente obra, sin que deba haber un reconocimiento económico por este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior

Urcuquí, Mayo 2021.

Jackeline Pamela Pereira Carrillo
CI: 1722723788

DEDICATORIA

A mi padre, mi hermano, mi madre y mis abuelitas Mariana y Clarita.

AGRADECIMIENTOS

Agradezco a mi familia, por ser siempre mi soporte y el motor para seguir cada día. Gracias en especial a mi padre y mi abuelita que me han hecho la mujer que soy, por estar en cada paso del camino y enseñarme todo lo que se, sin su apoyo incondicional esto no sería posible. A mi hermano por enseñarme que la vida no es solo el objetivo sino el camino que se recorre.

Agradezco profundamente a la vida, que me dio la oportunidad de estudiar esta carrera en esta Universidad, y a todas las personas maravillosas que formaron parte de esta gran experiencia de crecimiento personal e intelectual. En especial a Erick y Fernanda por ser mis compañeros de camino en esta aventura que empezamos juntos llamado Universidad. A Oscar por ser parte de este proyecto y aportar con su tiempo y conocimiento. Y sobre todo a Diego, que ha sido mi compañero de vida y mejor amigo a través de los años, por creer en mí, en mi potencial y sobre todo por apostar por mis ideas siempre.

Agradezco a mi tutor Diego por compartir sus conocimientos conmigo y siempre impulsarme a salir adelante y a mi co-tutor el profe Oscar. A todos mis maestros que me han inspirado en el camino de la ciencia e investigación, sobre todo a Si Amar. Y en general a todos mis amigos y compañeros de carrera por ser parte de estos 5 años espectaculares. Agradezco a todas las personas que fueron parte de toda mi vida universitaria que compartieron conmigo mis logros y mis vicisitudes y también a Dios por cada cosa buena y mala en esta etapa.

Créanme que cada uno de ustedes tiene relevancia en este trabajo y en mi vida, y que este título es un poquito suyo.

RESUMEN

El cáncer de mama es un grave problema de salud mundial al que todos somos propensos, teniendo en cuenta los factores de riesgo a los que estamos expuestos diariamente. Un diagnóstico incorrecto podría traducirse en un mal o inexistente tratamiento y, en el peor de los casos, desembocar en la muerte de un paciente. Hoy en día, los enfoques tecnológicos nos permiten crear y diseñar herramientas para identificar y clasificar estas patologías utilizando métodos de machine learning. Sin embargo, las redes neuronales actuales están diseñadas para identificar y clasificar objetos de imágenes naturales con propiedades diferentes de las que presenta una imágenes médica, lo que provoca que las predicciones hechas a partir de ellos no tengan validez en este campo. Por estas razones, este proyecto de tesis presenta un estudio completo de los enfoques más recientes de un detector de cáncer de mama y un software clasificador, una revisión de comparación entre dos modelos de redes neuronales convolucionales, basada en arquitecturas modificadas con nuestro propio modelo; que pretenden adaptarse a las características únicas de las imágenes médicas basadas en toda la información recogida previamente, para crear una herramienta que podría ser útil para el radiólogo. Este trabajo demuestra la relevancia de esta tecnología, su impacto en el campo médico, y su repercusión e importancia de estas nuevas herramientas para el futuro cercano de la medicina.

Palabras Clave: Redes neuronales convolucionales, Cáncer de Seno, Software de detección, Imágenes médicas.

ABSTRACT

Breast cancer is a serious global health problem to which we are all prone, taking into account the risk factors we are exposed to daily, especially those who work abroad. An incorrect diagnostic could be translated into a bad or inexistent treatment, and in the worst-case flowing into a patient's death. Nowadays, technological approaches allow us to create and design tools to identify and classify these pathologies using Machine learning methods. Nevertheless, the current neural networks are designed to identify and classify natural objects with different properties than medical images have, causing that the predictions made from them do not have medical validity. For those reasons, this thesis project presents a complete study of the most recently approaches of a breast cancer detector and classifier software, a comparison review between two models of convolutional neural networks, based on modified architectures with our own model; that pretend to adapt to the unique characteristics of medical images based in all the information previously collected, to create a tool that could be useful for radiologist. This work proves the relevance of this technology, its impact into the medical field, and its repercussion and importance of these new tools for the near future of medicine.

Key Words: Convolutional Neural Networks, Breast Cancer, Software Detection, Medical Images.

CONTENTS

List of Abbreviations 7

Glossary 9

Figure Index 10

Table Index 12

CHAPTER I : General Introduction 13

Introduction 14

1.1. Problem Statement 14

1.2. Justification 14

1.3. Objectives 15

1.3.1 General Objective 15

1.3.2. Specific Objectives 15

CHAPTER II: State of Art 16

2.1. Breast Cancer 17

2.2. Cancer Detection 19

2.3. Database 20

2.4. Artificial Intelligence (AI) 21

2.5. Supervised Learning 21

2.6. Deep Learning 21

2.8. Residual Neural Network (ResNet) 22

2.9. Medical and Natural Images 23

2.10. Graphic Processing Units storage problems 23

2.11. Repositories 24

2.11.1. Repository 1: An interpretable classifier for high-resolution breast cancer screening images utilizing weakly supervised localization 24

2.11.2. Repository 2: Deep Neural Networks Improve Radiologists' Performance in Breast Cancer Screening. 28

2.11.3. Hybrid Models in both repositories 31
CHAPTER III: Materials and Methods 32
3.1. Database 33
3.2. Conversion from DICOM to PNG format 33
3.3. CNN Models Backbones 34
3.4. Transfer Learning 35
3.5. Importance of Hyperparameters in the model 36
3.6. Experimentation 36
3.6.1. Experiment 1: Classifier Based in ResNet 18 36
3.6.1.1. Dataset and Procedure 36
3.6.1.2. Images Preprocessing 38
3.6.1.3. Hardware selection 38
3.6.1.4. Second Running of Experiment 1: Classifier Based in ResNet 18 39
3.6.2. Experiment 2: Detector and Classifier Approach 39
3.6.2.1. Dataset and Procedure 39
3.6.2.2. KITTY format and Annotations 39
3.6.2.3. Database image´s relabel 41
3.6.2.4. Detector and Classificatory models 42
3.6.2.5. Image Preprocessing 43
3.6.2.6. Hardware selection 44
3.6.3. Experiment 3: Mask R-CNN Model 44
3.6.3.1. Dataset and Procedure 44
3.6.3.2. Feature Pyramid Network 44
3.6.3.3. Bounding Boxes based in Regions of Interest 45
3.6.3.4. Model Architecture 45
3.6.3.5. Transfer Learning adaptation 46
3.6.3.6. Dataset verification 47

3.6.3.7. Configuration and Hyper-parameters Setting 47

3.6.3.8. Image Preprocessing 48

3.6.3.9. Hardware selection 49

3.6.4. Experimentation of Mask R-CNN Model with different Hyper-parameters 49

3.6.4.1. Experiment I 49

3.6.4.2. Experiment II 50

3.6.4.3. Experiment III 50

3.6.4.4. Experiment IV 51

CHAPTER IV: Results and Discussion 52

4.1. Experiment 1: Classificatory ResNet 18 Model 53

4.2. Experiment 2: Detector and Classifier Approach 53

4.3. Experiment 3: Mask R-CNN Model 53

4.3.1. Experimentation of Mask R-CNN Model with different Hyper-parameters 54

4.3.1.1. Experiment I 54

4.3.1.2. Experiment II 55

4.3.1.3. Experiment III 56

4.3.1.4. Experiment IV 56

4.4. Repositories Results 57

4.4.1. Table of metrics 58

4.4.1.1. Repository 1 Model: GMIC-ResNet-18-ensemble 58

4.4.1.2. Repository 2 Model: ‘View-wise’ model 59

4.4.2. Comparisons between repositories and model developed 61

CHAPTER V: Conclusion and Recommendations 62

5.1. Conclusions 63

5.2. Future Works 64

5.5.2. Scientific Divulgation 65

References 66

Attachments 70

LIST OF ABBREVIATIONS

WHO: World Health Organization

SOLCA: Cancer Control Society

MSP: Public Health Ministry

AI: Artificial Intelligence

AUC: Area under the ROC curve

ROC: Receiver Operating Characteristic

CNN: Convolutional Neural Network

MRI: Magnetic Resonance Image

GPU: Graphic Processing Unit

AI: Artificial Intelligence

CAD: Computer Aided Detection

ROI: Region of Interest

GMIC: Globally-Aware Multiple Instance Classifier

ResNet: Residual Neural Network

R-CC: right craniocaudal mammography view

L-CC: left craniocaudal mammography view

R-MLO: right mediolateral oblique mammography view

L-MLO: left mediolateral oblique mammography view

DMV-CNN: Deep Multi-view Convolutional Neural Network model.

ReLU: Rectifier Linear Unit

BI-RADS: Breast Imaging Reporting and Data System

PRAUC: Area Under the Precision-Recall Curve

DICOM: Digital Imaging and Communications in Medicine

TLT: Transfer Learning Toolkit

API: Application Programing Interphase

Bbox: Bounding box

ANN: Artificial Neural network

GLOSSARY

Apoptosis: Programmed cell death.

Incidence: Number of new cases of a disease in a period.

Prevalence: The total number of cases of a disease in a given population at a specific time.

Metastasis: Pathogenic spread from an initial or primary site to a different or secondary site within the host's body.

Tumor: Abnormal growth of tissue.

Virtual container: virtual workstations and servers granted users access to software and computing power across a local network or the internet.

RGB: additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors.

Human epidermal growth factor: Protein that stimulates cell growth and differentiation

Computer-aided detection (CAD): Systems that assist doctors in the interpretation of medical images.

Anchor: Boxes distributed over the image area.

Inference: A conclusion reached on the basis of evidence and reasoning.

FIGURE INDEX

Figure 1: Most common types of cancer according to their incidence in 2018. 7

Figure 2: Breast cancer incidence according women age sets in Quito-Ecuador from 2011 to 2015 . 8

Figure 3: Diagram of the basis of a convolutional neural network architecture. 11

Figure 4: Representation of the calculations between convolutional layers in a ResNet. 12

Figure 5: Diagram representation of the different modules that compose the "GMIC-ResNet-18-ensemble" model. 15

Figure 6: Diagram representation of the architecture of the "view-wise model". 18

Figure 7: Diagram of basic architecture of ResNer 22 (left) and ResNet 18 (rigth) figures. 25

Figure 8: General procedure for Experiment 1 27

Figure 9: Division of the number of images in the three different sets for the model training. 28

Figure 10: loss function of the second run Resnet 18 model. As is visible the error in classification is very high. 29

Figure 11: Representation of the process that allow the creation of bounding boxes and crop the images into the specific site that needs to be identified. 31

Figure 12: Block diagram; correction of the database images process. 32

Figure 13: Labeler "PamPam" Application designed to make Database images correction 32

Figure 14: Configurations for the first model run. The number of images set in the training set, to obtain the KITTI and bounding boxes. 34

Figure 15: Structure of a pyramid network for basic feature extraction. 36

Figure 16: Basic configuration of a deconvolutional neural network. 37

Figure 17: Example of a mask obtained from a dog image; the mask is represented for the right purple figure. 37

Figure 18: Orientation of breast images after the flip procedure. 40

Figure 19: Configuration of the ResNet 18 network, and accuracy obtained after run the model. 44

Figure 20: Metric of the Loss function vs the number of training epoch, as is visible, as much epoch, the loss decrease. 46

Figure 21: Metric of the Loss function vs the number of training epoch in the second experimentation, as is visible, as much epoch, the loss decrease. 47

Figure 22: Metric of the Loss function vs the number of training epoch in the third experimentation, as is visible, as much epoch, the loss decrease. 48

Figure 23: Metric of the Loss function vs the number of training epoch in the second experimentation, as is visible, as much epoch, the loss decrease. 49

TABLE INDEX

Table 1: Main indicators of breast cancer in women in Quito-Ecuador from 2011 to 2015. 7

Table 2: Parameters used in the KITTI format to elaborate the bounding boxes of the images. 30

Table 3: Main characteristics of the run model. 33

Table 4: Main parameters and accuracy obtained after run the model. 44

Table 5: Comparison between the hyper-parameters across the experiments based on Mask R-CNN model. 49

Table 6: Comparison of performance of GMIC model and the baselines on screening mammogram interpretation. 50

Table 7: AUCs of the second repository models on screening and biopsied populations. All models, except the ones* were pretrained on Bi-Rads classification. 51

Table 8: Comparison of different metrics between the paper's models. The most relevant parameters are compared: AUC, database used, if they use or not BI-RADS Pretrain, the PR-AUC and the type of GPU used (its computational power) 52

Table 9: Comparison between the two repositories in three main characteristics AUC, database, and GPUs processing power. 53

CHAPTER I : General Introduction

1. Introduction

Cancer is the word that represents many types of diseases that starts when a healthy normal cell cycle is damaged. The cells can sometimes ignore the signals that normally stop their division or induce them to apoptosis, starting an uncontrollable process to divide themselves, which means that they grow and become invasive and dangerous for the host [1]. In this way, cancerous cells develop a degree of "independence" from the normal control signals. This is a remarkable issue where it is important to say that almost 90% of cancer deaths are related to tumor spreading inside the body (metastasis) [2]. According to the World Health Organization (WHO) [3], breast cancer is one with more incidence rates in women globally.

Although breast cancer mostly affects women, men also can develop it, with an increasing incidence over the course of the past four decades [4, 5]. The etiology of male breast cancer is poorly understood, but apart from known genetic risk, incidence is associated with obesity/ high body mass index, liver disease, testicular illness, alcoholism, radiation exposure among others [6]. Additional distinctions between male and female breast cancer have been demonstrated [7]. In comparison to female breast cancers, male breast cancers show higher rates of estrogen and progesterone receptor positivity and lower rates of human epidermal growth factor receptor positivity [8]. It has also been shown to be molecularly distinct, with differences in gene expression [5].

However, in any case, detection is essential to start the corresponding treatment and fight the illness. Technological advances, especially in the medical field, have been upgrading importance to bring better life quality to the patient. Nowadays, a great number of computer and software solutions are actually in development to fight against this pathology. Cancer, and more specifically breast cancer, has a complex detection in early phases even now with all the mentioned technical help, that include biomedical equipment and image machines. For that reason, it is essential to develop new software tools to help in the diagnosis and increase the accuracy of the radiologists and increase the research in this area that is the future of medicine.

1.1. Problem Statement

Breast cancer is a serious global health problem. For this reason, the diagnosis has to be the most exactly as possible. An incorrect diagnostic could be translated into a bad or inexistent treatment, and in the worst-case flowing into a patient's death. Nowadays, technological approaches allow us to create and design tools to identify and classify these pathologies using Machine learning methods. Nevertheless, the current neural networks are designed to identify and classify natural objects with different properties than medical images, causing that the predictions made from them do not have medical validity. For those reasons, this project presents a comparison between a pre-trained convolutional neural network programmed for natural objects Mask R-CNN, and two modified architectures that pretend to adapt to the unique characteristics of medical images.

1.2. Justification

To achieve a correct diagnosis, a radiologist and an oncologist will see the images and examinations to determine if a tumor is present; it depends a lot on the expertise and experience. Approximately the sensitivity of a radiologist is 62.1% of a correct diagnostic of cancer [9]. Traditional computer-aided detection (CAD) in mammography is commonly used by radiologists to assist with image interpretation, but many studies showing these CAD programs do not improve their diagnostic performance [10]. An attractive alternative to conventional clinical methods is to use machine learning to solve these detection accuracy problems [11]. Many papers show that computational models can perform a better accuracy performance diagnosis than a radiologist [12,13, 14, 15, 16]. These approaches mostly use convolution neural networks (CNN) to detect and classify breast cancer. In the present work, the utility of CNNs in the medical field for detecting breast cancer is presented.

In particular, we intend to show detector and classifier network models that have been modified from natural objects CNNs to specifically work with mammograms, and specialized for cancer detection to improve the accuracy in the diagnosis of breast cancer [17]. The proposed repositories not only talk about the possibility of software, but also a hybrid system of diagnosis joining the training and expertise of radiologists with the accuracy and sensitivity of a trained computer. Thus, the great ideal is to obtain a system that allows to move forward in the diagnosis of these diseases. In this way, I pretend to build a detector and classifier networks from models pretrained with natural images.

To do that, a ResNet architecture will be our backbone to create these models that we expect to work sequentially. In other words, first the detector will detect regions of interest (ROIs) creating masks of them, and then the classifier will predict the correct class. We want to propose not only the possibility of software, if not, a system of diagnosis joining the training and expertise of radiologists with the accuracy and sensitivity of a trained computer. Thus, we finally obtain a system that allows us to move forward in the way that the diagnosis of these diseases works. All of this is to improve the patients' health; because, as we already know, early detection means a quick treatment and, in fact, more chances to survive this illness. Visualizing that it is not only relevant for medical diagnosis, since there has been a growing interest in therapeutic treatment and possible applications in other fields.

1.3. Objectives

1.3.1 General Objective

Compare the metrics obtained between a CNN for a natural object with two repositories that modify the network's architecture to optimize and adapt the processes to breast images to emit a conclusion about the importance of having a specific architecture for detection and classification of medical images.

1.3.2. Specific Objectives

Obtain a database with validated mammography images. To obtain a valid model accuracy, the images that are used must be clinically viable. In this way, we assure that the network will be trained with pictures that help CNN find and classify breast lesions.

Train a pre-trained neural network for natural objects with mammography images divided into three categories, Benign, Benign without call-back, or Malignant. This will be a significant part of the proposed model to posteriorly confirm or discard a cancer diagnosis.

Purpose many experiments of the final model, with variation in their hyper-parameters to compare them to obtain the best possible. In other words, find the adequate parameters to have the best results and the less computational resources.

CHAPTER II: State of Art

In this chapter, the necessary concepts to understand the present work are introduced. In the first place, it is important to know about breast cancer, and specify the definitions of artificial intelligence, supervised learning, and deep learning. After that, the convolution concepts and how a convolutional neural network works, especially a residual one (ResNet) are exposed. Finally, the main point is explaining why the two repositories selected are so crucial for the development of this project by a summary of both sources and how they solve the proposed problems. For this thesis project's development, the first thing to accomplish is using specialized search engines such as Google Scholar Advanced, Scopus, Web of Science, and PubMed to obtain the data. This work is descriptive and systematic analysis; experimental studies are described as a comparative analysis between authors, and a rigorous process (to minimize biases) is used. Also, it identifies, evaluates, and synthesizes scientific articles to show the evolution of deep learning tools for breast cancer screening and finally draw conclusions on the data collected. Documents were selected according to the relevance and contributions to the present work.

2.1. Breast Cancer

Cancer is the second leading cause of death globally and responsible for the surrounding of 9.6 million deaths in 2018. Globally, about 1 in 6 deaths is due to cancer. Almost around 70% of cancers death occurs in low and middle-income countries, which also have a major number of risk factors than in other places. It is also true that the causes of this illness can be many, and there are types of cancer much more aggressive than others [18]. Around the world, breast cancer is the most common in women and the one that more death rates has. In 2018, there were 2,088,000 new cases and 626,000 women died [19]. Two out of ten new cancer cases in women are in the breast, and one in 10 deaths are caused for this illness [20]. In Ecuador, a country with mid-low incomes per capita, this is one of the most frequent diseases in female population according to SOLCA Ecuador, acronym for Cancer Control Society as is visible in **Table 1,** the table does not include information of metastasis. An estimated 2,787 cases were diagnosed, and 821 women died from this cause [21]. In Quito, the incidence rate in the five-year period of 2011-2015 is 39.4 per 100,000 women. In 2018 the most common types of cancers according to their incidence in this population are: breast 2787 (18.2%), cervix 1612 (10.6%), thyroid 1374 (9%), stomach 1225 (8%) colorectal 1123 (7.4%), as can be seen in **Figure 1** [22]. According to MSP (Public Ministry of Health of Ecuador) records, from January to June 2018, there were 1,287 new breast cancer diagnostic cares, of which 1,254 correspond to women representing 97.6% of the cases presented by this pathology. Quito's overall average is among the low-risk regions of 62 out of 69 countries, according to the International Agency for Research on Cancer [23].

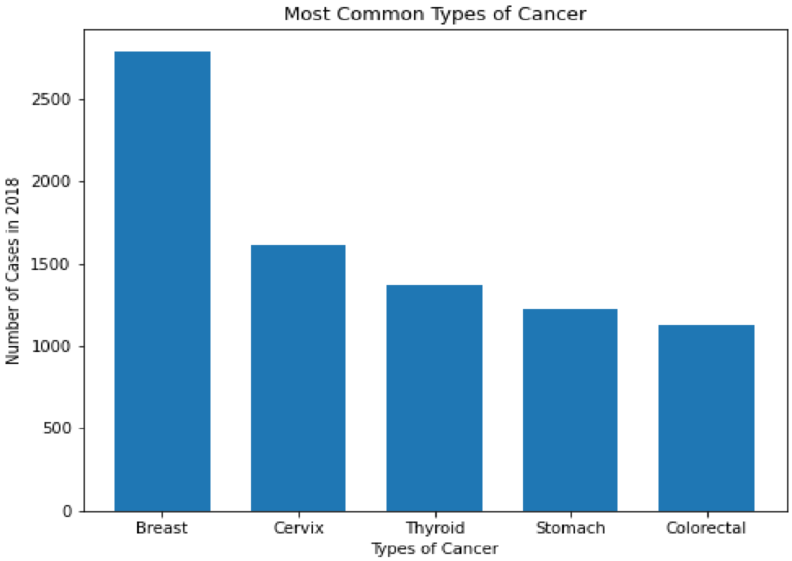


Figure 1: Most common types of cancer according to their incidence in 2018. Developed from [21].

Table 1: Main indicators of breast cancer in women in Quito-Ecuador from 2011 to 2015. Developed from [23].

Indicators	***Women***
Rank	2
Average cases per year	364
Cases percentage	15.1
Average deaths per year	133
Crude incidence rate	39.7
Standarized incidence	39.4
Crude mortality rate	14.5
Standarized mortality	14
Accumulated risk 0-74%	4.6
Mortality/Incidence ratio%	36.4
Histological verification	96.7
Death certificate only	1.8
5 years net survival	84.2

The breasts are tissues made of smaller lobules; each lobule has tiny glands that can make milk for newborns and they are connected through little ducts to the nipple. Fibrous tissue and fat fill the empty spaces; it also exists many lymph nodes near the breast. Cancer cells could appear anywhere [1]. The etiology of breast cancer involves multiple factors such as endocrine and reproductive. Diet, sedentary lifestyle, obesity, alcohol and cigarettes consumption, are also risk factors that should be avoided. One important factor to take into account is age; as is visible in **Figure 2**, as age increases, the rate of cancer incidence increases. [24, 2]. The most common sign for its detection is a new lump or mass in the tissue area, but other symptoms are also possible. It's important to have any breast change checked by a medical doctor as soon as possible. Most of begin breast cancers in the ducts that carry milk to the nipple called ductal cancers and the invasive carcinoma, others starts in the lobules, and the less common are phyllodes tumor and angiosarcoma [26].

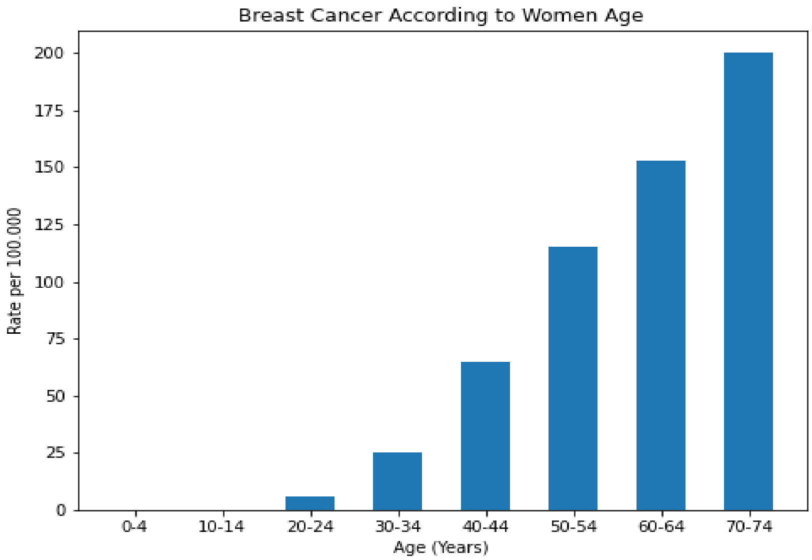


Figure 2: Breast cancer incidence according women age sets in Quito-Ecuador from 2011 to 2015 [27].

Tumors that appear in the breast can be benign and usually not harmful, rarely invade the surrounding tissues, do not spread to other parts of the body and can be removed easily, and usually do not grow back. On the other hand, in general, malignant tumors are exactly the opposite. They can invade nearby organs and tissues, spread themselves to other parts of the body, and in the worst case, if they are removed, they can grow back [25]. For that reason, an examination to determine the type of mass is of the utmost importance. It is true that in the clinical field exist many screening methods like ultrasound, mammography, computed aided-detection, magnetic resonance imaging (MRI), positron emission tomography (PET) scans, among others [18, 28], but the most common is mammography, that consist of an x-rays tube, a molybdenum anode (inside the X-ray tube), a collimator, an x-rays bundle, a compressor, and chassis..

2.2. Cancer Detection

Mammography is the standard method of breast cancer detection. This technique consists of a specialized X-ray picture to visualize the internal structure of the breast [26]. High-quality mammography is the most effective technology for this purpose. The image could be digital (that allows a pre-processing of the parameters of it to has a better visualization or printed in an X-ray film, a highly sensitive material that allows in a short time of X radiation impact have an image of the part of the body that has to be studied [1, 25]. The image depends highly on the tissue that the radiation passes through. The different colors on a grayscale allow the radiologist the possibility to identify what happens in the breast. The following are the five principal densities are recognized on plain radiographs in an increasing density order:

1. Air or gas: gives a black color;

2. Fat: gives a dark grey color;
3. Soft tissues or water: gives a light grey color;
4. Bone: gives an off-white color;
5. Contrast material or metallic material: gives a bright white color [29].

In fact, this exam reduces the mortality product of this illness at a high rate across the years. Important facts about mammography are that the breast's radiographic appearance varies depending on the differences in tissue composition (mammographic density) that causes a difference in the radiological attenuation. It impacts directly into the image, that affects women for the possibility of overlay tissue and, in consequence, omits important details that can result in a wrong diagnosis [30]. Then, the number of cancers escaping mammography detection is substantial, particularly in dense-breasted women, with sensitivity as low as 30–48% per each view, so tumors may not be visualized because of overlying dense tissue. It could also miss posterior cancers in the retro-mammary space because of inadequate deep tissue positioning [31]. In these cases, the medical doctor will suggest more exams, like ultrasound or the invasive biopsy, in which a nail takes a sample of the suspicious tissue and get it to pathology to get a conclusive result.

2.3. Database

Have a database of mammography is a priority for the development of the project. Nevertheless, as with all the medical images, finding a good and extensive database is too complicated for all the legal procedures behind these kinds of pictures. Personal information, lack of acquisition homogeneity and annotation cost, and handling are incredibly delicate. In many cases, direct patient consent is necessary to be able to access them, as hospitals and health centers usually have confidentiality agreements with them that prevent this information from being released. With that in mind, the Database CBIS-DDSM provides convenience images, which are focused on crops of abnormalities in the breast tissue. It means (the portion of the image in the ROI) without needing the full mammography, in this case and depending what is useful for the researcher's needs. To give more accuracy to the ROIs, they applied an automated lesion segmentation algorithm that is initialized by the general original DDSM contours. This program was designed to provide an exact delineation of the mass from the surrounding tissue. An important point to consider is that this segmentation was done only for masses and no calcifications. Lesion segmentation was accomplished by a modification to the local level set framework.

As presented in previous works, even set models follow a non-parametric deformable model, thus handling topological changes during evolution. Chan-Vese is a region-based method that estimates spatial statistics of image regions and finds a minimal energy amount where the model best fits the image, resulting in a convergence of the contour towards the desired object. The author's modifications to the model are on the local framework includes automated evaluation of the local region surrounding each contour point [32]. The images are distributed at the full mammography and abnormality level as DICOM files. Full mammography images include MLO and CC views of the mammograms and subdivide into the four main views of the exam: The L-CC and R-CC, L-MLO, and R-MLO. Abnormalities are represented as binary mask images of the same size as their associated mammograms. These mask images delineate the ROI of each abnormality. An updated version of the DDSM database providing easily accessible data and improved ROI segmentation. This resource will contribute to the advancement of decision support system research in mammography, supplying standardized mammography data. They have some issues by updating the ROI segmentation and by gathering and reformatting the metadata into a more accessible format [32].

The database has four CSV, two for masses (mass) and two for calcifications (calc). The documents have the following information: patient_id (the first seven characters of images in the case file), breast density (Density category, between 1 to 4), left or right breast, image view (CC or MLO), abnormality id (Number of abnormality for the image (This is necessary as there are some cases containing multiple abnormalities.), abnormality type (Mass shape or margin (when applicable)), calc type (Mass margin (when applicable)), calc distribution Calcification distribution (when applicable), assessment BI-RADS assessment, pathology Benign, Benign without call-back, or Malignant, subtlety (Subtlety rating: Radiologists' rating of difficulty in viewing the abnormality in the image), image file path (route), cropped image file path (route), ROI mask file path (route) [33, 32].

2.4. Artificial Intelligence (AI)

Artificial Intelligence is a field of science that describe the capacity of a machine to learn as a human (with all the process involved). Is mainly based on algorithms and models as a technique which is designed based on scientific findings such as math, statists, biology among others branches, these algorithms help the system to determine the expected response which will basically tell the computer what to expect and work accordingly [34]. AI is divided in many sections, including the Machine learning that is based in the construction of a statistical model considered an underlying distribution from which the data is drawn from which can classify it into different categories, it means, a sub-file that focus specifically in helps to make prediction about introduced data, the system learns and is able to develop this process. At the time this is also divided into categories, which include deep learning, supervised learning, unsupervised learning, reinforcement learning, etc. [35].

2.5. Supervised Learning

Supervised learning is a term used to describe a model or algorithm that makes predictions using class or labels distribution in terms of predictor features. The resulting classifier is then used to assign labels of their corresponding class to the testing images where the values of the predictor features are known, but the value of the class label is unknown [36]. The goal of supervised learning is to build an artificial intelligence system that can learn the mapping between the input and the output and predict the system's output given new inputs [37]. As its names show, this field of machine learning is supervised because in the training process, the labels that the data have, indicates to the program which is the class to which this image belongs, and based on that, later the model will be able to determine whether future images belong to it or not. This is a very common type of algorithm used in the develop of classifier and detector software for natural and medical images.

2.6. Deep Learning

Deep learning concept involves a neural network with several layers. As names note, a neural network is a junction of neurons (simple processing units) that get inputs and, based on them, obtain an output, in which inputs are equivalent to neuron's dendrites and outputs like neuron´s axons [38]. All of these are inspired by biological neural network models, this field is especially focus on mimicking the learning and memory processes made for the brain. In general, there are neurons working together to develop complex functions, this is used because is proved that this process helps to living organisms to have conscience, think and act [39]. Deep learning tries to mimic the things that happen on human learning, having an abstraction process. Each layer will have a different "weight," and this weighting reflects what was known about the images' components, and how important is the information that is provided been the heaviest the most relevant parameter.

2.7. Convolutional Neural Networks

A convolutional neural network is mainly designed for processing structured arrays of data such as images, the convolution operations in which are based a great part of these layers, have in consideration the distribution of the information in a matrix space, making them suitable for detection or classification process even in complex images [40, 41]. In the first filters, the CNN obtain the high-level features, like contours, colors, etc., whereas the inner layers extract abstract characteristics as can be seen in **Figure 3.**

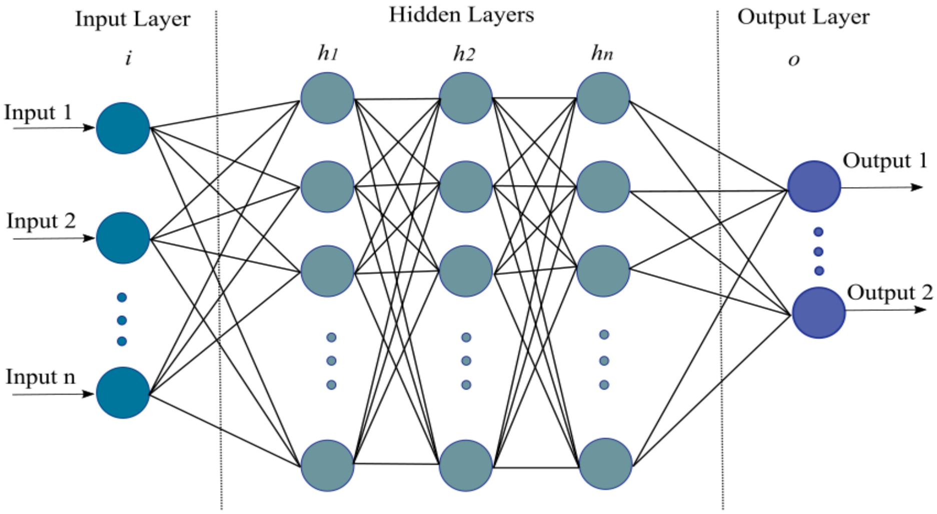


Figure 3: Diagram of the basis of a convolutional neural network architecture, figure developed from [40].

Convolution is one of the most important and effective processes in the analysis of signal processing in general, including image analysis. A convolution is a mathematical operation, more specifically an integral that expresses the amount of overlap of one function as it is shifted over another function producing a third function expressing how the shape of one is modified by the other [42]. The convolutional kernel is a small matrix that has an anchor point that is used to determine the position of the kernel with respect to the image, and numbers in each cell. The anchor point starts at the top-left corner of the image and moves over each pixel sequentially. At each position, the kernel overlaps a few pixels on the image. Each overlapping pair of numbers is multiplied and added. Finally, the value at the current position is set to this sum [43].

2.8. Residual Neural Network (ResNet)

One of the variants of this kind of network is a Deep Residual Learning for Image Recognition named ResNet, which is called “deep” because of its large number of layers that make it produce an improvement in the feature extraction, making it more useful. Is residual because it could be compared to signals from the apical dendrite that skipping over layers, while the basal dendrite collects signals from the previous and/or same layer [44]. It means that those models are implemented with double- or triplelayer skips that contain nonlinearities (ReLU) and batch normalization in between, meaning that the input to some layer is passed directly or as a shortcut to some other layer [40] As is visible in **Figure 4**. The number of convolutional networks that it can have is represented in the name, for example, ResNet18 or ResNet150. The resulting network can resolve many recognitions high-quality issues and is very scalable to maintain the desired balance between inference speed and accuracy [45, 46].

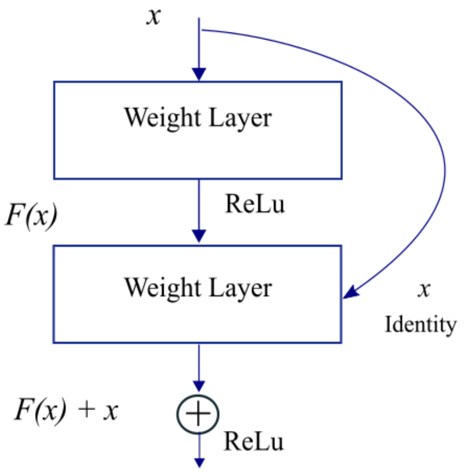


Figure 4: Representation of the calculations between convolutional layers in a ResNet, figure developed from [44].

2.9. Medical and Natural Images

As previously mentioned, it is not the same talk about a medical or a natural image, mainly for two reasons. The first one is the regions of interest ROI. Those are the specific points in the image that we want to identify because the information they can give us is relevant to the investigation. These areas are considerably small in medical images compared with the ones in natural pictures. For that reason, it could be much more complicated for the neural network to recognize them. The second reason and maybe the most important one is that the quality of a medical image is 4294,97 megapixels [45], much higher than a regular image is 4 to 12 megapixels, more pixels more information. Making the image heavier and more challenging for the software to treat [11, 12]. Medical imaging data is scarce, expensive, and fraught with legal concerns regarding patient privacy. It means that the majority of this kind of data is inaccessible for general public research [46]. Nevertheless, although it is hard to access them, it is also essential to get a database with medical validation to do the testing with the CNNs.

2.10. Graphic Processing Units storage problems

With all the previously exposed, the question that arises is: Why a CNN, particularly a ResNet is not enough to identify or classify medical images as mammography? The answer is related to the specific characteristics of those images. The high-quality picture needed to give a diagnosis makes that the weight (memory needed to process) increases into the network and also in the computer. Hence, they occupy a lot of GPU (Graphics processing unit) memory. This specialized electronic circuit is designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. It works in parallel data cache or shared memory shared by all the threads, meaning that it is a programmable processing unit independently from the CPU and responsible for graphics manipulation and, therefore, for the output response [47,48]. It is crucial to think about the limited memory of GPU, and the needed it for the processing of those images, making it not sustainable. Another reason in large part is related to the need for high-quality images is the small region of interest that a medical doctor or radiologist could have in the picture that is of utmost importance for an early diagnosis [49].

2.11. Repositories

The bibliographic search throws many repositories that try to solve the problems previously mentioned. From them, two were chosen for their multidisciplinary approaches and the way that they solve them. The other papers exanimated have different approaches. In the first place, “An interpretable classifier for high-resolution breast cancer screening images utilizing weakly supervised localization” published in 2020, has a very clear point of view about the medical images and that their characteristics differ from natural ones. Starting from that premise, they recognize that neural network architectures that work well for natural images might not apply to medical, so hey purpose changes the architecture and develop a module-based strategy to manage these images. They use mammography to train their model and be able to predict three different classes such as the absence of lesions, benign lesion, and malignant lesion. They use as backbone a ResNet 22 (with 22 convolutional layers), obtaining an accuracy higher than 90% [11]. The following is an overview of both jobs that will then be used to compare the model proposed in this thesis.

2.11.1. Repository 1: An interpretable classifier for high-resolution breast cancer screening images utilizing weakly supervised localization

In the first place, the authors take into account that mammography interpretation is a challenging task even for expert radiologists because most asymptomatic cancer lesions are small, sparsely distributed over the breast, and may present subtle changes in the breast tissue pattern and have many problems like false positives. So the aim is to use this technology to help health professionals. It is fascinating to note that most existing studies on this classification and detection task utilize models initially designed for natural images applied to medical images. However, recently the progression is to adapt them to localize suspicious findings and regions of interest (ROIs). Researchers point many characteristics that difference breast images from natural ones, which are mainly regions of interest ROI in mammography images, such as masses, asymmetries, and microcalcifications, are often smaller in comparison to the salient objects in natural images moreover, as suggested in multiple clinical studies both the local details, such as lesion shape and global structure, among other that are essential for an accurate diagnosis [11].

This contrasts with common natural images where objects outside the salient regions provide little information towards predicting the label of the image; in our case, we have the opposite. Also, it is mentioned that mammography images are usually of much higher resolution than typical natural images. So, the first big problem appears when they release that the most accurate deep CNN architectures for natural images do not apply to mammography images due to the limited size of GPU memory, so a solution is needed and proposed as a first model (in a previous paper) called "Globally-Aware Multiple Instance Classifier (GMIC)." First applies a low-capacity that is memory-efficient; the global module on the whole image generates saliency maps (that show each pixel's unique quality.) that provide coarse localization of possible benign or malignant findings. Achieve to process screening mammography images in their original resolutions while keeping GPU memory manageable. The model also manages the identification of the most informative regions in the image and utilizes a local module with a high-capacity to extract visual details from them and finally predict the presence or absence of benign and malignant breast lesions. While existing other methods often require lesion segmentation during training, this one is trained with only image-level labels and can generate pixel-level saliency maps indicating possible malignant findings [11].

However, they are some failures, so the present paper adds other modules to the GMIC model. This model is based on a customized ResNet. That takes the task of screening mammography interpretation as a multi-label classification problem, which means it is not binary. Proposing a classification framework that resembles the diagnostic procedure of a radiologist, first using a global network to extract a feature map from the input image, to process high-resolution images while keeping GPU memory consumption manageable, we parameterize it as a ResNet-22 because it has one additional residual block and a quarter of the filters in each convolutional layer. Then applying those convolution layers, they transform the feature map into two saliency maps indicating approximate locations of benign and malignant lesions. Each detected element has a particular position on the x and y-axis. After that, the general procedure indicates a down-sampled process, but in this case, the process will distort important visual details such as lesion margins and blurs small calcifications [11].

To avoid this and at the same time reduce the GPU use, they control the consumption of memory in the machine by reducing the complexity of the global network in addition to a high-capacity local network to extract fine-grained details from a set of informative regions. After that, a heuristic patch-selection is an algorithm in which each iteration selects the rectangular bounding box that corresponds to an interest region. The algorithm then maps each selected bounding box to its corresponding location on the input image. It is important to be sure that extracted ROI patches do not significantly overlap with each other. Using the information given from these sections, the machine can focus learning on a selected set of small yet informative patches. Then, apply a local network with a higher capacity (wider or deeper) that is able to utilize fine-grained visual features, to extract a representation from every patch [11].

Since ROI patches are retrieved using the saliency maps, the information relevant for classification carried in each patch varies significantly, then they use the Gated Attention Mechanism (GA), allowing the model to selectively incorporate information from all patches, providing a learnable non-linearity which increases model flexibility. To combine information from both ROI patches and saliency maps, the authors apply a global max pooling, after that is passed to a fully connected layer with sigmoid activation to generate a prediction. In order to limit the saliency maps to only important highlight regions, they make the saliency maps sparser. Despite the complexity of this framework, the model can be trained end-to-end using stochastic gradient descent with the following loss function, defined for a single training. This final step can be explained as a fusion network that combines information from both global structure and local details to produce a prediction. This is analogous to modeling a radiologist that considering the global and local information to render a full diagnosis [11].

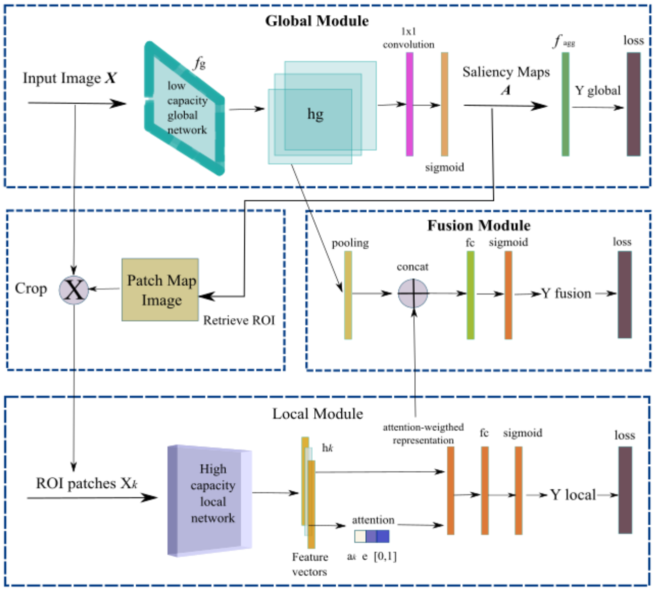


Figure 5: Diagram representation of the different modules that compose the “GMIC-ResNet-18-ensemble" model, figure developed from [11].

To summarize, this model first uses a low-capacity, memory-efficient network on the whole image to identify the most important informative regions. Then it applies another higher-capacity network to collect details from the regions previously chosen. Finally, the algorithm employs a fusion module that aggregates global and local information to make a prediction. The proposed model is able to localize breast lesions in a weakly supervised way, compared with the existing approaches that rely on pixel-level lesion annotations. As is visible in **Figure 5**, initially with the generation of saliency maps (red spots), and after with the patch map indicates positions of ROI patches (blue spots) on the input. In ResNet-22, they denote the number of output channels, strides, and size of the padding. In comparison to canonical ResNet architectures [50], ResNet-22 has one more residual block and only a quarter of the filters in each convolution layer. Narrowing network width decreases the total number of hidden units, which reduces GPU memory consumption.

To demonstrate the effectiveness of the proposed GMIC model on high-resolution image classification, the authors evaluate it on screening mammography interpretation tasks: predicting the presence or absence of benign and malignant findings in a breast. The NYU Breast Cancer Screening Dataset is used to train the networks with 1,001,093 images from 141,472 patients. Taking into account that a mammography exam has four images for each breast R-CC (right craniocaudal), L-CC (left craniocaudal), R-MLO (right mediolateral oblique), and L-MLO (left mediolateral oblique). For each screen, the researchers assign labels corresponding to the absence of malignant and benign findings in both breasts, with all the findings confirmed by at least one biopsy. All images are cropped to a new size of 2944 × 1920 pixels and normalized to have zero mean and unit standard deviation. The model is proved against a standard ResNet 34 model and a variant of this CNN by replacing the fully connected classification layer with a 1×1 convolutional layer. All of this to predict the presence of malignant and benign findings in a breast. In fact, ResNet-34 is the highest capacity model among all those architectures that can process mammography in its original resolution while fitting in the memory of an NVIDIA Tesla V100 GPU.

In addition, it is compared with a previously developed model with Deep Multi-view CNN (DMV-CNN) proposed by that is a previous ResNet-like network dedicated to mammography, which has two versions. In the first one, DMV-CNN applies a ResNet-based model on four standard views to generate two breast-level predictions for each exam. DMV-CNN can also be enhanced with pixel-level heat maps generated by a patch-level classifier. Finally, a model of Faster R-CNN that is a network adapted to localize suspicious findings in mammograms is proved too. They want to prove the level of classification accuracy, number of parameters, computation time, and GPU memory consumption. In addition, the localization performance of GMIC by qualitatively and quantitatively comparing the resulting saliency maps with the ground truth segmentation provided by the radiologists. As each breast is associated with two views images (CC and MLO) and the model generates a prediction for each image, the definition of breast-level predictions is the average of the two image-level predictions. For classification performance, they report area under the ROC curve (AUC) on the breast-level. In addition to accuracy, computation time and memory efficiency are also necessary for medical image analysis. To measure memory efficiency, we report the peak GPU memory usage during training [11].

To compare GMIC to radiologists, they use sensitivity and specificity as evaluation metrics. The first compute is the radiologists' sensitivity and specificity using the data from a reader study. Then the average specificity and sensitivity among readers as the proxy for radiologists' performance is used under a single-reader setting and use the statistics of the consensus reading to approximate the performance. The predictions for the consensus reading are derived using majority voting. The 14 radiologists achieved an average specificity of 85.2% and average sensitivity of 62.1%. The consensus reading yields a specificity of 94.6% and a sensitivity of 76.8%. The performance of the radiologists in the reader study is lower than that for community practice radiologists' performance (Lehman et al., 2016), which reported a sensitivity of 86.9% and a specificity of 88.9% that falls within acceptable national performance standards. At the average radiologists' sensitivity level of 62.1%, GMIC achieves a specificity of 90%, which is higher than the average radiologists' specificity of 85.2% [11].

The paper also talks about the derivation of the ROC/PRC for the average reader by computing the average real positive rate and precision across all readers for every false positive rate and recall. GMIC has 28.8% fewer parameters, uses 78.4% less GPU memory, is 4.1x faster during inference, and 5.6x faster during training, as compared to ResNet-34 while being more accurate. They demonstrate the clinical potential of the GMIC by comparing the improved model to human experts. The AUC for the proposed model was more significant than the average AUC obtained from radiologists by a margin of 11%, reducing the error approximately by half. The proposed model is computationally efficient. It means that GMIC requires significantly less memory and is much faster to train than standard image classification CNN models such as ResNet-34 [11].

Besides, they observe that GMIC can provide meaningful localization when the lesions are hardly visible to radiologists in the image. It is also capable of processing input images in a memory-efficient manner, thus handling medical images in their original resolutions while still using a high-capacity neural network to pick up on fine visual details. Moreover, despite being trained with only image-level labels, GMIC is able to generate pixel-level saliency maps that provide additional interpretability. Evaluated on a large mammography dataset, the proposed model outperforms the ResNet-34 while being 4.3x faster and using 76.1% fewer memory of GPU. Moreover, it is demonstrated that the model can generate predictions that are as accurate as radiologists, given equivalent input information. To conclude with this paper, it is true that it has a great approach to the goal. It is not clinically viable [11].

2.11.2. Repository 2: Deep Neural Networks Improve Radiologists' Performance in Breast Cancer Screening.

The second repository published in 2020, also presents a variant of ResNet architecture for cancer screening exam classification. The customized network makes a balance of depth and width and optimizes them for high-resolution images. This model is specialized to predict the presence of the illness in the tissue. A two-stage architecture and training procedure are combined to take advantage of the high quality of the mammography in each pixel, using a high-capacity patch-level network to learn from pixel-level labels plus network learning from macroscopic breast-level labels as is visible in **Figure 6**. The model also combines multiple input views in an optimal way among several possible choices, getting an AUC of 0.895. The authors also show a hybrid model, is important to mention that many models are presented, but only the one with better output parameters is consider in this summary [12].

It is considered that although mammography reduced breast cancer mortality according to many bibliographic sources, there has been discussion regarding the potential harms of screening. Including false-positive recalls and associated false-positive biopsies. So many women asked to return following an inconclusive screening mammogram undergo another mammogram or ultrasound for clarification that in case of mammography implies another exposition to X-Rays. As was mentioned in the introduction CAD, these programs generally use handcrafted features to mark sites on a mammogram that appear distinct from normal tissue. The radiologist professional decides then whether to recall these findings, determining clinical significance and actionability. But is recently demonstrated that those methods do not improve their diagnostic performance. All of those complications open the gate to deep learning models to create a new generation of tools [12].

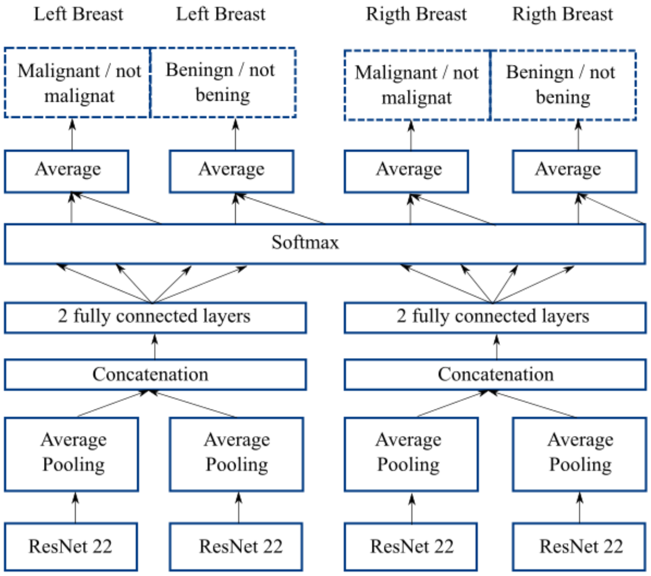


Figure 6: Diagram representation of the architecture of the "view-wise model", figure developed from [12]

In fact, the paper makes several technical contributions to achieve the goal of developing neural networks to support radiologists in interpreting breast cancer screening exams. The model purpose a novel two-stage neural network for incorporating global and local information with an appropriate training procedure, achieving a human-competitive performance and producing interpretable heat maps indicating locations of suspicious findings in the tissue. It also showed the utility of pixel-level labels even in a regime where we have many image-level labels. A remarkable point is the performance of careful error analysis of the obtained predictions. The identification patterns that the network was incapable of capturing to improve architecture in future designs. The utility of pre-training the network is evaluated using a related task with a noisier outcome (screening BIRADS classification) and finds it an essential part of the pipeline that markedly improves the models' performance. This point is of particular significance in medical imaging, where most data sets are small [12].

The authors probe many ways to combine information from different mammographic views within a single neural network. This analysis's results are also of value to the radiologists, particularly about the margin in performance between models trained on a subset of the views because it is as known is common for medical imaging tasks to have multiple inputs. One crucial point of this repository is the option that the researchers give with this contribution to research groups working on improving screening mammography, who may not have access to an extensive training dataset (that is very common). They will be able to directly use this model in their research or use the pre-trained weights as initialization to train models with fewer data [12].

1,001,093 high resolution mammographic images, an extremely large data set in medical imaging, from 141,473 patients. The used images are a carefully curated version of a dataset. Each exam contains at least four images, two correspondings to the four standard views used in screening mammography: R-CC, L-CC, R-MLO, and L-MLO as in the other repository. The images in the dataset are coming from four types of scanners like Mammomat Inspiration (22.81%), Mammomat Novation DR (12.65%), Lorad Selenia (40.92%), and Selenia Dimensions (23.62%). A relevant found is that approximately 32.8% of exams were mammographically occult, meaning that the lesions that were biopsied were not visible on mammography or were identified using other imaging modalities: ultrasound or MRI [12].

In some cases, the same breasts contains both malignant and benign findings. So, the screening cancer classification is seen as a learning task using the multi-task learning framework. For each breast, two binary labels were assigned: the absence or presence of malignant findings in the tissue, and the absence or presence of benign findings in a breast. The goal is to produce four predictions corresponding to the four labels for each exam predicting the presence or absence of benign findings serves a vital role of an auxiliary task regularizing learning the primary task. For the model's inputs, it takes four high-resolution images corresponding to the four standard screening mammography views, crop each image to a fixed size of 2677×1942 pixels for CC views and 2974×1748 pixels for MLO views [12].

As was mentioned, the network architecture consists of two core modules: In the first place, four view-specific columns, each based on the ResNet 22 customized architecture that output a fixed-dimension hidden representation for each mammography view. This network has a different balance of depth and width, which is adjusted to very high-resolution images. The second module is two fully connected layers to map the computed hidden representations to the output predictions. The presented models differ in how the view-specific hidden representations from all views are aggregated to produce the final predictions. A novel variant of a ResNet was used as a building block of the network, specifically designed for medical imaging, which has a balance of depth and width that allows the model to process a massive number of images while maintaining reasonable memory consumption. The L-CC and R-CC representations and L-MLO and R-MLO representations are concatenatenaded. It makes separate predictions for CC and MLO views, which are averaged during inference [12].

The primary consideration in adapting the standard ResNets for mammograms is to process very high-resolution images without prior downsampling–fitting the forward pass and gradient computation within GPU memory. Several changes are needed to create the ResNet-22 model. First, because the hidden representations at the lowest layers are the largest, it is crucial to set the first convolutional layer to have relatively fewer channels: 16 compared to 64 in the standard models. A 5 ResNet block is used, compared to 4 in standard ResNets, doubles the number of channels, our final hidden representation has 256 channels, compared to 512. Finally, a classification layer is applied directly after global average pooling. In our model, we additionally apply two fully-connected layers before the classification layer. On an Nvidia V100 GPU, the model takes about 12 hours to train to the best validation performance (24 hours when using the heatmaps). It is essential to mention that a high amount of training overhead is associated with the time to load and augment with the high-resolution mammography images [12].

High resolution of the images and the limited memory of GPUs problems make the authors use the fine-grained detail in mammograms and trained an auxiliary model to classify 256 × 256-pixel patches of the images, predicting the presence or absence of malignant and benign findings in a given patch. The labels for these patches are determined based on the pixel-level segmentation of the corresponding mammograms produced by clinicians. Subsequently, this auxiliary network was applied to the full resolution mammograms in a sliding window fashion to create two heat maps for each image. One contains an estimated probability of a malignant finding for each pixel, and the other contains an estimated probability of a benign finding. Altogether, obtain eight additional images. These patch classification heat maps can be used like additional input channels, so the network does not have to process the entire high-resolution image at once [12].

Given the small number of biopsied examples with benign or malignant labels, transfer learning was apply to improve the robustness and performance of the models. The transfer learning from a network pretrained on a BI-RADS classification task is applied, this is a numerical scale ranging between 0 and 6 that is used in mammogram, breast magnetic resonance imaging breast ultrasound, which corresponds to predicting a radiologist's assessment of a patient's risk of having breast cancer based only on screening mammography. The three BI-RADS classes consider here are: Category 0 ("incomplete"), Category 1 ("normal") and Category 2 ("benign"). The pretraining on BI-RADS classification contributes significantly to the performance of the proposed model. Using a BI-RADS model, the experimental results trained for 111 epochs (326 hours on four Nvidia V100 GPUs), which obtained an averaged validation AUC of 0.748 [12].

Finally, the model achieved a higher AUC of 0.876 and PRAUC of 0.318. AUCs varies from 0.705 to 0.860. Unfortunately, for the lack of related works with higher databases, the results and procedures obtained cannot yet be classified as clinically usable or not. Making the code and the weights of our model public, the authors want to seek to enable more direct comparisons to their work. The successful model is thanks to the significant amount of computation operations that are encapsulated in the patch-level model, which was applied to the input images to form heat maps as additional input channels to a breast model. The expectation is to continue with the development of models that improve the accuracy and decrease the GPU needed to process and classify the images. They invite other groups to validate our results and test their robustness to shifts in the data distribution [12].

2.11.3. Hybrid Models in both repositories

Now, talking about hybrid models, in both papers, the authors purpose their version of it; in the first one, in addition to the primary model, they experiment with hybrid models that combine predictions from both GMIC and each of the radiologists separately. At radiologists' sensitivity (62.1%), the hybrid models achieve an average specificity of 91.9% improving radiologists' average specificity by 6.3%. The given name is "Human-machine Hybrid," made to demonstrate the clinical potential of GMIC. Its predictions are a linear combination of predictions from each reader and the model. They compute the AUC and PRAUC of the hybrid models by setting $\lambda = 0.5$. Seeing that $\lambda = 0.5$ is not the optimal value for hybrid models. On the other hand, the performance obtained by retroactively fine-tuning λ on the reader study is not transferable to realistic clinical settings. Therefore, the $\lambda = 0.5$ is the most natural way of aggregating two sets of predictions when not having prior knowledge of their quality. For each of the hybrid models, it also calculated its specificity at the average radiologists' sensitivity (62.1%). The 14 hybrid models achieve an average specificity of 91.5%, which is higher than the average radiologists' specificity (85.2%). These results indicate that the model sees different aspects of the task compared to the radiologist's interpretations and can be used as a tool to assist in interpreting breast cancer screening exams [11].

The second papers show a hybrid model in which averaging the probability of malignancy predicted by a radiologist with a prediction of our neural network is more accurate than separately. They also evaluated the accuracy of a human-machine hybrid, whose predictions are a linear combination of predictions of a radiologist and the model–that is, hybrids between each reader and the model achieved an average AUC of 0.891 and an average PRAUC of 0.431. These results suggest that the model can be used as a tool to assist radiologists' personnel in reading breast cancer screening exams and that it captured different aspects of the task compared to experienced breast radiologists. However, in our study, a hybrid model including both a neural network and expert radiologists outperformed either individually, suggesting that the use of such a model could improve radiologist sensitivity for breast cancer detection [12].

CHAPTER III: Materials and Methods

3.1. Database

The public Database CBIS-DDSM is the best option, having 2,620 scanned film mammography studies, which contains 753 calcification cases and 891 mass cases, providing a data-set size capable of analyzing decision support systems in mammography [33]. This CBIS-DDSM (Curated Breast Imaging Subset of DDSM) is an updated and standardized version of another called "Digital Database for Screening Mammography (DDSM)." It contains normal, benign, and malignant cases with verified pathology information, which is an essential factor for the validation of clinical use of the purpose model, giving more relevance to the fact of training using images with a real proven diagnosis.

The collection of mammograms from the following sources: Massachusetts General Hospital, Wake Forest University School of Medicine, Sacred Heart Hospital, and Washington University of St Louis School of Medicine. The DDSM was developed through a grant from the DOD Breast Cancer Research Program, US Army Research and Material Command. The original developers obtained the necessary patient consent. The set includes a subset of the DDSM data selected and curated by a trained mammographer. The images have been decompressed and converted to DICOM format. The pictures are initially in an (LJPEG), an obsolete image format. They use many algorithms and libraries, especially the Stanford PVRG-JPEG Codec v1.1to transform them in addition to a python tool to read this raw data and store it as 16-bit grayscale Tag Image File Format (TIFF) files. Later, they are converted to DICOM format, which is standard for medical images. This process is entirely lossless and preserving all information from the original files. After that, for image processing, DDSM data descriptions provide methods to convert raw pixel data into 64-bit optical density values, which are standardized across all images. Optical density values were then re-mapped to 16-bit grayscale TIFF files and later converted to DICOM format for the data repository [33].

For the download process an application called "NBDIA Data Retriever" is necessary. After its installation, you can choose the type of download that you want for the data. Approximately for all the files, the needed memory for storage is 163.3 GBs (only for images in DICOM format without any treatment). After that, a revision of the database was made, and it is seen that image labels are found to be different from CSV files. So, the image paths specified in the metadata are not useful to access the images. However, the pattern is observed that in the CSV, the images terminated at 00000.dcm are the images that in the names consist of 1-1.dcm and those ending with 00001.dcm represent the images terminated with 1-2.dcm. The problem is fixed, and the CSV is edited with the new routes.

3.2. Conversion from DICOM to PNG format

After that, as is known, DICOM format is specialized for medical equipment and computers that have the needed software to open the files. As is not the case, the option is to transform the format into one that standard computers are able to read, so we have the option of JPG or PNG [50]. As is known, for DICOM binary format (black-white), its grayscale has a broad spectrum of colors, especially in grays tonalities, which helps to a better resolution and visualization of possible tissue lesions. However, in no format, neither jpg nor png, can be maintained all the image characteristics. In addition, it is essential to mention that both repositories also use PNG format to process their database images [11, 12].

In the PNG by exit info, the color scale is maintained to a large extent. Besides, PNG images provide lossless compression in a large size file. PNG format may have several layers of transparency and even include short text descriptions, which help search engines to examine the file [51]. On the other hand, in the JPG format transformation, the colors are lost in a high quantity in the grayscale, losing much quality in the images. For all those reasons and in addition to the fact that the reference repositories also convert their DICOM images into PGN, we decide to go with that format. To make it possible, a python algorithm created for this purpose is used [52]. This program changes the format and the last part of the names of the images to .png. This process reduces a little bit the weight talking about the space used for the database using 163 GBs. After this, the experimentation starts.

3.3. CNN Models Backbones

In the first place, a ResNet 22 model, used as a backbone in both repositories, is searched. However, the researchers' model is not available, and the models available on the internet are not with 22 layers. For the experiment, a ResNet 18, with 18 convolutional layers, which is the most similar for the existent one, is chosen **Figure 7**. Simultaneously, it is one of the most used models for natural object identification and classification. The network needs to get new weights for the medical images that we want to use, so the first idea is to train the CNN with the database. However, the number of images is deficient compared to the one needed to make it network learn and later make decisions and predictions. So, there is when transfer learning appears. Transfer learning from natural image collections is a standard practice that attempts to tackle shape, texture, and color discrepancies all at once through pre-trained model fine-tuning, the model that we want to use is a network module that focuses on color adaptation, thus preprocessing the input into a form (RGB) that is closer to the domain the classification backbone was trained on. [51]

All of this to keep all the characteristics, and the machine can see all the characteristics of the images. The combination of learning from scratch of the color module with transfer learning of different classification backbones, obtaining an end-to-end, easy-to-train architecture for diagnostic image recognition on X-ray images [52]. It should be noted that although it is true, a deeper CNN has larger receptive fields and can capture richer and more complex features in high-resolution images. Narrowing network width can decrease the total number of hidden units, which reduces GPU memory consumption. However, this can presenter other problems like overfeeding, that is to say, that the network only memorizes and cannot take decisions later or make predictions for the extreme amount of data that, as its name indicates it overpowers the network.

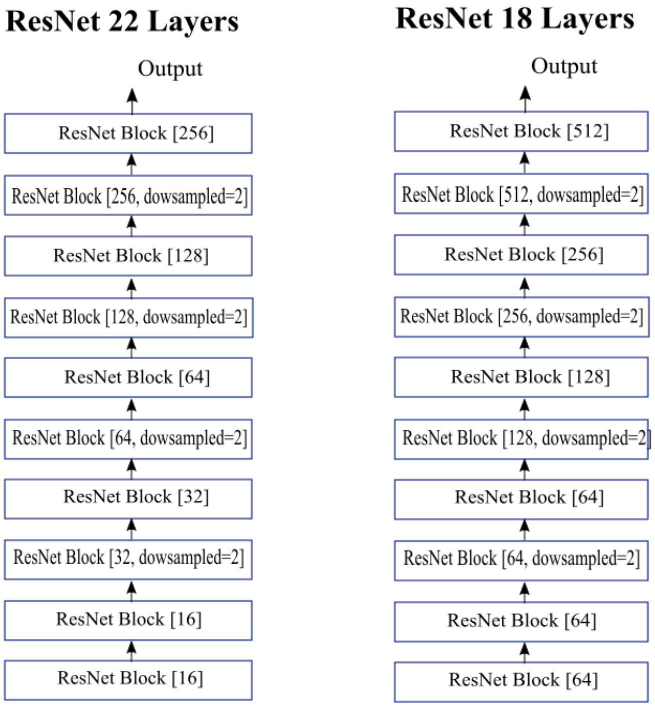

ResNet 22 Layers
Output
ResNet Block [256]
ResNet Block [256, dowsampled=2]
ResNet Block [128]
ResNet Block [128, dowsampled=2]
ResNet Block [64]
ResNet Block [64, dowsampled=2]
ResNet Block [32]
ResNet Block [32, dowsampled=2]
ResNet Block [16]
ResNet Block [16]
ResNet 18 Layers
Output
ResNet Block [512]
ResNet Block [512, dowsampled=2]
ResNet Block [256]
ResNet Block [256, dowsampled=2]
ResNet Block [128]
ResNet Block [128, dowsampled=2]
ResNet Block [64]
ResNet Block [64, dowsampled=2]
ResNet Block [64]
ResNet Block [64]

Figure 7: Diagram of basic architecture of ResNer 22 (left) and ResNet 18 (rigth) figures developed from [12, 52].

3.4. Transfer Learning

The aim is to try to compensate for the lack of enough data to make a goof network training is to transfer knowledge from domains where data are abundant. This task means the development of approaches to bridge the gap between natural and medical images and deal with the distribution shifts due to texture, shape as well as color. The strategy to achieve the goal is basically to take the existing networks like ResNet or ImageNet and simplify the fine-tune their parameters with a limited amount of medical images, in this case with mammographies, in this way, the network maintains a fixed knowledge capacity and the original weights of the model are progressively forgotten in favor of those users to the new domain. It is crucial to take into account that the information coded into the three standard color channels for natural images is hugely different from that saved in a grayscale of X-ray scans, so finding how to recompose this shift provides essential support to learning with limited annotated data. So, a colorization module is introduced, it can be combined with different backbones whose weights are pre-trained on ImageNet. The result is an end-to-end, easy to train architecture in which the color module is learned from scratch. We also apply the fine-tuning concept for ANN, which allows us to retrain neural networks to process actual data features for specific jobs using fewer resources. Which means that by making small adjustments to something (parameters) in the model in order to achieve the best or the desired performance, we can make it.

As many studies say, networks that were trained on ImageNet or other networks that nowadays have rapidly become a standard practice in this field have proven to transfer extremely well to a large variety of tasks that involve medical imaging use. Evidence shows that a pre-trained network is better only if we train it with the images of the part of the body that we want to study, in this case, the breast. In other cases, it means pre-training on another chest X-ray dataset yielded remarkable performance even when fine-tuning on very small datasets, pre-training on images from other anatomical sites like musculoskeletal X-ray did not offer any advantage compared to pre-training on other CNNs, so it remains one of the major strategies for training deep learning models on medical tasks, especially for 2D images [53]. So the proposed model that is used is to let the network learn its optimal color transformation to close the gap while solving the medical recognition task of interest. In this way, the main approaches can take full advantage of all the available data for training at the low cost of updating only a specific part of the network. It does not require any strong assumption that transformations are potentially useful and can be easily adapted to any type of inputs and any pre-trained CNN architecture. The authors define tailored colorization modules for medical images and a multi-stage transfer procedure involving end-to-end update of the main pre-trained backbone [54].

3.5. Importance of Hyperparameters in the model

Hyper-parameters are very important in Deep Learning because they have relevant effects on training a deep neural network. First, a hyper-parameter is an input that is empirically selected according to the specific model that you want to run. They are setting in the concordance of the researcher's needs and the expected function and results of the software. These features have default values and, set these parameters is basically a task and error job. So, you iterate over and over it, tweaking the network, until you find out the specific set of values that will work for your particular problem [55]. In high quantity, the correct setting of these characteristics determines the resultant metrics of the model. This is something that must be taken into account time to perform experiments and run models. Some of those characteristics are optimizer, learning rate, mini-batch size, activation functions, weight initializer, among others.

3.6. Experimentation

3.6.1. Experiment 1: Classifier Based in ResNet 18

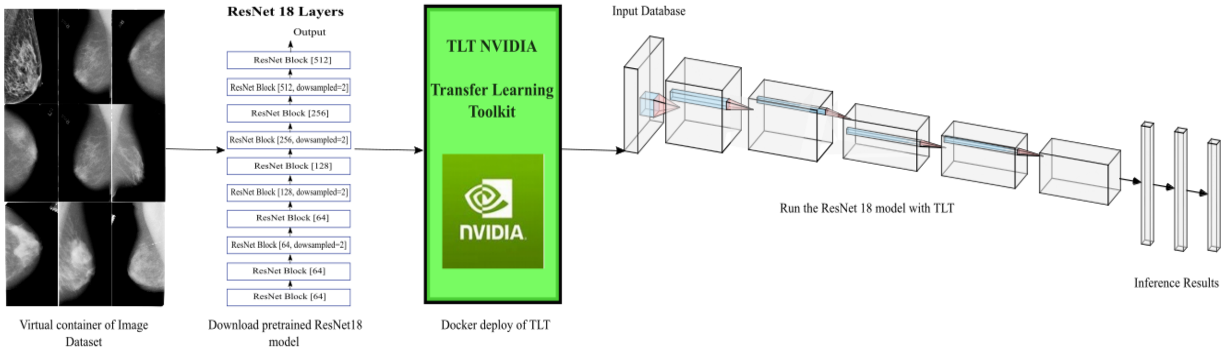
3.6.1.1. Dataset and Procedure

The dataset used is the CBIS-DDSM. Then, the first experimentation is carried out with ResNet 18. The model is obtained from [56]. A virtual container is created to save deployment time, memory in the computer, and above all, preserve existing data on the machine, so in case of damage, the dame data or software is only in the virtual container, and it will not affect the computer. From the model, we take Docker (which is an open-source project that automates the deployment of applications within software containers, providing an additional layer of abstraction and application virtualization automation across multiple operating systems). This link tells you everything you need to run the neural network. All the work of the Transfer Learning Toolkit (TLT) and the application programming interface (API) that basically is a computing interface that defines interactions between multiple software intermediaries. It defines the kinds of calls or requests that can be made, how to make them, the data formats or the conventions to follow, etc.; is done on Jupyter notebook; the images used for this test are the ones that have the complete breast tissue without segmentation. The general experimentation procedure can be seen in **Figure 8.**

For the Network Setup: Set the settings (create an account and set up to create (API) key) and leave everything as default. The only thing that needs to be updated is the key (unique code).

- Data (PNG database), then you put it in Data Directory.
- Then the code unpacks the images and then lists them to verify that they are there.
- Metadata is downloaded to the same directory as the data.

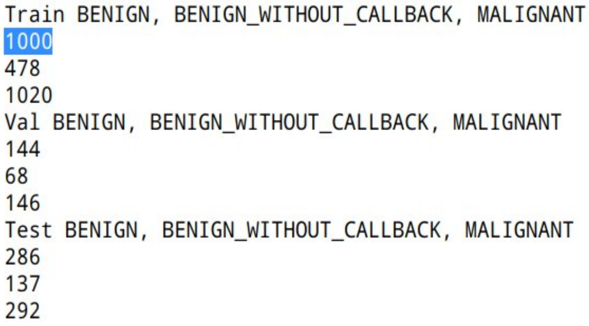
All the theoretical reasons that are listed before about why a “normal” ResNet is not worth it for this classification task are going to be proved experimentally and get a conclusion if they are real or not and if they are true, are easily fixable or not maintaining this architecture.



Virtual container of Image Dataset
ResNet 18 Layers
Output
ResNet Block [512]
ResNet Block [512, dowsampled=2]
ResNet Block [256]
ResNet Block [256, dowsampled=2]
ResNet Block [128]
ResNet Block [128, dowsampled=2]
ResNet Block [64]
ResNet Block [64, dowsampled=2]
ResNet Block [64]
ResNet Block [64]
Download pretrained ResNet18 model
TLT NVIDIA
Transfer Learning Toolkit
NVIDIA
Docker deploy of TLT
Input Database
Run the ResNet 18 model with TLT
Inference Results

Figure 8: General procedure for Experiment 1

After that, it runs, which indicates that it works. Posteriorly, for data split, a folder is created for each pathology class. After that, the images in the database are divided into three directories depending on the pathology (indicated in CSV document) that has the three categories: Benign without a callback (BWC), Benign (B), and Malignant (M). The algorithm reads the path of each image and sees if it belongs to the categories, it stores the picture in a specific directory. The number of images per category is the following (1438 B, 683 BWC, 458 M). Finally, for network training, the total image number is subdivided into three sets (Training, Test, and Validation) where we divide (Training 70% of pictures (Train), Test 20% (Test), Validation 10% (Val)); these percentages are in taken as default parameters of the ResNet docker. Then for each category of training, exist three subfolders for each of the pathologies and the same for the rest sets, having the following information for each of them, which is shown in **Figure 9.**



```
Train BENIGN, BENIGN_WITHOUT_CALLBACK, MALIGNANT
1000
478
1020
Val BENIGN, BENIGN_WITHOUT_CALLBACK, MALIGNANT
144
68
146
Test BENIGN, BENIGN_WITHOUT_CALLBACK, MALIGNANT
286
137
292
```

Figure 9: Division of the number of images in the three different sets for the model training.

Hence, continuing with the Jupyter notebook. The next step is to download the model that is pre-trained (because we are doing transfer learning). I set the model of ResNet is required, in this case, the 18). When the model is completely downloaded, then I set up the file as will train its network, making a copy of the original downloaded ResNet 18 and set some parameters like the size of the image that is the default that comes in the data base (3024 width x 4808 height). The model of the architecture of the model used is for image classification, the NGC model card "TLT Image classification." The extraction of features generated in network training is critical. Those, in theory, can then be trained with any other type of image, including medical images, and would have to work.

As we can observe, the database is labeled, the images are separated into folders, and each one has its own label, so in fact, we are using supervised learning. The model is launched, and it stops almost immediately. The theory is that the image is too big and the GPU used is not enough to process the images, so for the next attempt, the size of the image (as is mentioned we work with the complete image of the mammography) so the size is decreased to the half, to prove our theory that in fact, the size of the image is the cause of the failure in the run of the model. Rerunning the model, it run about 25 minutes and stops again, so we saw that the size and the memory used is the problem. Again the size is decreased in half (standard size divided to 4).

3.6.1.2. Images Preprocessing

In addition to the corrections made in the images' metadata, no pre-processing of the images was done to run this experiment.

3.6.1.3. Hardware selection

The final parameters are: channels 3, width 756, height 1202. So the decision is to use a machine with a high compute capacity is a need for this task. The model (p3.8xlarge) that has 32 nuclei and 240 GB of RAM memory has selected, using 1 of their 4 GPUs Tesla V100, to make the first test and then relaunch the model.

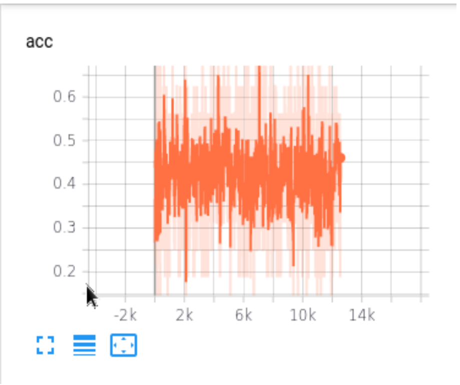


Figure 10: loss function of the second run Resnet 18 model. As is visible the error in classification is very high.

3.6.1.4. Second Running of Experiment 1: Classifier Based in ResNet 18

For the second run of the experiment, the 4 GPU available in the machine are used, which means that now we have 4 Tesla V100 of 16 GB, each one designed for visual computing, but even with the full capacity memory in use, the model run approximately 21 minutes and stops **Figure 10**. The images in their total size are too heavy to work with them. So, if we want to see a real advance, we should consider the possibility of taking a sort of resizing of the images, in which it is space. Therefore, the memory they occupy is more manageable for the machine without losing their original quality.

After analyzing the results obtained and due to the low level of accuracy and the difficulty of working with the complete tissue images, we decided to go for another approach that would be to perform a model that has a detector and classifier modules, i.e., first it will detect the regions of interest (ROI) in mammograms and then analyze whether these regions belong to any of the three classes mentioned above (classify the areas of interest).

3.6.2. Experiment 2: Detector and Classifier Approach

3.6.2.1. Dataset and Procedure

The dataset CBIS-DDSM is used (the complete image and the mask). It is decided to go along this path since this is the most attached approach to what is done when a specialist performs an analysis of these images; first examines the image in search of regions of interest and then determines what they are. In order to train the detector, I need to obtain for each image the annotations of the bounding boxes, that as its name indicates it is a box that encloses the region of interest of each of the ROIs founded within the image. For this purpose, a framework is used, that is, a set of libraries that help to perform a specialized function, in this case, the annotations in each of the images.

3.6.2.2. KITTY format and Annotations

TensorFlow and its implementation TLT of API class of black box are used (in this type of framework, you cannot modify the things that are within the program, the only thing that can be edited is the SPECT file, which is where the training settings like validation set, etc.; are setting. Annotations are made in KITTI format. This is a structure in which the images directory contains the images to train on. The labels directory contains the labels to the corresponding images. A KITTI format label is a file based on a simple text file containing one line per object. Each line has multiple fields, from which we are going to use two of them that are: Class names and Bounding box coordinates. Those features can be seen in **Table 2**. Currently, for detection, the toolkit only needs the two features previously mentioned. This is because the TLT training pipe supports training only for the classes and bbox coordinates. The remaining fields may be set to 0 value [57, 58].

Table 2: Parameters used in the KITTI format to elaborate the bounding boxes of the images, developed from [58].

Parameter name	**Description**	**Num elements**	**Type**	**Range**	**Example**
Class names	The class to which the image allows.	1	String	No range	Malign, Benign.
Bounding box coordinates	Location of the region of interest in the image	4 [xmin, ymin, xmax, ymax]	Float(0 based index)	[0 to image width],[0 to image_height], [top_left, image_width], [bottom_right, image_height]	[100, 20, 15, 400]

To obtain these annotations in this format, we need to take into account the mask and the original image (considering that as mentioned in the data word, there are three images per folder, the complete image of the mammography, the mask of the region of interest and the crop which is the cropping of the image in the region of interest), based on the two images I get a boundary box. To create the annotations is used the "scikit-image" library, which is a collection of algorithms for image processing in python language [59]. With the help of this library, the bboxes of the mask are obtained and superimposed on the original image. This format is only used for the detector part of the model. The process can be visually observed in **Figure 11.**

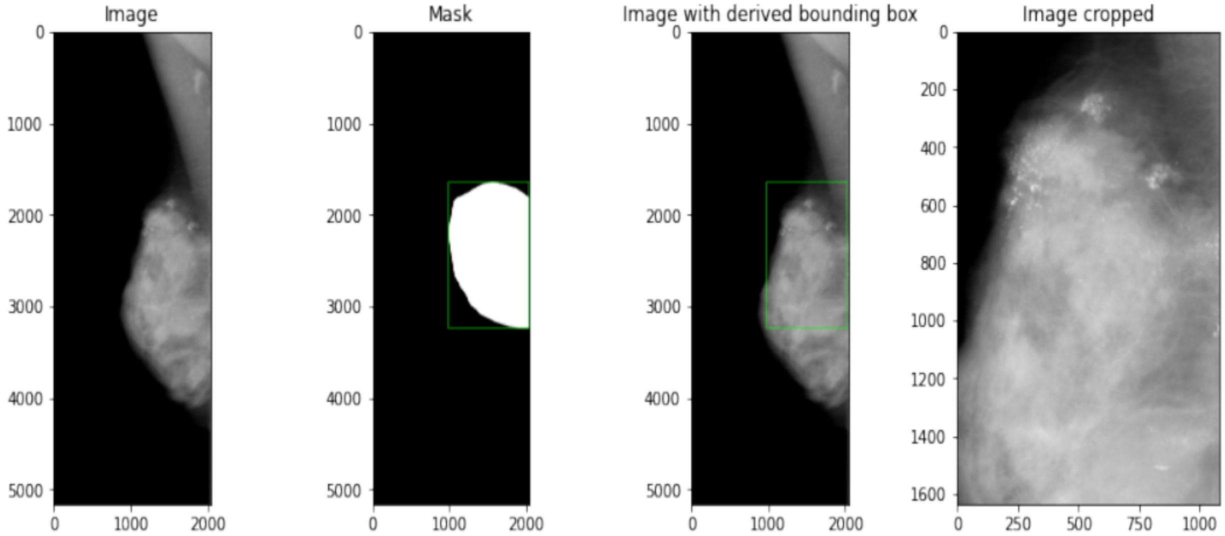
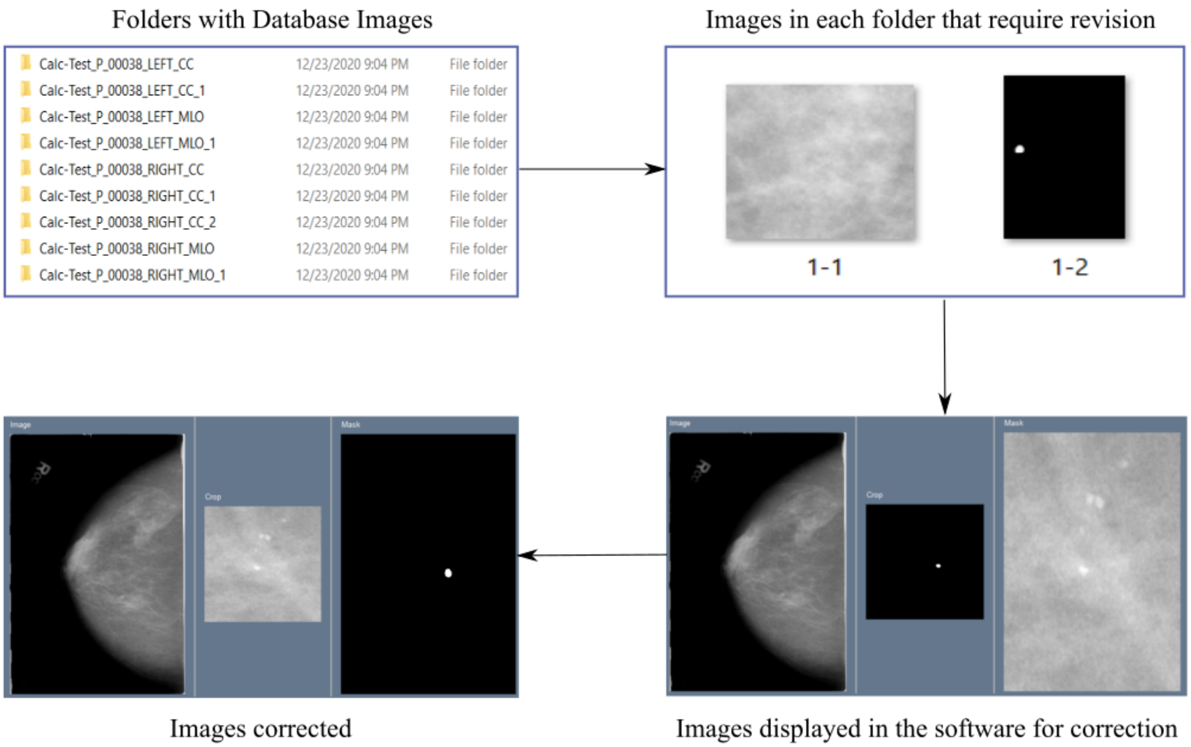


Figure 11: Representation of the process that allow the creation of bounding boxes and crop the images into the specific site that needs to be identified. a) Original image b) Mask of the image c) Bounding box d) Cropped image ready to enter in the CNN.

3.6.2.3. Database image´s relabel

A last check of the database was done, and we release that the crop and mask images are incorrectly labeled. So, in addition to the previous correction of the databases images, it has to be made another one in which the tree images in each folder have to be re-labeled if it is necessary. It is essential to mention that not all the folders have this problem. So, the process has to be made one by one to change the ones that need it. To make this process easier, a desktop application was developed using python; in addition to pysimplegui libraries, whose aim is to be able to visualize the elements of the CSV where the labels and locations of the images are contained for the training, once what have displayed the three images per folder. Labels are reviewed and corrected for errors, and changes are then stored in a new CSV document, as is shown in **Figure 12.**

The software “Labeler PamPam” was designed as an application to display the images in the folders of the database, in this way we can observe if the mask and the crop images that must have the numeration 1-1 and 1-2 respectively, are correctly stablished. In other case, we manually change the numeration to correct the numbers and assure that the images are in a correct position to start the training, as can be appreciated in **Figure 13**. We check one by one each folder, and go to the next with the button NEXT at the end of the names, in addition the application also has a buttom to save the changes and create a new CSV with the correct image name information. There can be clearly shown how looks the database images previous to correction, note that the crop and the mask images are exchanged, and in the posterior image how looks the database images after the correction, the mask and crop images are in their place.



Folders with Database Images

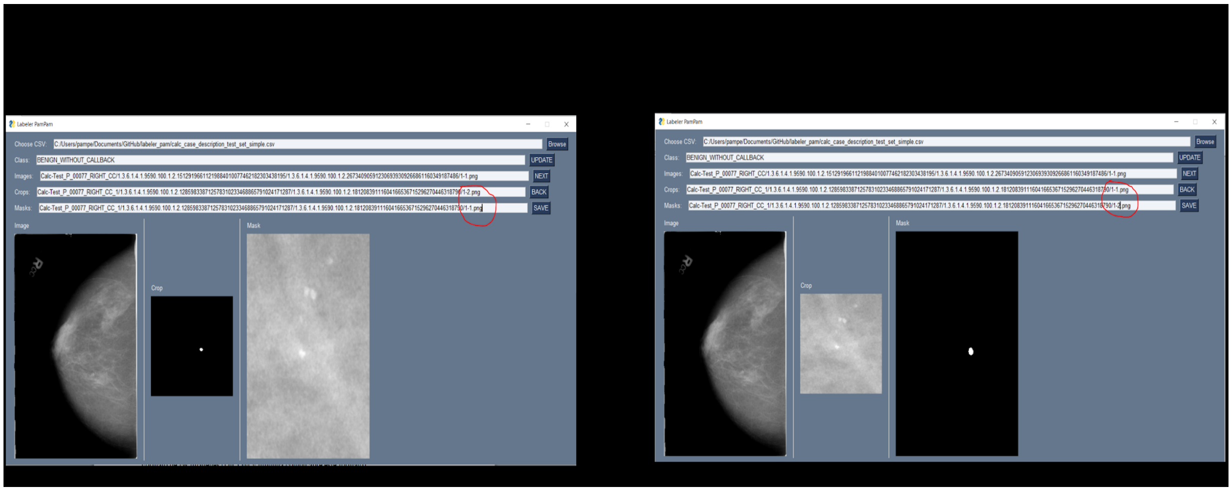
Calc-Test_P_00038_LEFT_CC	12/23/2020 9:04 PM	File folder
Calc-Test_P_00038_LEFT_CC_1	12/23/2020 9:04 PM	File folder
Calc-Test_P_00038_LEFT_MLO	12/23/2020 9:04 PM	File folder
Calc-Test_P_00038_LEFT_MLO_1	12/23/2020 9:04 PM	File folder
Calc-Test_P_00038_RIGHT_CC	12/23/2020 9:04 PM	File folder
Calc-Test_P_00038_RIGHT_CC_1	12/23/2020 9:04 PM	File folder
Calc-Test_P_00038_RIGHT_CC_2	12/23/2020 9:04 PM	File folder
Calc-Test_P_00038_RIGHT_MLO	12/23/2020 9:04 PM	File folder
Calc-Test_P_00038_RIGHT_MLO_1	12/23/2020 9:04 PM	File folder

Images in each folder that require revision

Images displayed in the software for correction

Images corrected

Figure 12: Block diagram; correction of the database images process.



Choose CSV: C:/Users/pampe/Documents/GitHub/labeler_pam/calc_case_description_test_set_simple.csv
Class: BENIGN_WITHOUT_CALLBACK
Images: Calc-Test_P_00077_RIGHT_CC/1.3.6.1.4.1.9590.100.1.2.15129196611219884010077462182303438195/1.3.6.1.4.1.9590.100.1.2.267340905912306939309266861160349187486/1-1.png
Crops: Calc-Test_P_00077_RIGHT_CC_1/1.3.6.1.4.1.9590.100.1.2.128598338712578310233468865791024171287/1.3.6.1.4.1.9590.100.1.2.18120839111604166536715296270446318790/1-2.png
Masks: Calc-Test_P_00077_RIGHT_CC_1/1.3.6.1.4.1.9590.100.1.2.128598338712578310233468865791024171287/1.3.6.1.4.1.9590.100.1.2.18120839111604166536715296270446318790/1-1.png

Choose CSV: C:/Users/pampe/Documents/GitHub/labeler_pam/calc_case_description_test_set_simple.csv
Class: BENIGN_WITHOUT_CALLBACK
Images: Calc-Test_P_00077_RIGHT_CC/1.3.6.1.4.1.9590.100.1.2.15129196611219884010077462182303438195/1.3.6.1.4.1.9590.100.1.2.267340905912306939309266861160349187486/1-1.png
Crops: Calc-Test_P_00077_RIGHT_CC_1/1.3.6.1.4.1.9590.100.1.2.128598338712578310233468865791024171287/1.3.6.1.4.1.9590.100.1.2.18120839111604166536715296270446318790/1-1.png
Masks: Calc-Test_P_00077_RIGHT_CC_1/1.3.6.1.4.1.9590.100.1.2.128598338712578310233468865791024171287/1.3.6.1.4.1.9590.100.1.2.18120839111604166536715296270446318790/1-2.png

Figure 13: Labeler "PamPam" Application designed to make Database images correction

With the corrected images, the KITTI´s annotations (bboxes) are obtained. After that, the CSVs previously corrected are joined into one single document, from which the crops are selected and copy into a new folder, and also keep the images in a single folder. After that, the detection process can be started.

3.6.2.4. Detector and Classificatory models

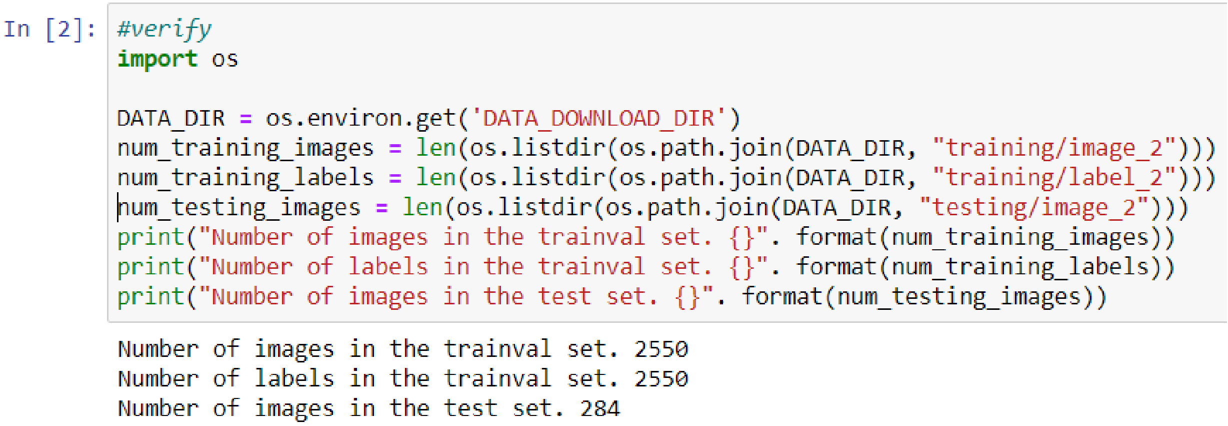
In the first place, we are going to try a train detector model named "TLT DetectNet V2 Detection" that has a ResNet 18 as a backbone and is pre-trained with Google Image Data Set, obtained in [60]. This model was selected from 6 meta-architectures. It is basically a Transfer learning toolkit based on object detection pre-trained weights. It helps to accelerate the AI training process and eliminates the time-consuming process of training from scratch. It means that this model takes purpose-built pre-trained AI models and customizing them with your own data, and accelerates the training process, and reduces computational costs associated with large-scale data collection, labeling, and training models. Considering that ResNet is a network properly used for the classification task, this is an adaptation for detection task, so it will have a major accuracy percentage of 79 as is visible in **Table 3,** in addition to some important parameters, as the GPU model, the number of epochs for training. The accuracy number can also be explained by the fact of the 18 convolutional layers of the network **Table 3**, so you don't get many features of the images to analyze. ResNet makes additive convolutions, making possible that the operations are easier to process and use a low number of convolutional layers to decrease the memory used and make the model easy to run.

Table 3: Main characteristics of the run model.

Main Characteristics of the Model	Quantity
Accuracy Reached	79%
Batch Size	1
GPU Model	Tesla V 100
Number Of Epochs	80
Version Id	ResNet18

So, once the model is selected, we start to set parameters, in which we configure the classes that are going to be used, the number of epochs and the annealing that is the percentage from which the machine is going to decide if is an interest region or not, in this case, the 70%. We prepare tf records from kitti format dataset spec file, so the first place is to update the tfrecords that is a simple format for storing a sequence of binary records, to take in your KITTI format dataset and create the tfrecords using the tlt-dataset-convert. The Tf-Records only need to be generated once. Then we provide training specifications for the train datasets, augmentation parameters for on the fly data augmentation, and other training (hyper-) parameters such as batch size, number of epochs, learning rate, etc. [61]. After that, we move the training images (2550) and the labels of those images into two separated folders, and the same for the 284 testing pictures as is shown in **Figure 14**.

We set our data in the model and change the extension from .kitti to .txt as a requirement of the software. Set the code and run the TLT training. The model was launched with a single GPU and had an estimated time of 7 hours with 43 minutes. The model does not work properly and stops after 3 hours **Figure 14.**



```
In [2]: #verify
        import os

        DATA_DIR = os.environ.get('DATA_DOWNLOAD_DIR')
        num_training_images = len(os.listdir(os.path.join(DATA_DIR, "training/image_2")))
        num_training_labels = len(os.listdir(os.path.join(DATA_DIR, "training/label_2")))
        num_testing_images = len(os.listdir(os.path.join(DATA_DIR, "testing/image_2")))
        print("Number of images in the trainval set. {}". format(num_training_images))
        print("Number of labels in the trainval set. {}". format(num_training_labels))
        print("Number of images in the test set. {}". format(num_testing_images))

        Number of images in the trainval set. 2550
        Number of labels in the trainval set. 2550
        Number of images in the test set. 284
```

Figure 14: Configurations for the first model run. The number of images set in the training set, to obtain the KITTI and bounding boxes.

3.6.2.5. Image Preprocessing

For that reason, we decide to make a deeper image preprocessing, that is basically the use of computer algorithms to perform image processing on digital images. All of this is to improve the image data (features) by suppressing unwanted distortions and/or enhancement of some important image features, so the models can benefit from this improved data to work. The problem with the model is that, if the output image height and the output image width of the preprocessing block doesn’t match with the dimensions of the input image, the dataloader either pads with zeros, or crops to fit to the output resolution. It does not resize the input images and labels to fit. So, data augmentation has to be made as an individual process. We use this to increase the amount of data by adding slightly modified copies of existing or newly created synthetic ones. The augmentation process does not include a resize path. This is necessary to solve the detector model's problem, making the annotations (bbox) match the newly created images.

3.6.2.6. Hardware selection

One of the main concerns was choosing the right hardware that would allow me to have enough computing resources for the models to run. The first option was to test the ASUS TUF GAMING A15 laptop with GTX 2060via a remote connection to a Jupiter Server, but it did not work since the operating system of the computer is Ubuntu desktop, which is not design for server connections because it has a lot of limitations about it ports and have a lot of restrictions, and the necessary one is an Ubuntu Server to be able to perform the necessary actions. Below, we try to create a GCP (Google Cloud Platform) in Microsoft Assure, but eh quota of our region is very limited. We can not acquire more quota to achieve the goal is of creating a Deep Learning Virtual Machine to start to run the models, which means that it does not allow the parameters that we need of the GPUs and RAM needed.

Not having enough processing capacity and taking into account that for monetary reasons could not continue to have access to model (p3.8xlarge) with 32 nuclei, 240 GB of RAM memory, and 4 GPU Tesla V100, and not having any viable machines, the purpose model is not viable.

3.6.3. Experiment 3: Mask R-CNN Model

3.6.3.1. Dataset and Procedure

We continue with the use of the same dataset. In this case, we are going to use the complete image and the mask image. The new model proposed is the backbone used in the second repository Deep Neural Networks Improve Radiologists' Performance in Breast Cancer Screening" [12]. The Mask R-CNN model that means mask regional convolutional neural network is a two stages software: the first stage scans the image and generates areas that contain an object. And the second stage, this classifies the proposals and generates bounding boxes and masks. This model has a very interesting thing called Instance segmentation, which is the task of identifying object outlines at the pixel level, which means that it could identify the presence of an object in addition to the pixels that it occupies in the image. The standard neural network used is typical as the backbone, ResNet50, or ResNet101 that serves as a feature extractor. The first layers detect low level features edges and corners, and the subsequent layers successively detect higher-level features mass, calcifications, etc [62].

3.6.3.2. Feature Pyramid Network

FPN Feature Pyramid Network improves the standard feature extraction pyramid by adding a second pyramid that takes the high-level features from the first pyramid and passes them down to lower layers, as is seen in Figure 15. By doing so, it allows features at every level to have access to both, lower and higher level features. Doing this allows features at every level to have access to both lower and higher-level features. So the Mask R-CNN uses a ResNet101 + FPN backbone. The (RPN) Region Proposal Network is a light neural network that scans the image and finds areas containing objects. Instead, the RPN scans over the backbone of the feature map [63]. This allows the RPN to reuse the extracted features efficiently and avoid duplicate calculations. With the mentioned optimizations, the RPN runs faster and finally generates two outputs for each anchor: Anchor Class and Bounding Box Refinement. After that, the final proposals regions of interest ROIs that allow us to pass to the next stage.

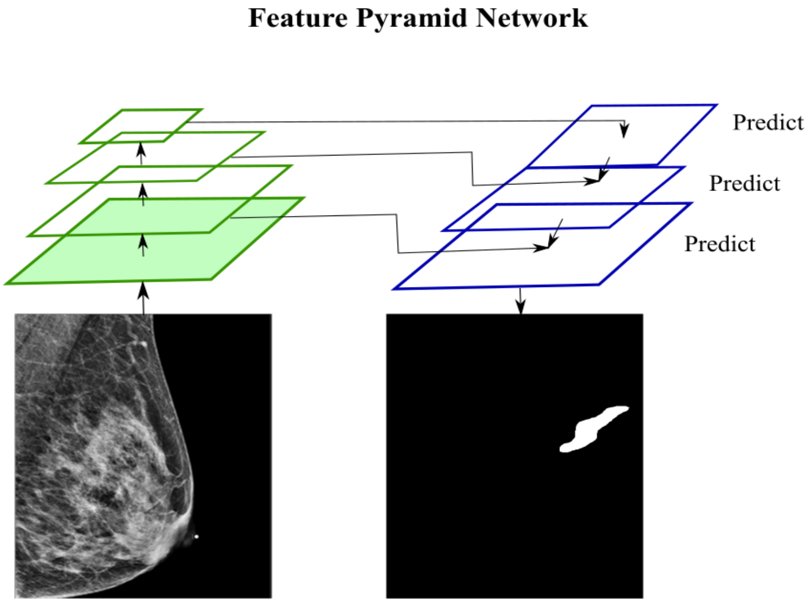

Feature Pyramid Network

Figure 15: Structure of a pyramid network for basic feature extraction, figure developed from [63]

3.6.3.3. Bounding Boxes based in Regions of Interest

Then, the ROI Classifier & Bounding Box Regressor in this stage runs on the (ROIs) proposed by the RPN. Furthermore, it generates two outputs for each ROI. So, the object's class is a significant output as the Bounding Box Refinement, which aims to refine the location and size of the bbox to encapsulate the object. Then, pooling is necessary because classifiers and detectors do not handle variable input size very well. They typically require a new input size. Due to the bounding box refinement in the RPN, the ROI boxes can have different sizes. That's where ROI Pooling helps. It mainly refers to cropping a part of a feature map and resizing it to a fixed size. It's similar in principle to cropping part of an image and then resizing it. The authors suggest a method they named ROIAlign, in which they sample the feature map at different points and apply a bilinear interpolation [64]. Then, with the creation of a segmentation mask, we can correctly say that the mask network is the thing that makes the Mask R-CNN itself. This convolutional network takes the positive regions selected by the ROI classifier and generates masks for them.

3.6.3.4. Model Architecture

We decide to use this model for it architecture. The use of a convolution network fully linked with a deconvolution one, makes that in a general view, the model can extract a lot of features from the input image by the increasing channels in each layer and at the same time reduce the filters used (in the convolution network). After all those extraction, the information is used as in input in the deconvolution network, in which the previously obtained characteristics are used to make a very defined mask of the edge of the object detected. All of this make the obtained mask of the ROIs are very precise by their pixel by pixel information [65]. We decide to use this model for its architecture. The use of a convolution network fully linked with a deconvolution one **Figure 16,** makes that in a general view, the model can extract a lot of features from the input image by the increasing channels in each layer and at the same time reduce the filters used (in the convolution network). After all those extractions, the information is used as input in the deconvolution network, in which the previously obtained characteristics are used to make a very defined mask off the edge of the object detected. **Figure 17**. All of this makes the obtained mask of the ROIs are exact by their pixel-by-pixel information.

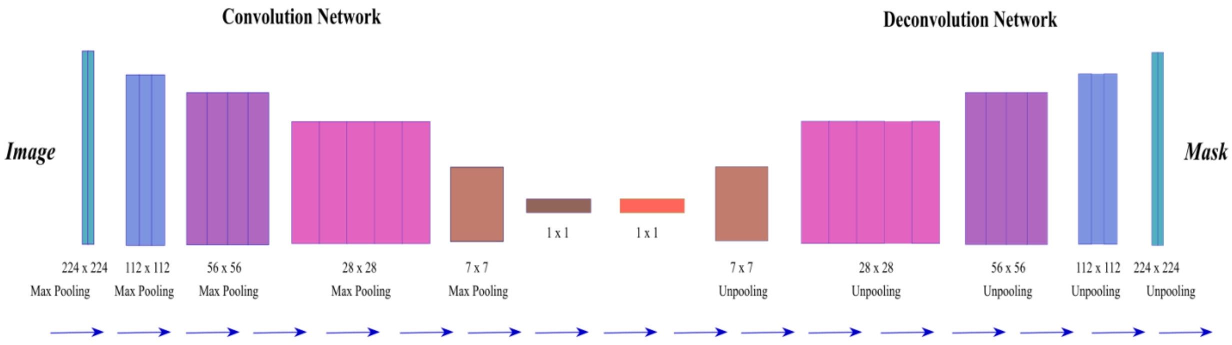
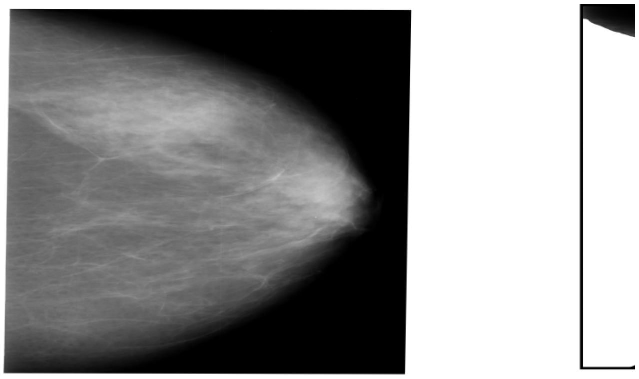


Figure 16: Basic configuration of a deconvolutional neural network, figure developed from [66].



Normal Image

Figure 17: Example of a mask obtained from a dog image; the mask is represented for the right purple figure.

3.6.3.5. Transfer Learning adaptation

In this model, they firstly apply transfer learning, using the COCO dataset to start with a weights file that's been trained instead of Google Image Net in the last model. The author wants to train the model with balloon images. Nevertheless, COCO does not contain any balloon class. It contains a high volume of other images, so the trained weights have already learned a lot of the features common in natural images, which helps not to need a lot of balloon images to train the net. The thing is, this can be used in our experimentation because we also use the transfer learning with the COCO dataset and then adapt it to our database of mammography. So first, with all the procedure explained before, the initial step is to load the dataset, but we have one advantage and is that our data also include the mask images of the mammography images, so we do not need to label them again in a JSON file that is the indicated for the repository, if not we can give the mask directly to the models only changing our data type from .png to .npz it helps with the need for the software to set the mask as a set of polygon points. The format .npz are NumPy structures that we later could transform into tensors that will be used as inputs in the network.

3.6.3.6. Dataset verification

To verify if the dataset is implemented correctly, we develop all of this process in a Jupyter notebook, where the actions are visible, and we can follow the advances. Following the process, the configurations are setting in which we specify the parameters in which the model is going to base for running, so we needed to override three values, as is visible in the following code. In the name, we set a descriptive one, “mammography,” after that, in the number of classes, we set four. The three that we want to detect and classify into benign, malign, benign without a callback, and the last one called “Background,” in which all the noise means all the values that could not be classified in the other three classes goes to this category. The steps per epochs are the number of times that the data is going to enter into the network for the train.

3.6.3.7. Configuration and Hyper-parameters Setting

The batch size and number of epochs are hyper-parameters commonly confused, but basically, an epoch is composed of one or more batches. First of all, Stochastic Gradient Descent is an optimization algorithm used to train CNNs. Its main task is to find a set of parameters in the model that perform well against some performance measures like logarithmic loss (Loss) or mean squared error. The optimization algorithm is called gradient descent refers to calculating an error slope and moving down along that slope towards finding some minimum level of error. This algorithm occurs multiple times, each time with the expectation to slightly improving the software parameters. The procedure is different depending on the type of algorithm, but in artificial neural networks ANN, a backpropagation update algorithm is used [65].

So, the batch size defines the number of samples to work before updating the inner model parameters. The batch could be seen as a loop iterating over one or more samples and making predictions. At the end of this “loop,” the predictions are compared to the expected output, and an error is calculated. The batches can have different sizes, and when the size is more than one sample and less than the size of the whole training dataset, the learning algorithm is called mini-batch gradient descent. Generally talking batch sizes include 32, 64, and 128 samples [67].

The base configuration uses size in input images of 1024x1024 px for best accuracy. We resize them to the standard size fixed in repository 1 [11]. Now, for training, we have to consider that the Mask R-CNN is a large model. Especially taking into account, its implementation uses ResNet101 and FPN [62]. So, for the hardware selection, we must have, as a minimum, a GPU with 12GB of memory. Training should start from the pre-trained COCO weights. The code will be downloaded and the weights from the repository automatically.

It is important to take into account that, for use the COCO´s weights, we can not use the same network structure because the fully connected layers are configured to the exact number of classes of the COCO dataset, so the thing that we do for using these weights is excluded: fully connected layers, logics, bboxes and all the parameters that are setting according to the other classes and retrain them according to the classes that we need for our mammography images. The model accuracy depends mostly on the dataset that we are going to use. If the images have a satisfactory resolution, light, etc. is easier for the software to identify and classify the different objects that the picture has, in our case, if there are abnormal masses or calcifications or not.

With all the previous exposed, the purposed mode efficiently detects objects in an image while in simultaneously generating a high-quality segmentation mask for each picture instance. It is simple to train and adds only a small overhead to other models as Faster R-CNN, running at 5 fps. With a simple, quantization-free layer, called RoIAlign, it can preserves exact spatial locations. Is very important to mention the advantage of this model against other are the following features: RoIAlign has a large impact in improves mask accuracy by relative 10% to 50%, showing bigger gains under stricter localization metrics. Second, decouple mask and class prediction. The authors of the original Mask R-CNN found it essential to predict a binary mask for each class independently, without competition among classes, and rely on the ROI classification branch to predict the category [67].

All of this to improve the instance segmentation performance that is going to be reflected in the obtained results. The authors bet to the fast train and test speeds, together with the framework's flexibility and accuracy, will benefit and ease future research on instance segmentation. For that reasons we choose it to solve the problems posed and fulfill the objective of this thesis project. Basically, the Mask R-CNN consists of two stages. The first stage called a Region Proposal Network (RPN), proposes candidate object bounding boxes. The second stage extracts feature using RoIPool from each candidate box and perform classification and bounding-box regression in parallel to predicting the class and box offset.

3.6.3.8. Image Preprocessing

Continuing with the previous experiment's preprocessing, the images resize to the dimension 2944 × 1920 has to be done both in the mask and in the standard image. To do this, we have to apply some mathematical formulas to draw the relationship between the x-axis and y-axis to this ratio, then multiply it or divide it according to the resize applied to the images, and then the overlap of the mask to the can be made full breast image. Later, a flip is applied to all the images with a left orientation, so all the databases are shown into the right orientation, as shown in **Figure 18**. For this purpose, we use a Python OpenCV code [68] that is basically a library of programming functions mainly aimed at real-time computer vision.

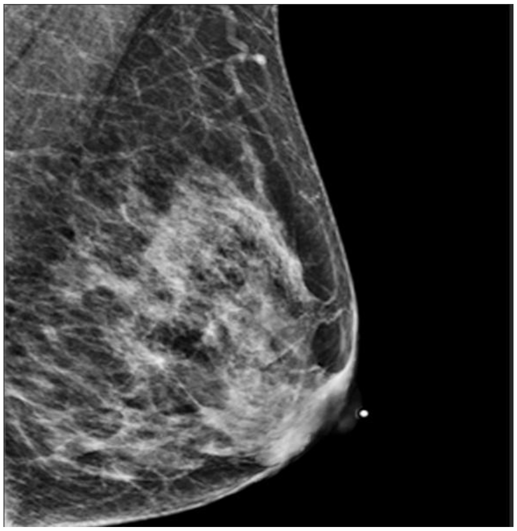


Figure 18: Orientation of breast images after the flip procedure.

3.6.3.9. Hardware selection

For this model and having the latest experience of failure searching a high-performance computer, we decide to use Google Collaborator PRO to run the pre-processing of the data and posteriorly the model. This allows us to use virtualization instances, which means that we have access to an aggrupation of many CPUs, graphic units, RAM memories, etc., that is called a cluster, which gives us an adequate computer capacity. So, in this platform, we have 20 GB of RAM, fulfilling the basics requirement of the model that is 12 GB. The importance of this parameter is that how many images we can make inference simultaneously depend entirely on the RAM memory.

3.6.4. Experimentation of Mask R-CNN Model with different Hyper-parameters

In this section, I am going to describe the experimental results obtained after training the network. First, one of the most important steps that are needed to find the best possible model is the setting of hyperparameters. For this specific case of use, the tunning of hyperparameters was made in a class of configuration called MamographyConfiguration(Configuration), which inherits the Configuration class of Mask R-CNN. In this class, the following parameters were set:

- GPU count: which refers to the numbers of GPUS used to rain or infer the model
- Images per GPU: which states the number of images loaded in a GPU at a time.
- Number of epochs: which is the number of times we are going to feed the model with the dataset to train.
- Number of steps per epoch: the number of batch iterations before a training epoch is considered finished.
- Min confidence value: This is the minimum value of confidence to consider a detection as valid.

With this hyperparameters, a set of different configurations are tested initializing the experiments with:

3.6.4.1. Experiment I

This experiment was performed in a Google Colab PRO instance with a single Tesla P100 GPU. Then the hyperparameters configuration was:

- GPU count: 1
- Images per GPU: 1
- Number of epochs: 50
- Number of steps per epoch: 50
- Min confidence value: 0.8

This experiment was executed with one image per GPU to test how much video memory is allocated when a single image is loaded. I have also used this first experiment to test the computing time using a single GPU. Besides, this first test is necessary to make a diagnosis to determine if the model is learning at least over the training dataset. This feature can be verified if the loss function (categorical cross entropy) is decreasing after finishing each epoch. Therefore, the training loss has to be plotted to visualize its behavior over each epoch. We can also detect some overfitting if after a section of decreasing in the loss function it starts to increase undefinedly.

3.6.4.2. Experiment II

This experiment was performed in a Google Colab PRO instance with a single Tesla P100 GPU. Then the hyperparameters configuration was:

- GPU count: 1
- Images per GPU: 4
- Number of epochs: 100
- Number of steps per epoch: 100
- Min confidence value: 0.9

This experiment was executed with four image per GPU to verify how much video memory is allocated and how much is free. Besides, this second test is necessary to determine if the model learns faster about training data set with four images per GPU. This feature can be verified if the loss function (categorical cross entropy) is decreasing after finishing each epoch. Therefore, the training loss has to be plotted to visualize its behavior over each epoch. We can also detect some overfitting if after a section of decreasing in the loss function it starts to increase undefinedly.

3.6.4.3. Experiment III

This experiment was performed in a Google Colab PRO instance with a single Tesla P100 GPU. We have used offline and online augmentation of the dataset. The offline augmentation was focused in horizontal flip to get all the images in the same orientation. Then, the configuration for the online augmentation trigger blur in the image, gaussian noise, and random changes in the brightness and contrast of the image. Then the hyperparameters configuration was:

- GPU count: 1
- Images per GPU: 4
- Number of epochs: 100
- Number of steps per epoch: 100
- Min confidence value: 0.9

This experiment was executed with four image per GPU to verify how much video memory is allocated and how much is free. Besides, this second test is necessary to determine if the model learns faster about training data set with four images per GPU. This feature can be verified if the loss function (categorical cross entropy) is decreasing after finishing each epoch. Therefore, the training loss has to be plotted to visualize its behavior over each epoch. We can also detect some overfitting if after a section of decreasing in the loss function it starts to increase undefinedly.

3.6.4.4. Experiment IV

This experiment was performed in a Google Colab PRO instance with a single Tesla P100 GPU. We have used offline and online augmentation of the dataset. The offline augmentation was focused in horizontal flip to get all the images in the same orientation. Then, the configuration for the online augmentation trigger blur in the image, gaussian noise, and random changes in the brightness and contrast of the image. Then the hyperparameters configuration was:

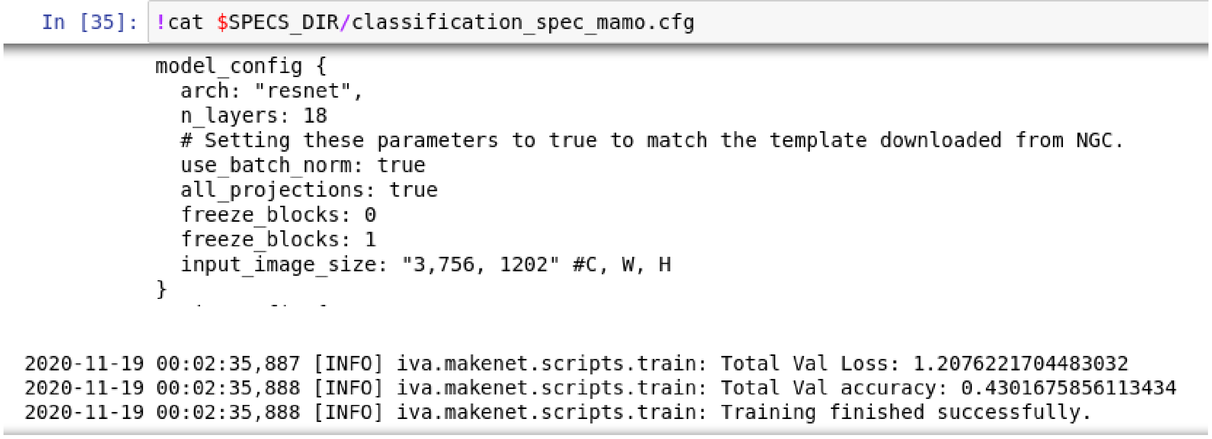
- GPU count: 1
- Images per GPU: 4
- Number of epochs: 50
- Number of steps per epoch: 50
- Min confidence value: 0.9

This experiment was executed with four image per GPU to verify how much video memory is allocated and how much is free. Besides, this second test is necessary to determine if the model learns faster about training data set with four images per GPU. This feature can be verified if the loss function (categorical cross entropy) is decreasing after finishing each epoch. Therefore, the training loss has to be plotted to visualize its behavior over each epoch. We can also detect some overfitting if after a section of decreasing in the loss function it starts to increase undefinedly.

CHAPTER IV: Results and Discussion

4.1. Experiment 1: Classificatory ResNet 18 Model

The model runs completing the 80 epochs (1 epoch is when the whole data amount means all the images enter the network and pass for all the CNN layers). Giving the following results:



```
In [35]: !cat $SPECS_DIR/classification_spec_mamo.cfg
```

```
           model_config {
             arch: "resnet",
             n_layers: 18
             # Setting these parameters to true to match the template downloaded from NGC.
             use_batch_norm: true
             all_projections: true
             freeze_blocks: 0
             freeze_blocks: 1
             input_image_size: "3,756, 1202" #C, W, H
           }
```

```
2020-11-19 00:02:35,887 [INFO] iva.makenet.scripts.train: Total Val Loss: 1.2076221704483032
2020-11-19 00:02:35,888 [INFO] iva.makenet.scripts.train: Total Val accuracy: 0.4301675856113434
2020-11-19 00:02:35,888 [INFO] iva.makenet.scripts.train: Training finished successfully.
```

Figure 19: Configuration of the ResNet 18 network, and accuracy obtained after run the model.

Configuration of the first run of ResNet 18 architecture. The name of the architecture, number of layers and size of the images, and the general configurations are visible **Figure 19.** In addition, the figure at the bottom shows the AUC percentage after complete 80 epochs of training. After 48 hours of the model run, it gives a final accuracy (AUC) of 43.02%. A lot of information is loose for the high reduction of the size of the image. Also, and according to the obtained graphics, apparently, the network does not learn from the new data if it does not infer from the weights, it already had. The main parameters of the model are summarized in **Table 4.**

Table 4: Main parameters and accuracy obtained after run the model.

Parameters	*Quantification*
Model Used	ResNet 18
Input image (channels, weight, high)	(3, 756,1202)
Number of Epoch	80
Total Accuracy	0.43 or 43.02%

4.2. Experiment 2: Detector and Classifier Approach

Not having enough processing capacity and not having any viable machines, we cannot obtain results for this model.

4.3. Experiment 3: Mask R-CNN Model

In the previous experiments, we want to obtain the AUC metrics based on the number of false positives in the two repositories that we have shown before. Understanding that the number of true positives is the number that the method finds, the number of positives is the total number of positives areas that can be found. The number of false positives is the number of negative points that your method classifies as positive. So, this helps us in future comparisons between our model and the presented by the other investigations. Nevertheless, in this case, the metrics that are used in this model Mask R-CNN are different, and instead, to present the AUC, they present the Categorical Cross-Entropy as a loss function. This is a measurement that is built upon entropy that is the number of bits required to transmit a random selected event from a probability distribution.

It generally calculates the difference between two probability distributions and is mainly used to optimize classification models like logistic regression or artificial neural networks. So, the categorical cross-entropy builds upon the idea of entropy from information theory and calculates the number of bits required to represent or transmit an average event from one distribution compared to another distribution. In other words the cross entropy is the average number of bits needed to encode data coming from a source with distribution a when we use model b. Where A is the image dataset as input of the network and B is the model Mask RCNN [68]. So, the categorical cross-entropy builds upon the idea of entropy from information theory and calculates the number of bits required to represent or transmit an average event from one distribution compared to another distribution. In other words, the cross-entropy is the average number of bits needed to encode data coming from a source with distribution a when we use model b. Where A is the image dataset as the input of the network and B is the model Mask R-CNN [69].

If we consider a target or underlying probability distribution A and an approximation of the target distribution B., Then the cross-entropy of B from A is the number of additional bits to represent an event using B instead of A. The significant impact of using it instead of other methods as the sum-of-squares for a classification problem leads to faster training as well as improved generalization. Suppose the model can estimate the probability of an example belonging to each class label. Cross-entropy can then be used to calculate the difference between the four probability distributions in our case in a multi-class classification task. The calculated probabilities are two, the Expected Probability, that is, the known probability of each class label for an example in the dataset (A). The Predicted Probability represents the probability of each class label an example predicted by the model (B).

Therefore, we can estimate the cross-entropy for a single prediction using the cross-entropy calculation, which is the basic principle of this process. Then, for each actual and predicted probability, the prediction must be converted into a distribution of probabilities across each event. In this case, the classes 0 and 1 as one minus the probability for class 0 and probability for class 1. The model can then calculate the cross-entropy and repeat the process for all the samples. Finally, it calculates the average cross-entropy across the dataset and reports it as the cross-entropy loss for the model's model. In simple words, Loss is used to calculating the gradients in the model. Moreover, gradients are used to update the weights of the neural network. This is how a CNN is trained [70].

4.3.1. Experimentation of Mask R-CNN Model with different Hyper-parameters

4.3.1.1. Experiment I

In this experiment we have reached a lot of valuable outcomes to lead the next experiment in that way. The first thing we have to notice is that our model is learning from the training set. The loss function is decreasing through the training dataset. We can observe this in the following figure.

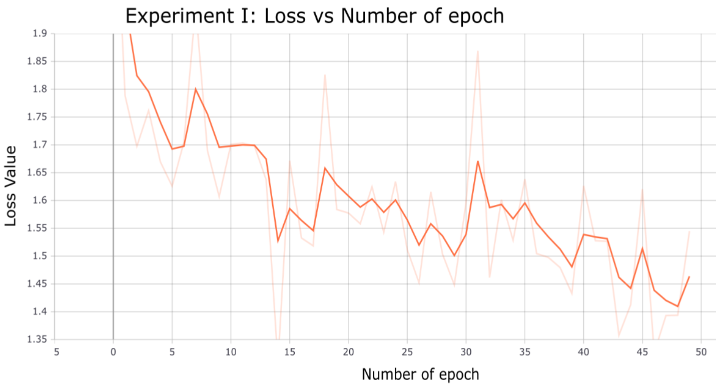


Figure 20: Metric of the Loss function vs the number of training epoch, as is visible, as much epoch, the loss decrease.

Here we can observe how the function is decreasing from 1.9 to almost 1.35 as the number of epochs increases. The figure is using a smoothing of 0.6 to appreciate in a better way the tendency of the line **Figure 20**. Besides, the model trained with one image per GPU just used 6 Gb of video memory from 16 Gb available. This result means we can train with more images per GPU and increase the performance of the training and reduces the computing time. Finally, we have seen that there is not overfitting with this configuration until the 50th epoch, then a bigger number of epochs and number of steps per epoch can be used to obtain smaller values in the loss function.

4.3.1.2. Experiment II

In this experiment we have reached a lot of valuable outcomes to lead the next test. The first thing we have to notice is that our model is learning from the training set, faster than the last experiment. Because the number of images in the GPU decreases the loss function faster. We can observe this in the following figure:

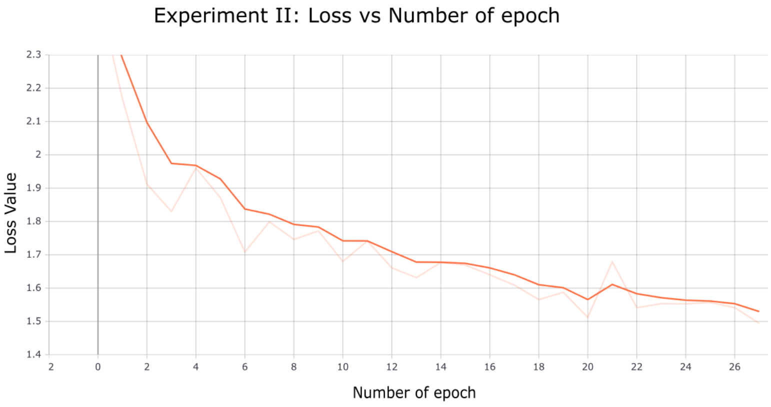


Figure 21: Metric of the Loss function vs the number of training epoch in the second experimentation, as is visible, as much epoch, the loss decrease.

Here we can observe how the function is decreasing from 2.168 to almost 1.4 in just 27 epoch. The figure is using a smoothing of 0.6 to appreciate in a better way the tendency of the line **Figure 21**. Besides, the model trained with four image per GPU just used 10 Gb of video memory from 16 Gb available. This result means we can train with more images per GPU but we do not know if this will increase the performance of the training and reduces the computing time. Finally, we have seen that there is not overfitting with this configuration until the 27th epoch, then a bigger number of epochs can be used to obtain smaller values in the loss function. The lowest validation value for the loss function was 0.9

4.3.1.3. Experiment III

In this experiment the load time of the data was bigger and notable around 15 min because we introduce offline and online augmentation.

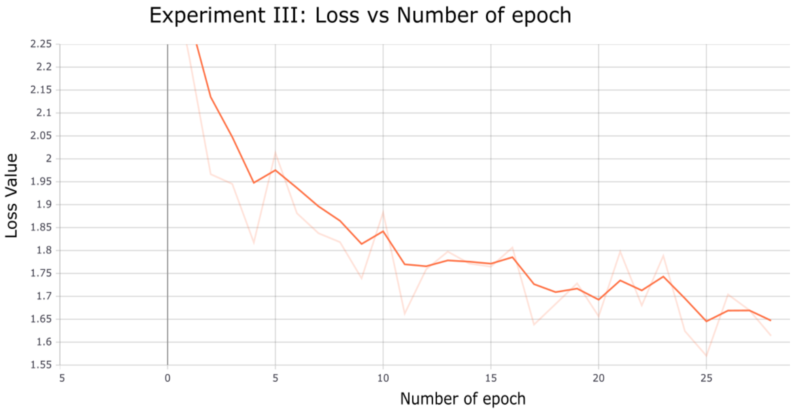


Figure 22: Metric of the Loss function vs the number of training epoch in the third experimentation, as is visible, as much epoch, the loss decrease.

Here we can observe how the function is decreasing from 2.25 to almost 1.55 in just 25 epochs. The figure is using a smoothing of 0.6 to appreciate in a better way the tendency of the line **Figure 22**. Besides, the model trained with four image per GPU just used 10 Gb of video memory from 16 Gb available. Because we have used a bigger dataset the training time was increased almost five times which tell us that we probably need more time per epochs and more of them to get a more accurate training. However, because Colab sessions just last 24 hours the training was stopped in the 26th epoch. Finally, we have seen that there is not overfitting with this configuration until the 26th epoch. The lowest validation value for the loss function was 0.6.

4.3.1.4. Experiment IV

In this experiment the load time of the data was bigger and notable around 15 min because we introduce offline and online augmentation.

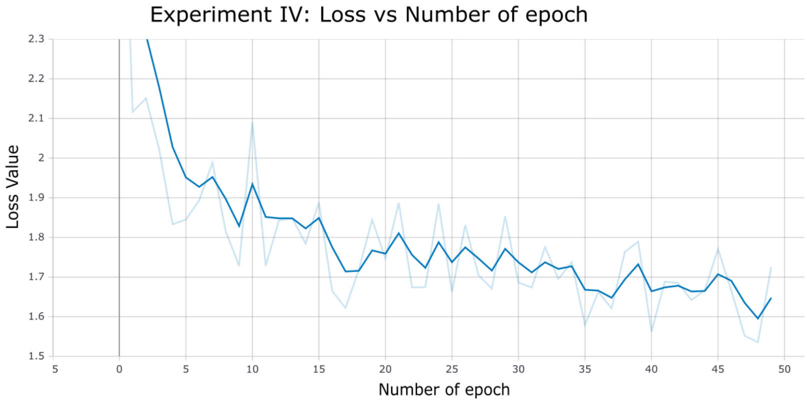


Figure 23: Metric of the Loss function vs the number of training epoch in the second experimentation, as is visible, as much epoch, the loss decrease.

Here we can observe how the function is decreasing from 2.3 to almost 1.55 in 50 epochs. As we can see the use of less steps per epoch make the training faster but less accurate, and less efficient if we correlate the accuracy with the training time. The figure is using a smoothing of 0.6 to appreciate in a better way the tendency of the line **Figure 23**. Besides, the model trained with four image per GPU just used 10 Gb of video memory from 16 Gb available. Because we have used a bigger dataset the training time was increased almost five times which tell us that we probably need more time per epochs and more of them to get a more accurate training. Finally, we have seen that there is not overfitting with this configuration until the 50th epoch. The lowest validation value for the loss function was 0.7.

4.3.1.5. Comparison between the experiments

Table 5: Comparison between the hyper-parameters across the experiments based on Mask R-CNN model.

Hyper-Parameters	***Experiment I***	***Experiment II***	***Experiment III***	***Experiment IV***
GPUs	1	1	1	1
Images x GPU	1	4	4	4
Number of Epoch	50	100	100	50
Lowest validation Loss function value	1.35	0.9	0.6	0.7

4.4. Repositories Results

In the first repository " An interpretable classifier for high-resolution breast cancer screening images utilizing weakly supervised localization ", to demonstrate the effectiveness of the GMIC model on high-resolution image classification, the authors first evaluate it on the task of screening mammography interpretation, predicting the presence or absence of benign and malignant findings in a breast. According to [11]. Besides, it also evaluates the localization performance of GMIC by comparing the resulting saliency maps with the ground truth segmentation provided by the radiologists. On the other hand, the second repository "Deep Neural Networks Improve Radiologists' Performance in Breast Cancer Screening" compares the accuracy of the best of the four purpose models against radiologists with the same data. In addition, they also show a hybrid model, averaging the probability of malignancy predicted by a radiologist with a prediction of our neural network. Finally, they conduct a thorough analysis of the customized network's performance on different subpopulations of the screening population, the model's design, training procedure, errors, and properties of its internal representations [13, 14].

4.4.1. Table of metrics

4.4.1.1. Repository 1 Model: GMIC-ResNet-18-ensemble

Table 6: Comparison of performance of GMIC model and the baselines on screening mammogram interpretation. For both GMIC; AUC test was reported (mean and standard deviation) of 5 best models that achieved highest validation AUC in identifying breasts with malignant findings [11].

Model	*AUC(M)*	*AUC(B)*	*#Paramet*	*Memory (GB)*	*Fwd/Bwd(ms)*	*FLOPs*
ResNet 34+fc	0.736±0.026	0.684±0.015	21.30M	19.95	189/459	1622B
ResNet 34+1X1 conv	0.889±0.015	0.772±0.008	21.30M	12.58	201/450	1625B
DMV-CNN(with/o heat maps)	0.827±0.008	0.731±0.004	6.13M	2.4	38/86	65B
DMV-CNN(with heat maps)	0.886±0.003	0.747±0.002	6.13M	2.4	38/86	65B
Faster R-CNN	0.908±0.014	0.761±0.008	104.8M	25.75	920/2019	
GMIC-ResNet-18	0.913±0.007	0.791±0.005	15.17M	3.01	46/82	122B
GMIC-ResNet-34	0.909±0.005	0.790±0.006	25.29M	3.45	58/94	180B
GMIC-ResNet-50	**0.915**±0.005	**0.797**±0.003	27.95M	5.05	66/131	194B

GMIC-ResNet-18 -ensemble	**0.930**	0.800	-	-	-	-
GMIC-ResNet-34 -ensemble	0.920	0.795	-	-	-	-
GMIC-ResNet-50 -ensemble	0.927	**0.805**	-	-	-	-

4.4.1.2. Repository 2 Model: 'View-wise' model

Table 7: AUCs of the second repository models on screening and biopsied populations. All models, except the ones* were pretrained on Bi-Rads classification [12].

		Screening population				***Biopsied population***			
		Single		5x ensemble		Single		5x ensemble	
		Malignant	Benign	Mal	Ben	Malignant	Benign	Mal	Ben
Image-only	View-wise	0.827±0.008	0.731±0.004	0.840	0.743	0.781±0.006	0.673±0.003	0.791	0.682
	View-wise	0.687±0.009	0.657±0.006	0.703	0.669	0.693±0.006	0.564±0.006	0.709	0.571
	Image-wise	0.830±0.006	0.759±0.002	0.841	0.766	0.740±0.007	0.638±0.001	0.749	0.642
	Breast-wise	0.821±0.012	0.757±0.002	0.836	0.768	0.726±0.009	0.639±0.002	0.738	0.645
	Joint	0.822±0.008	0.737±0.004	0.831	0.746	0.780±0.006	0.682±0.001	0.787	0.688
Image-and-heat maps	View-wise	**0.886±0.003**	0.747±0.002	**0.895**	0.756	**0.843±0004**	0.690±0.002	**0.850**	0.696
	View-wise	0.856±0.007	0.701±0.004	0.868	0.708	0.828±0.008	0.633±0.006	0.841	0.640
	Image-wise	0.875±0.001	**0.765±0.003**	0.885	0.774	0.812±0.001	0.653±0.003	0.821	0.658
	Breast-wise	0.876±0.004	0.764±0.004	0.889	**0.779**	0.805±0.004	0.652±0.004	0.818	0.661
	Joint	0.860±0.008	0.745±0.002	0.876	0.763	0.817±0.008	**0.696±0.005**	0.830	**0.709**

According to the results expressed in metric tables of both repositories, it is visible that the first paper is the one that has the most specificity in terms of parameters that were used and compared around all models, including AUC, Number of Parameters, peak GPU memory usage (Mem (GB)), time is taken for forward and backward propagation in milliseconds Fwd/Bwd (ms), and the number of floating-point operations FLOPs. While the other only presents accuracy data in two different populations, the screening, and biopsied ones. In addition, the first paper makes comparisons against different networks **Table 6,** and the purpose model GMIC, giving more information about other networks for natural objects and their performance. The second paper only presents the AUC results of screening predictions (mammographies) evaluated by the network (model) versus the results of biopsies of tissues sent and evaluated by a pathologist, **Table 7**.

For logistics classification problems, the AUC metrics check the model performance, the higher is better. However, any value is considered good, and the larger the means the model is behaving great. It is used in classification analysis in order to determine which of the used models predicts the classes best. An example of its application is ROC curves. Here, the true positive rates are plotted against false positive rates as in the case of both repositories. It is important to note that precision-recall curves PR-AUC can only be calculated for types of neural networks, generally classifiers, which output a probability also called confidence (both AUC and PR-AUC goes under the principle of obtaining true or false positives). For that reason, we develop a table **Table 8,** in which we can appreciate the main differences between them.

Table 8: Comparison of different metrics between the paper's models. The most relevant parameters are compared: AUC, database used, if they use or not BI-RADS Pretrain, the PR-AUC and the type of GPU used (its computational power)

Characteristic	***Repository 1*** ***Model:*** ***GMIC-ResNet-18-ensemble***	***Repository 2*** ***Model: 'view-wise' model***
AUC	93%	89,5%
DataBase	NYU Breast Cancer Screening Dataset (229,426 high-resolution screening) mammograms	NYU Breast Cancer Screening Dataset (229,426 digital screening mammography exams)
BI-RADS Pretrain	Yes	Yes
PR-AUC	0.39	0.318
Data Center GPU	NVIDIA Tesla V100 GPU.	NVIDIA Tesla V100 GPU.

Now, talking about hybrid models, in both papers, the authors propose their version of it; in the first one, in addition to the primary model, they experiment with hybrid models that combine predictions from both GMIC and each of the radiologists separately. At radiologists' sensitivity (62.1%), the hybrid models achieve an average specificity of 91.9% improving radiologists' average specificity by 6.3%. The given name is "Human-machine Hybrid," made to demonstrate the clinical potential of GMIC. Its predictions are a linear combination of the reader and the model. They compute the AUC and PRAUC of the hybrid models by setting λ = 0.5. Seeing that λ = 0.5 is not the optimal value for hybrid models [11,12].

On the other hand, the performance obtained by retroactively fine-tuning λ on the reader study is not transferable to realistic clinical settings. Therefore, λ = 0.5 is the most natural way of aggregating two sets of predictions without prior knowledge of their quality. For each of the hybrid models, it also calculated its specificity at the average radiologists' sensitivity (62.1%). The 14 hybrid models achieve an average specificity of 91.5%, which is higher than the average radiologists' specificity (85.2%). These results indicate that the model captures different aspects of the task than radiologists and can be used as a tool to assist in interpreting breast cancer screening exams [11].

The second paper shows a hybrid model in which the averaging of the probability of malignancy predicted by a radiologist with a prediction of our neural network is more accurate than either separately. They also evaluated the accuracy of a human-machine hybrid, whose predictions are a linear combination of predictions of a radiologist and the model–that is, hybrids between each reader and the model achieved an average AUC of 0.891 and an average PRAUC of 0.431. These results suggest that the model can be used as a tool to assist radiologists in reading breast cancer screening exams and that it captured different aspects of the task compared to experienced breast radiologists. However, in our study, a hybrid model including both a neural network and expert radiologists outperformed either individually, suggesting that the use of such a model could improve radiologist sensitivity for breast cancer detection [12].

4.4.2. Comparisons between repositories and model developed

First, is important to say that it was previous mentioned, the metrics that we obtain in the last and final model based on Mask R-CNN has not the same metrics than the ones in both repositories, for that reason the only comparison that could be done with the same parameters is with the first experiment of the classifier based on ResNet 18 **Table 9**.

Table 9: Comparison between the two repositories in three main characteristics AUC, database, and GPUs processing power.

Characteristic	***Repository 1 Model: GMIC-ResNet-18-ensemble***	***Repository 2 Model: 'view-wise' model***	***Classifier Based in ResNet 18***
AUC	93%	89.5%	43.02%
DataBase	NYU Breast Cancer Screening Dataset (229,426 high-resolution screening) mammograms	NYU Breast Cancer Screening Dataset (229,426 digital screening mammography exams)	CBIS-DDSM (2620 digital screening mammography exams)
Data Center GPU	NVIDIA Tesla V100 GPU.	NVIDIA Tesla V100 GPU.	Tesla V100 4 GPUs.

CHAPTER V: Conclusion and Recommendations

5.1. Conclusions

We can say that CNNs are a handy tool for detection and classification tasks and for developing great performance accuracy in the cases of repository 1 and repository 2. Indeed, the main objectives of the analyzed papers are similar but not equal. The first article bases its work in finding malignant and benign lesions while the second one tries to predict the presence of cancer. Besides, the second paper mentions that a benign tumor recognition is also made as a secondary task. Therefore, models developed in both works can be comparable, having an accuracy of almost 90% in recognizing the presence of cancer (by lesions, or abnormalities in mammograms) in breast tissues. This is possible thanks to the modification and adaptation of a natural object CNN architecture to achieve this percentage, showing that these modifications are necessary to be clinically viable in the medical field. The approaches in detecting cancer improve continuously, and a promising future for this type of Artificial Intelligence research is being seen in the near future.

One of the most important things that we must consider in this work is the fact that a mammography as is mentioned at the beginning is an X-Ray image, that do not have the best available resolution to differentiate the biological tissues that appears in the image. In addition to the problem of overlapping that one tissue or even the tumors can have making the abnormality detection more difficult even. Despite efforts to get 4 images per mammography exam, the information given from them is not enough to give a high accuracy diagnosis, especially in the disease early stages, the shying of this medical imaging technique does not allow it. For that reason, the approaches presented in this work, including the one develops from ourselves may not have clinical validity, for the lack of clarity of medical images used. If is true that mammography as is seen in this thesis project, is not the best option to a precise diagnosis the fact is that is the less consuming, cheaper and available method for breast cancer detection.

The hardware is an important limitation that has to be taken account to the develop of this kind of projects. Using a computational potent machine, the resize, and image preprocessing maybe could not be necessary and the network could run fast with more images per GPU using at least four of them. Another important thing to consider is that normal computers and neural networks works with RGB scale, making that images in grey scale lose a lot of information because the machine can not distinguish all the details in them. In our case, black and white color layers had to be replicated 3 times to cover the color scale that is not ideal. Have a scale of its own that only covers black and gray white to respect the shades of medical images.

Based in the obtained results and comparative tables we can conclude that the model based only in ResNet 18 without any modifications in the architecture has an AUC considerably minor than the other two models achieving only a 42.03% of accuracy in diagnosis of the three established categories Benign, Benign without call-back, or Malignant. Thus, confirming the theory that a classifier with the CNN architecture for natural objects is not very much useful for medical image classification. In the final model using as a bade Mask R-CNN software, the metrics could not be comparable with the ones in the papers, because the obtained results show the Loss function against the AUC presented the other works.

The main conclusion about the model after four experimentation runs with different hyper-parameters, is that Experiment III and fourth are the best. This could be explained for many reasons, being the most important the use of data augmentation in each one of them, making a big difference as compared with the other two. Besides, the use of 4 images per GPU helps in the time of running, and the loss value is very low. Finally, between both models, I will select the fourth one because it achieves a very low loss value (0.7), and its performance is high with only 50 epochs and half of the training time compared with Experiment III. It is essential to mention that our model was derived from all-natural object weight training, not from specific weights (trained from scratches in medical images). Then, training only in mammography could make a big difference the final results of the model. For all the reasons exposed the model could be improved in significant ways, so the current model is not viable for clinical purposes yet.

It is almost essential to mention that with the current information available until now. It is impossible to know if machine learning is more sensitive and accurate when the tumor development is at an early stage, especially for the lack of existent data images in this stage of the illness. Another question that arises is: Can other relevant characteristics of the pictures or even metastasis be detected with this kind of software?. Until now, the answer is that the primary function of the different software presented in this work is trying to identify the existence or not of cancer in breast tissue, so the network is not trained to search for anything else. As is mentioned in the introduction of this work, ideally, a diagnosis method should detect a tumor the earliest possible and not just the category of tumor present with high accuracy, but given the complexity of the duty of finding a dataset with this kind of images, is not possible for the moment.

5.2. Future Works

As is visible, all the models work with the premise of false positives. But what happens with the false negatives? It means that a person has a malignant tumor, and the exams say the opposite. It could be considered worst even. For this reason, future investigations could involve focuses on the false-negative scenario and work to develop solutions against this problem.

The exciting thing about the Mask R-CNN is that its filter is smart, which means that it can find the objects automatically, which becomes even more useful if you want to apply it to videos rather than a single image; making this model applicable in other types of images, maybe with significant results as magnetic resonance ones.

Finally it is important to note that in fact it is not clear if the death rate due to this illness in Ecuador, which seems high compared to developed countries, is due to diagnosis challenges or lack of treatments available. So, we suggest as an additional step for the future development of this work to make a research about this point.

5.5.2. Scientific Divulgation

Accepted in MICRADS ´21: Multidisciplinary International Conference of Research Applied to Defense; International Congress. With the work entitle: "Comparison between Two Novel Approaches in Automatic Breast Cancer Detection and Diagnosis and its Contribution in Military Defense."

Accepted Manuscript in: Proceedings by Springer, a book of its Smart Innovation, Systems and Technologies (SIST) series. With the work entitle: "Comparison between Two Novel Approaches in Automatic Breast Cancer Detection and Diagnosis and its Contribution in Military Defense."

References

1. Promotion NC for CDP and H. Cáncer de mama : Lo que debe saber. Pdf [Internet]. 2010;1(800):1–1. Available from: http://www.cdc.gov/spanish/cancer/breast/pdf/BreastCancerFactSheet.pdf
2. Hejmadi M.: Introduction to Cancer Biology. Second Edition. 2010;
3. World Health Organisation: Cancer death rate (2021), https://www.who.int/es,Last accessed on 2021-01-16
4. Curado MP, Edwards B, Shin HR, Ferlay J, Storm H. Cancer Incidence in Five Continents. 2007;XI(November):837. Available from: http://books.google.co.uk/booksid=KU_FMQEACAAJ%5Cnhttp://www.iarc.fr/en/publications/pdfs-online/epi/sp160/CI5vol9.pdf
5. Cueva P, Yepez J. Epidemiología del cancer en Quito. SOCIEDAD DE LUCHA CONTRA EL CÁNCER SOLCA NÚCLEO DE QUITO. 2019. 234 p.
6. Flaherty, D.C., Bawa, R., Burton, C., Goldfarb, M.: Breast Cancer in Male Adolescents and Young Adults. Annals of Surgical Oncology 24(1), 84–90 (2017)
7. Fentiman, I.S.: Male breast cancer is not congruent with the female disease. Critical Reviews in Oncology/Hematology 101, 119–124 (2016), http://dx.doi.org/10.1016/ j.critrevonc.2016.02.017
8. Serdy, K.M., Leone, J.P., Dabbs, D.J., Bhargava, R.: Male breast cancer: A singleinstitution clinicopathologic and immunohistochemical study. American Journal of Clinical Pathology 147(1), 110–119 (2017)
9. Abdelwahab Yousef, A.J.: Male Breast Cancer: Epidemiology and Risk Factors. Seminars in Oncology 44(4), 267–272 (2017), https://doi.org/10.1053/j. Seminoncol.2017.11.002
10. Bradburne, C., Lewis, J.A.: Personalizing Environmental Health. Journal of Occupational and Environmental Medicine 59(11), e209–e214 (2017)
11. Shen Y, Wu N, Phang J, Park J, Liu K, Tyagi S, et al. An interpretable classifier for high-resolution breast cancer screening images utilizing weakly supervised localization. arXiv. 2020;
12. Wu N, Phang J, Park J, Shen Y, Huang Z, Zorin M, et al. Deep Neural Networks Improve Radiologists' Performance in Breast Cancer Screening. IEEE Trans Med Imaging. 2020;39(4):1184–94.
13. Hutchison, D.: and Data Labeling pp. 197–205 (2016)
14. Ragab, D.A., Sharkas, M., Marshall, S., Ren, J.: Breast cancer detection using deep convolutional neural networks and support vector machines. PeerJ 2019(1), 1–23 (2019)
15. Amdani, D.S.Y.: Breast Cancer Detection using Deep Learning. International Journal for Research in Applied Science and Engineering Technology 7(5), 2672–2674 (2019)
16. Akselrod-Ballin, A. Karlinsky, L. Alpert, S. Hasoul, S. et al. A Region Based Convolutional Network for Tumor Detection and Classification in Breast Mammography. IBM Research. 2016;
17. Geras KJ, Wolfson S, Shen Y, Wu N, Gene Kim S, Kim E, et al. High-resolution breast cancer screening with multi-view deep convolutional neural networks. arXiv. 2017;1–9.
18. Gagnon J, Lévesque E, Borduas the CAC on BCS and P (F., Chiquette J, Diorio C, Duchesne N, et al. Current oncology. Curr Oncol [Internet]. 2016;23(6):615–25. Available from: https://current-oncology.com/index.php/oncology/article/view/2961/2330

19. Fentiman, I.S.: Male breast cancer is not congruent with the female disease. Critical Reviews in Oncology/Hematology 101, 119–124 (2016), http://dx.doi.org/10.1016/ j.critrevonc.2016.02.017
20. Gucalp, A., Traina, T.A., Eisner, J.R., Parker, J.S., Selitsky, S.R., Park, B.H., Elias, A.D., Baskin-Bey, E.S., Cardoso, F.: Male breast cancer: a disease distinct from female breast cancer. Breast Cancer Research and Treatment 173(1), 37–48 (2019), http://dx.doi.org/10.1007/s10549-018-4921-9
21. Cotto, D.J.R., Feijoo, I.L.J., Briones, D.R.Q.: Mortalidad por Cáncer de Mama en Guayaquil. Per´ıodo 2009-2018 pp. 2018–2019 (2019)
22. Ministerio de Salud Pública del Ecuador.: Cifras de Ecuador – Cáncer de Mama. From: https://www.salud.gob.ec/cifras-de-ecuador-cancer-de-mama/
23. REGISTRO NACIONAL DE TUMORES / NATIONAL CANCER REGISTRY "Fabián Corral Cordero". 2019;
24. Gucalp, A., Traina, T.A., Eisner, J.R., Parker, J.S., Selitsky, S.R., Park, B.H., Elias, A.D., Baskin-Bey, E.S., Cardoso, F.: Male breast cancer: a disease distinct from female breast cancer. Breast Cancer Research and Treatment 173(1), 37–48 (2019), http://dx.doi.org/10.1007/s10549-018-4921-9
25. National Cancer Institute (NCI).: About This Booklet. U.S. DEPARTMENT OF HEALTH AND HUMAN SERVICES National Institutes of Health. 2012.
26. American Cancer Society. About Breast Cancer. Breast Cancer Facts Fig [Internet]. 2017;1–19. Available from: https://www.cancer.org/content/dam/CRC/PDF/Public/8577.00.pdf%0Ahttp://www.breastcancer.org/symptoms/understand_bc/what_is_bc
27. Xiong Y, B B Du, Yan P. Reinforced Transformer for Medical [Internet]. Vol. 1. 2019. 673–680 p. Available from: http://dx.doi.org/10.1007/978-3-030-32692-0_68
28. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 07-12-June, 1–9 (2015)
29. Lisle, D. Imaging for Students. Fourth edition. LECTURE 1 Introduction to Medical Imaging. 2012;
30. Linver MN. 4-19 Mammographic Density and the Risk and Detection of Breast Cancer. Breast Dis. 2008;18(4):364–5.
31. Kelly KM, Dean J, Comulada WS, Lee SJ. Breast cancer detection using automated whole breast ultrasound and mammography in radiographically dense breasts. Eur Radiol. 2010;20(3):734–42.
32. Lee RS, Gimenez F, Hoogi A, Miyake KK, Gorovoy M, Rubin DL. Data Descriptor: A curated mammography data set for use in computer-aided detection and diagnosis research. Sci Data. 2017;4:1–9.
33. CBIS-DDSM - The Cancer Imaging Archive (TCIA) Public Access - Cancer Imaging Archive Wiki
34. Alsedrah MK. Running Head : ARTIFICIAL INTELLIGENT Artificial Intelligence Advanced Analysis and Design : CNIT 380 Instructors : Dr . Hiba Tabbarah & Mr . Abdullah Abdulghafar Semester : Fall 2017 Section : U1 Mariam Khaled AlSedrah. Researchgate. 2018;(December 2017).
35. Mijwil, Maad.: History of Artificial Intelligence. 3. 1-8. (2015);
36. Liu, Qiong & Wu, Ying.: Supervised Learning. 10.1007/978-1-4419-1428-6_451. (2012);

37. Kotsiantis S. B. Supervised Machine Learning: A Review of Classification Techniques. Department of Computer Science and Technology University of Peloponnese, Greece. 2007.
38. Cichy RM, Kaiser D. Deep Neural Networks as Scientific Models. Trends Cogn Sci. 2019;23(4):305–17.
39. Kriesel, D.: A Brief Introduction to Neural Networks. 2005;
40. Steen M, Downe S, Bamford N, Edozien L. DenseNet:Densely Connected Convolutional Networks arXiv:1608.06993v5. Arxiv. 2018;28(4):362–71.
41. Limonova E, Alfonso D, Nikolaev D, Arlazarov V V. ResNet-like Architecture with Low Hardware Requirements. 2020; Available from: http://arxiv.org/abs/2009.07190
42. Alan V. Oppenheim, Alan S. Willsky, Syed Hamid Nawab.: Signals and systems. Prentice Hall. 1997;
43. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit. 2015;07-12-June:1–9.
44. Rawat, W. Wang, Z. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. Massachusetts Institute of Technology. 2017;
45. Grupo PAS – Universidad de Deusto. Estándar y Protocolo de Imágenes Medicas DICOM. :1–17. Available from: http://www.sicec.unam.mx/app/webroot/files/archivos_portal/archSISEC254505.pdf
46. Guibas, J. Virdi, T. Li, P.: Synthetic Medical Images from Dual Generative Adversarial Networks. 31st Conference on Neural Information Processing Systems. 2017;
47. He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. IEEE Trans Pattern Anal Mach Intell. 2020;42(2):386–97.
48. Lecun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436–44.
49. Lefohn A. GPU memory model overview. ACM SIGGRAPH 2005 Courses, SIGGRAPH 2005. 2005;
50. Ragab DA, Sharkas M, Marshall S, Ren J. Breast cancer detection using deep convolutional neural networks and support vector machines. PeerJ. 2019;2019(1):1–23.
51. Sneyers, J. Wuille, P.: FLIF : FREE LOSSLESS IMAGE FORMAT BASED ON MANIAC COMPRESSION Jon Sneyers Pieter Wuille Blockstream TL. 2000;
52. Ruiz, P. ResNets. 2018;
53. mritopng/README.md at master · danishm/mritopng (github.com)
54. Morra L, Piano L, Lamberti F, Tommasi T. Bridging the gap between natural and medical images through deep colorization. arXiv. 2020;
55. Bergstra, J. Bardenet, R. Bengio, Y. Kégl, B.: Algorithms for Hyper-Parameter Optimization. 2014;
56. https://ngc.nvidia.com/catalog/containers/nvidia:tlt-streamanalytics (poner esto como referencia) (catalogo de containers como parte de los recursos de NVIDIA CLOUD de visual computing).
57. The KITTI Vision Benchmark Suite (cvlibs.net)
58. NVIDIA TRANSFER LEARNING TOOLKIT FOR INTELLIGENT VIDEO ANALYTICS. 2020;
59. https:// scikit-image: Image processing in Python — scikit-image (scikit-image.org)
60. TLT DetectNet V2 Detection | NVIDIA NGC
61. https://realpython.com/pysimplegui-python/

62. Splash of Color: Instance Segmentation with Mask R-CNN and TensorFlow | by Waleed Abdulla | Matterport Engineering Techblog
63. Tsung-Yi, L. Piotr, D. Ross, G. Kaiming, H. et al.: Feature Pyramid Networks for Object Detection. Cornell University. 2017;
64. He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. IEEE Trans Pattern Anal Mach Intell. 2020;42(2):386–97.
65. NVIDIA Transfer Learning Toolkit | NVIDIA Developer
66. Zeiler, D. M. krishnan, D. Taylor, W. G. Fergus, R.: Deconvolutional Networks. Dept. Of computer science, New York University. 2008;
67. Browniee, J.: Deep Learning with Python. Machine learning Mastery. 2018;
68. Python OpenCV – cv2.flip() method - GeeksforGeeks
69. Murphy, P. K.: Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series) . 2012;
70. Liu, J. Li, P.: A Mask R-CNN Model with Improved Region Proposal Network for Medical Ultrasound Image. International Conference on Intelligent Computing. 2018;

Attachments

Repository

https://github.com/ZosoV/Mammography_Mask_RCNN.git

Snipped code (the most relevant part KITTI code)

```
    img_0 = cv2.imread(join(DATA_DIR,base_name))
    print("Base image shape: ", img_0.shape)
    mask_0 = cv2.imread(join(DATA_DIR,mask_name))
    print("Mask shape: ", mask_0.shape)
    crop_0 = cv2.imread(join(DATA_DIR, crop_name))
    print("Crop shape: ", crop_0.shape)

    #Get the bounding boxes
    lbl_0 = label(mask_0)
    props = regionprops(lbl_0)

    img_1 = img_0.copy()

    xmin, ymin, xmax, ymax = (-1,  -1, -1, -1)
    for prop in props:
      print('Found bbox', prop.bbox)
      xmin, ymin, xmax, ymax = (prop.bbox[0], prop.bbox[1], prop.bbox[3], prop.bbox[4])
      cv2.rectangle(img = img_1,
                    pt1 = (ymax, xmin), #top-left
                    pt2 = (ymin, xmax), #bottom-right
                    color = (0, 255, 0),
                    thickness = 7)
```

Detector configuration

```
random_seed: 42
dataset_config {
  data_sources {
    tfrecords_path: "/workspace/tlt-experiments/data/tfrecords/kitti_trainval/*"
    image_directory_path: "/workspace/tlt-experiments/data/training"
  }
  image_extension: "png"
  target_class_mapping {
    key: "BENIGN"
    value: "BENIGN"
  }
  target_class_mapping {
    key: "BENIGN_WITHOUT_CALL_BACK"
    value: "BENIGN_WITHOUT_CALL_BACK"
  }
  target_class_mapping {
    key: "MALIGNANT"
    value: "MALIGNANT"
  }

  validation_fold: 0
}
augmentation_config {
  preprocessing {
    output_image_width: 1248
    output_image_height: 384
    min_bbox_width: 1.0
    min_bbox_height: 1.0
    output_image_channel: 3
  }
  spatial_augmentation {
    hflip_probability: 0.5
    zoom_min: 1.0
    zoom_max: 1.0
    translate_max_x: 8.0
    translate_max_y: 8.0
  }
  color_augmentation {
    hue_rotation_max: 25.0
    saturation_shift_max: 0.20000000298
    contrast_scale_max: 0.10000000149
    contrast_center: 0.5
  }
}
postprocessing_config {
  target_class_config {
    key: "BENIGN"
    value {
      clustering_config {
        coverage_threshold: 0.00499999988824
        dbscan_eps: 0.20000000298
        dbscan_min_samples: 0.0500000007451
        minimum_bounding_box_height: 20
      }
    }
  }
  target_class_config {
    key: "BENIGN_WITHOUT_CALL_BACK"
    value {
      clustering_config {
        coverage_threshold: 0.00499999988824
        dbscan_eps: 0.15000000596
        dbscan_min_samples: 0.0500000007451
        minimum_bounding_box_height: 20
      }
    }
  }
  target_class_config {
    key: "MALIGNANT"
    value {
      clustering_config {
        coverage_threshold: 0.00749999983236
        dbscan_eps: 0.230000004172
        dbscan_min_samples: 0.0500000007451
        minimum_bounding_box_height: 20
      }
    }
  }
}
model_config {
  pretrained_model_file:
"/workspace/tlt-experiments/detectnet_v2/pretrained_resnet18/tlt_pretrained_detectnet_v2_vresnet18/resnet1
8.hdf5"
  num_layers: 18
  use_batch_norm: true
  objective_set {
    bbox {
      scale: 35.0
      offset: 0.5
    }
    cov {
    }
  }
  training_precision {
    backend_floatx: FLOAT32
  }
  arch: "resnet"
}
evaluation_config {
  validation_period_during_training: 10
  first_validation_epoch: 30
  minimum_detection_ground_truth_overlap {
    key: "BENIGN"
    value: 0.699999988079
  }
  minimum_detection_ground_truth_overlap {
    key: "BENIGN_WITHOUT_CALL_BACK"
    value: 0.5
  }
  minimum_detection_ground_truth_overlap {
    key: "MALIGNANT"
    value: 0.5
  }
  evaluation_box_config {
    key: "BENIGN"
    value {
      minimum_height: 20
      maximum_height: 9999
      minimum_width: 10
      maximum_width: 9999
    }
  }
  evaluation_box_config {
    key: "BENIGN_WITHOUT_CALL_BACK"
    value {
      minimum_height: 20
      maximum_height: 9999
      minimum_width: 10
      maximum_width: 9999
    }
  }
  evaluation_box_config {
    key: "MALIGNANT"
    value {
      minimum_height: 20
      maximum_height: 9999
      minimum_width: 10
      maximum_width: 9999
    }
  }
  average_precision_mode: INTEGRATE
}
cost_function_config {
  target_classes {
    name: "BENIGN"
    class_weight: 1.0
    coverage_foreground_weight: 0.0500000007451
    objectives {
      name: "cov"
      initial_weight: 1.0
      weight_target: 1.0
    }
    objectives {
      name: "bbox"
      initial_weight: 10.0
      weight_target: 10.0
    }
  }
  target_classes {
    name: "BENIGN_WITHOUT_CALL_BACK"
    class_weight: 8.0
    coverage_foreground_weight: 0.0500000007451
    objectives {
      name: "cov"
      initial_weight: 1.0
      weight_target: 1.0
    }
    objectives {
      name: "bbox"
      initial_weight: 10.0
      weight_target: 1.0
    }
  }
  target_classes {
    name: "MALIGNANT"
    class_weight: 4.0
    coverage_foreground_weight: 0.0500000007451
    objectives {
      name: "cov"
      initial_weight: 1.0
      weight_target: 1.0
    }
    objectives {
      name: "bbox"
      initial_weight: 10.0
      weight_target: 10.0
    }
  }
  enable_autoweighting: true
  max_objective_weight: 0.999899983406
  min_objective_weight: 9.99999974738e-05
}
training_config {
  batch_size_per_gpu: 4
  num_epochs: 120
  learning_rate {
    soft_start_annealing_schedule {
      min_learning_rate: 5e-06
      max_learning_rate: 5e-04
      soft_start: 0.10000000149
      annealing: 0.699999988079
    }
  }
  regularizer {
    type: L1
    weight: 3.00000002618e-09
  }
  optimizer {
    adam {
      epsilon: 9.99999993923e-09
      beta1: 0.899999976158
      beta2: 0.999000012875
    }
  }
  cost_scaling {
    initial_exponent: 20.0
    increment: 0.005
    decrement: 1.0
  }
  checkpoint_interval: 10
}
bbox_rasterizer_config {
  target_class_config {
    key: "BENIGN"
    value {
      cov_center_x: 0.5
      cov_center_y: 0.5
      cov_radius_x: 0.40000000596
      cov_radius_y: 0.40000000596
      bbox_min_radius: 1.0
    }
  }
  target_class_config {
    key: "BENIGN_WITHOUT_CALL_BACK"
    value {
      cov_center_x: 0.5
      cov_center_y: 0.5
      cov_radius_x: 1.0
      cov_radius_y: 1.0
      bbox_min_radius: 1.0
    }
  }
  target_class_config {
    key: "MALIGNANT"
    value {
      cov_center_x: 0.5
      cov_center_y: 0.5
      cov_radius_x: 1.0
      cov_radius_y: 1.0
      bbox_min_radius: 1.0
    }
  }
  deadzone_radius: 0.400000154972
}
```