NATIONAL TECHNICAL UNIVERSITY OF ATHENS
School of Rural, Surveying and Geoinformatics Engineering

Investigating the potentials of AI on segmenting tumors depicted on digital mammograms

by

Ioannis Panagiotopoulos

Supervisors:

Nikolaos Doulamis, Anastasios Doulamis, Vasileios Vescoukis

Athens, March 2024

ACKNOWLEDGMENTS

This thesis would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the presentation and completion of this study. It's a pleasure to thank those who made it a possibility.

First and foremost, I would like to express my deep and sincere gratitude to my Professors Nikolaos Doulamis, Anastasios Doulamis and Vasileios Vescoukis for giving me the opportunity to select this very interesting topic, as well as being always there for me whenever I needed their assistance. Their guidance throughout the project has been invaluable.

Words cannot express my gratitude to my supervisor Ioannis Tzortzis, whose guidance, support, and outstanding feedback helped me in all the time of research and writing this thesis. His enthusiasm, deep knowledge of the topic and integral view on research and his mission for providing ‘only high-quality work and not less’, have made a profound impression on me, and inspired me to give my best self.

Last but not least, I would like to thank my family as a whole, whose constant encouragement fueled my perseverance during the completion of this research. My parents’ and brother’s unconditional love, care and tolerance made the hardship of writing the thesis worthwhile. Without their constant support, I do not think that I would have made it.

ΠΕΡΙΛΗΨΗ

Ο καρκίνος του μαστού αποτελεί σοβαρή απειλή για τις γυναίκες σε όλο τον κόσμο. Είναι ένας από τους πιο συχνά διαγνωσθέντες τύπους καρκίνου, με ποσοστό θνησιμότητας περίπου μία στις έξι γυναίκες, γεγονός που τον καθιστά μία από τις κύριες αιτίες θανάτου για τις γυναίκες παγκοσμίως. Μια πολύ αποτελεσματική ιατρική απεικονιστική τεχνική για την ανίχνευση του καρκίνου του μαστού, ιδιαίτερα σε πρώιμο στάδιο, στο οποίο είναι πιο θεραπεύσιμος, είναι η μαστογραφία. Η μαστογραφία είναι μια φθηνή και ασφαλής μέθοδος, που γίνεται από ακτινολόγους με τη χρήση ειδικών ακτινών, προκειμένου να οπτικοποιηθεί ο ιστός του μαστού. Χρησιμοποιείται για την υποστήριξη της έγκαιρης θεραπείας ασθενών με καρκίνο του μαστού και για την αύξηση του ποσοστού επιβίωσής τους, ενώ στοχεύει στην αποφυγή μιας ανεπιθύμητης επιθετικής λύσης, όπως η μαστεκτομή. Τις τελευταίες δύο δεκαετίες έχουν αναπτυχθεί διάφορα συστήματα ανίχνευσης με τη βοήθεια υπολογιστή για να βοηθήσουν τους ειδικούς ιατρούς να ανιχνεύσουν τις διάφορες ανωμαλίες του μαστού, όπως μάζες, ασβεστώσεις, αρχιτεκτονική παραμόρφωση του ιστού και ασυμμετρίες, στις μαστογραφίες. Σε αυτή την εργασία, προτείνεται ένα συνελικτικό νευρωνικό δίκτυο για την κατάτμηση του όγκου του μαστού σε ψηφιακές μαστογραφίες. Ο συγκεκριμένος τύπος συνελικτικού νευρωνικού δικτύου που χρησιμοποιείται είναι το U-net, με τις διαστάσεις της αρχικής εικόνας (εικόνα εισόδου) να είναι 256x256 pixel. Πιο αναλυτικά, το σύνολο δεδομένων INbreast, το οποίο αποτελείται από 410 μαστογραφίες και τις αντίστοιχες μάσκες τους, χρησιμοποιήθηκε για την εκπαίδευση και αξιολόγηση του μοντέλου, ενώ η μελέτη χωρίζεται σε δύο διακριτές προσεγγίσεις, στις οποίες πραγματοποιήθηκαν συνολικά οκτώ πειράματα, δηλαδή τέσσερα σε κάθε προσέγγιση. Πριν από τη διάκριση των δύο προσεγγίσεων, προηγείται η διαδικασία προεπεξεργασίας, η οποία περιλαμβάνει την περικοπή, την αλλαγή μεγέθους και την κανονικοποίηση των εικόνων. Όσον αφορά τις δύο προσεγγίσεις, η πρώτη αφορά το μοντέλο που εκπαιδεύεται με το «απλό» σύνολο δεδομένων INbreast, ενώ η δεύτερη εκπαιδεύεται με ένα επαυξημένο σύνολο δεδομένων, έξι φορές μεγαλύτερο από το αρχικό, το οποίο αντιστοιχεί σε συνολικά 2460 μαστογραφίες και τις αντίστοιχες μάσκες τους. Οι τεχνικές αύξησης εικόνων που επιλέχθηκαν περιλαμβάνουν την εξίσωση ιστογράμματος, τη διόρθωση γάμμα και την περιστροφή 180 μοιρών. Οι μετρήσεις αξιολόγησης που επιλέχθηκαν να χρησιμοποιηθούν για τον υπολογισμό της απόδοσης του μοντέλου είναι η βαθμολογία F1 και η τιμή απώλειας. Και στις δύο προσεγγίσεις, το σύνολο δεδομένων χωρίζεται σε τρία διαφορετικά σύνολα, το σύνολο εκπαίδευσης, το σύνολο επικύρωσης και το σύνολο δοκιμών, με 70%, 20%, 10%, αντίστοιχα. Επιπλέον, αναφέρεται πως ο βελτιστοποιητής που επιλέχθηκε για το μοντέλο είναι ο Adam. Όπως σημειώθηκε προηγουμένως, σε κάθε προσέγγιση έχουν γίνει τέσσερα πειράματα, τα οποία διαφέρουν ως προς το συνδυασμό του ρυθμού μάθησης και των εποχών. Πιο συγκεκριμένα, για κάθε προσέγγιση πραγματοποιήθηκαν δύο πειράματα 50 εποχών: ένα με μικρό ποσοστό μάθησης (lr = 0,001) και ένα με μεγάλο ποσοστό μάθησης (lr = 0,01) και δύο πειράματα 100

εποχών: ένα με μικρό ποσοστό μάθησης (lr = 0,001) και ένα με μεγάλο ποσοστό μάθησης (lr = 0,01). Όσον αφορά τα αποτελέσματα της μελέτης, σημειώνεται ότι η δεύτερη προσέγγιση αποδείχθηκε σημαντικά καλύτερη από την πρώτη. Στην πραγματικότητα, τα μη ικανοποιητικά αποτελέσματα της πρώτης προσέγγισης οδηγούν στην ιδέα της εφαρμογής των τεχνικών αύξησης δεδομένων και άρα, στη δεύτερη προσέγγιση. Πιο αναλυτικά, η καλύτερη επίδοση στην πρώτη προσέγγιση επιτεύχθηκε από το τέταρτο πείραμα, αυτό με τις 100 εποχές και το μεγάλο ποσοστό μάθησης, το οποίο είχε βαθμολογία F1 ίση με 0,64 για το σετ ελέγχου, ενώ τα υπόλοιπα πειράματα δεν κατάφεραν να «μάθουν» από τα δεδομένα, με αποτέλεσμα η βαθμολογία F1 και να είναι μικρότερη από 0,60 και να έχουν μεγαλύτερη τιμή σφάλματος. Μετά από βαθιά ανάλυση αυτών των αποτελεσμάτων, το συμπέρασμα ήταν ότι το μέγεθος δεδομένων δεν επαρκεί για την αποτελεσματική εκπαίδευση του μοντέλου, καθώς σε τρία από τα τέσσερα πειράματα το μοντέλο αντιμετώπισε υπερεκπαίδευση, ένα κλασικό πρόβλημα που συναντάται στα μικρά σύνολα δεδομένων. Έτσι, επιλέχθηκε να ελεγχθεί η απόδοση του μοντέλου σε ένα σύνολο δεδομένων μεγαλύτερου μεγέθους. Στη δεύτερη προσέγγιση, το πιο ακριβές πείραμα ήταν το τρίτο, δηλαδή αυτό με τις 100 εποχές και το μικρό ποσοστό μάθησης, το οποίο πέτυχε βαθμολογία F1 ίση με 0,81 στο σετ ελέγχου. Τα τρία υπόλοιπα πειράματα της δεύτερης προσέγγισης πέρασαν το 0,60 στη βαθμολογία F1 του σετ δοκιμών, κάτι το οποίο τα τρία πρώτα πειράματα της πρώτης προσέγγισης απέτυχαν να κάνουν. Η βαθμολογία F1 τους ήταν 0,71, 0,64 και 0,77 αντίστοιχα. Ως εκ τούτου, με βάση τα παραπάνω αποτελέσματα αποδεικνύεται ότι η επιλογή των μεθόδων αύξησης είναι επιτυχής και ότι το μοντέλο χρειάζεται πράγματι μεγαλύτερο αριθμό δεδομένων για να εκπαιδευτεί αποτελεσματικά. Επιπλέον, φαίνεται ότι η επιλογή του ρυθμού εκμάθησης εξαρτάται κυρίως από το σύνολο δεδομένων, αλλά και από τον καθορισμένο αριθμό εποχών για την εκπαίδευση του μοντέλου. Στην πρώτη προσέγγιση τα δύο ποσοστά μάθησης δεν ήταν συνεπή στις δύο διαφορετικές επιλογές των εποχών, αφού στα πειράματα των 50 εποχών το μικρό ποσοστό μάθησης ήταν πιο επιτυχημένο, ενώ στα πειράματα των 100 εποχών το μεγάλο ποσοστό μάθησης πέτυχε καλύτερα αποτελέσματα. Αντίθετα, στη δεύτερη προσέγγιση, τόσο στα πειράματα των 50 εποχών όσο και σε αυτά των 100, η καλύτερη απόδοση του μοντέλου επιτεύχθηκε με το μικρό ποσοστό μάθησης.

ABSTRACT

Breast cancer is a serious threat to women worldwide. It is one of the most commonly diagnosed types of cancer, with a mortality rate of around one in six women, making it one of the leading causes of death for women globally. A very effective medical imaging technique for breast cancer detection, especially at an early stage, in which is more treatable, is mammography. Mammography is a cheap and safe method, performed by radiologists with the use of a dedicated X-ray, in order to visualize the breast tissue. It is used to support early treatment for breast cancer patients and to increase their survival rate, while aiming to avoid an unwanted aggressive solution, such as mastectomy. In the last two decades, various Computer-aided-detection (CAD) systems have been developed to help medical experts detect breast abnormalities, including masses, calcifications, architectural distortion of the tissue, and asymmetries in mammograms. In this work, a Convolutional Neural Network (CNN) for breast tumor segmentation on digital mammograms is proposed. The specific type of CNN used is the UNET, with the starting (input) image dimension being 256x256 pixels. More analytically, the INbreast dataset, which consists of 410 mammograms and their corresponding masks, was utilized to train, and evaluate the model, while the study is divided into two distinct approaches, in which were conducted eight experiments in total, meaning four in each approach. Former to the distinction between the two approaches, is preceded the preprocessing procedure, which contains the cropping, resizing, and normalization of the images. Concerning the two approaches, the first is about the model being trained with the “simple” INbreast dataset, while the second is trained with an augmented dataset, six times bigger than the initial one, resulting in a total of 2460 mammograms and their corresponding masks. The augmentation techniques selected include the histogram equalization, the gamma correction, and the 180-degree rotation of the images. The selected evaluation metrics that were used to calculate the performance of the model are the F1 score and the loss value. In both approaches, the dataset is split into three different sets, the training set, the validation set, and the testing set, with 70%, 20%, 10%, respectively. The optimizer selected for the fine-tuning of the model is the Adam. As previously noted, in each approach there have been done four experiments, which differ in the combination of learning rate and epochs. More specifically, for each approach there were conducted two 50-epoch experiments: one with a small learning rate (lr = 0.001) and one with a big learning rate (lr = 0.01), and two 100-epoch experiments: one with a small learning rate (lr = 0.001) and one with a big learning rate (lr = 0.01). Concerning the results of the study, it is noted that the second approach proved to be significantly better than the first. In fact, the unsatisfactory results of the first approach lead to the idea of implementing the data augmentation techniques and the second approach. More analytically, the best performance in the first approach was achieved by the fourth experiment, the one with the 100 epochs and the big learning rate, which achieved an F1 score of 0.64 for the testing set. The rest of the experiments

didn’t manage to “learn” from the data, resulting in an F1 score in all three sets less than 0.60, and a bigger error value. After deep analysis of these results, the conclusion was that the dataset size is not enough to effectively train the model, since in three out of four experiments the model faced overfitting, a classic problem of small datasets. Thus, there was selected to check the performance of the model with a dataset of a bigger size. In the second approach, the more accurate example was the third, meaning the one with the 100 epochs and the small learning rate, which achieved an F1 score of 0.81 in the testing set. Moreover, the three rest experiments of the second approach passed the F1 score of 0.60 at the testing set that the three first experiments of the first approach failed to do. Their F1 score was 0.71, 0.64, and 0.77 respectively. Therefore, based on the above results it is proved that the selection of augmentation methods is successful and that the model indeed needs a larger number of data to train effectively. Additionally, it is shown that the selection of the learning rate depends primarily on the dataset, but also on the set number of epochs for model’s training. In the first approach two learning rates were not consistent on the two different options of epochs, since in 50-epoch experiments the small learning rate was more successful, but on the 100-epoch experiments the big learning rate achieved better results. On the contrary, in the second approach, in both 50 and 100 epochs experiments, the best performance of the model was achieved by the small learning rate.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF FUNCTIONS

LIST OF ABBREVIATIONS

Abbreviations	**Descriptions**
ACR	American College of Radiology
AF	Activation Function
AHE	Adaptive Histogram Equalization
AI	Artificial Intelligence
ANN	Artificial Neural Network
AUC	Area Under the Curve
BBPHE	Background Brightness Preserving Histogram Equalization
CBIS-DDSM	Curated Breast Imaging Subset of DDSM
CAD	Computer-Aided-Detection
CC	Cranio-Caudal
CE	Cross-Entropy
CHE	Classical Histogram Equalization
CHSJ	Centro Hospitalar de S. Joao
CLAHE	Contrast-Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Network
CPD	Canonical Polyadic Decomposition
DC	Dice Coefficient
DCNN	Deep Convolutional Neural Network
DDSM	Digital Database for Screening Mammography
DHE	Dynamic Histogram Equalization
DICOM	Digital Imaging and Communications in Medicine
DL	Deep Learning
DNA	Deoxyribonucleic Acid

DNN	Deep Neural Network
DP	Data Preprocessing
DT	Decision Tree
FC	Fully-Connected
FFDM	Full-Field Digital Mammography
FN	False Negative
FNA	Fine Needle Aspiration
FNN	Feedforward Neural Network
FP	False Positive
GD	Gradient Descent
GLOBOCAN	Global Cancer Observatory
GT	Ground Truth
HE	Histogram Equalization
KNN	K-Nearest Neighbors
MAE	Mean Absolute Error
MBI	Molecular Breast Imaging
MCC	Matthews Correlation Coefficient
MIAS	Mammographic Image Analysis Society
MIT	Massachusetts Institute of Technology
ML	Machine Learning
MLO	Mediolateral Oblique
MRI	Molecular Breast Imaging
NLP	Natural Language Processing
NMSC	Non-Melanoma Skin Cancer
PACS	Picture Archiving and Communication System
PET	Portion Emission Tomography

BI-RADS	Breast Imaging Reporting and Data System
ROC	Receiver-Operating Curve
ROI	Region of Interest
SGD	Stochastic Gradient Descent
SPC	Samples Per Class
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
TWS	Tensor Window Size
UCHC	University of Connecticut Center
UCHCDM	UCHC DigiMammo
WDBC	Wisconsin Diagnostic Breast Cancer
YOLO	You-Only-Look-Once

Introduction

Breast Cancer

Cancer is a major public health issue globally. It ranks as a leading death cause of humans worldwide, that settles a significant barrier to the increase of the life expectancy. Based on the estimation of the Global Cancer Observatory (GLOBOCAN), in the year 2020, there were around 19.3 million new cases and 10 million cancer deaths, or otherwise 18.1 and 9.9 million respectively, in the case that the non-melanoma skin cancer (NMSC) (except the basal cell carcinoma) is excluded. Some of the most commonly diagnosed types of cancer include breast cancer (11.7% of the new cases among all cancer types), lung cancer (11.4%), prostate cancer (7.3%), skin cancer (6.2%, excluding basal cell carcinoma), and colorectal cancer (6.0%). Correspondingly, the most fatal cancer types seem to be lung cancer (18% of the new deaths among all cancer types), followed by liver cancer, stomach cancer, and female breast cancer, which have 8.3%, 7.7%, and 69%, respectively [1]. Each different type of cancer may have distinct characteristics, risk factors, treatment approaches, and management in the realm of oncology. Cancer is a large group of diseases of higher multicellular organisms that can initiate in various organs or tissues of the body. It can develop in virtually any part of the body and is categorized into numerous different types, depending on the particular cell where it originates. Cells are the basic building blocks of the body since they are part of all tissues and organs. Cells are multiplied and renewed daily by the human body, due to the fact that the old cells die, and for that, the body creates new cells, with the aim of healing injuries and replacing the old worn-out tissue, etc. However, sometimes it can happen for some cells to become abnormal and continue growing, which can lead to cancer [2].

In a clinical context, cancer appears to be a diverse group of diseases exhibiting different phenotypic characteristics and is defined by an uncontrollably abnormal cell growth. This growth comes from various alterations in the gene expression, leading to a dysregulated balance of cell proliferation and cell death in the organ. This imbalance results in the emergence of a different population of cells, named cancer cells. Cancer cells are known for their capability to metastasize in other areas in the body. Metastasis is the process of these cancer cells traveling through the bloodstream or the lymphatic system and spreading to neighboring, or even distant, tissues and organs in the body, where they form new tumors. The characteristic that distinguishes a malignant cancer from a benign tumor is the ability of the former to invade locally, to travel to regional lymph nodes, and to metastasize to distant organs in the body. Put differently, the abnormal growth changes the expression of multiple genes in the host, which respectively results in the change of the normal cellular program. This cellular program concerns the cell division and cell differentiation. The loss of cell differentiation contributes to the creation of a part of cells with different characteristics, including the ability for quick division and evasion of the usual regulatory mechanisms that control cell behavior. Moreover, apoptosis (also known as programmed cell death) and DNA repair, can be affected by the altered gene expression patterns in cancer, through vital signaling pathways that regulate cell cycle progression. The abovementioned changes allow cancer cells continuously divide, without having to "obey" in the cell death signals and accumulate additional genetic alterations over time. Therefore, it is further propelled their malignant behavior. Thus, there is a significant morbidity, and if the host does not manage to be treated in time, dies. An exception to that seems to be latent, indolent cancers that often remain clinically undetectable (or in situ). The existence of an indolent cancer does not imply any danger, allowing the host to have a typical life expectancy [3].

Except in humans and mammals, cancer (or at least a tumorous growth, that is not necessary to spread from one area of the host to another) has been observed in phyla such as Cnidaria (almost 600 million years old), Echinodermata (>500 million years old), Cephalopoda (500 million years old), Amphibia (300 million years old), and Aves (150 million years old). However, it has never been observed in several other phyla including Nematoda, Tradigrada, and Rotifera. Hence, it is interesting to investigate the reason it has not appeared in latter organisms, given the fact that they may have some preventing mechanisms, which protect them from getting tumors. If that is valid, it would be invaluably important for the medical community to find them out, given the fact that it could help scientists to copy it and protect both humans and mammals from that very fatal disease [3].

Breast cancer is one of the most frequently diagnosed cancer types in women and the leading cause of cancer-related deaths, globally. Specifically, in the year 2020, around 2.26 million new cases have been diagnosed, which corresponds to being the most common type of cancer (11.7% among all types of cancer in both genders) and 684,996 deaths have been recorded, which corresponds to 6.9% among all the new cancer deaths in that year, making breast cancer a leading cause of cancer mortality [4, 5]. In other words, this corresponds to one in six deaths from the total patients, which undeniably is a significant percentage. Thus, the pursuit of enhanced prognosis and the development of more effective screening strategies stand as paramount priorities in addressing this pressing medical concern. It has a higher incidence at younger ages compared to lung cancer, with the reported incidence doubling approximately every 10 years until menopause, which indicates that the risk of developing breast cancer is more significant in older individuals. Among others, it has been proved that some of the established risk factors that can cause or affect the occurrence of breast cancer are age, family history, ionizing radiation exposure, alcohol consumption, and reproductive factors. These factors include early menarche, late age at first pregnancy, late menopause, low parity, and elevated levels of both endogenous and exogenous estrogen [6]. Additionally, waist circumference has been positively associated with breast cancer risk among post-menopausal women, as well as height with a risk of overall and hormone-receptor-positive breast cancer [7].

There are four basic types of breast cancer including benign, normal, in situ carcinoma, and invasive carcinoma [8]. A benign tumor slightly changes the breast's anatomy, but it is not toxic and is not considered dangerous [9]. On the other side, in situ carcinoma only affects the system of mammary duct lobules and does not spread to other organs, which makes it not a very harmful type of cancer, since it is relatively easy to treat if detected early [10]. The most severe and fatal type is the invasive carcinoma, which can spread to all other organs [11]. Based on all the above, a patient's health depends on an accurate diagnosis, through highly developed screening techniques. Based on various studies, an early detection significantly improves the chances of a successful treatment [12]. Some of the most common methods for the identification of breast cancer are mammography, X-ray, ultrasound, Portion Emission Tomography (PET), Computed Tomography, temperature measurement, Magnetic Resonance Imaging (MRI), and Molecular Breast Imaging (MBI) [13, 14, 15].

The reason for selecting the mammography technique for research in this thesis, and not one of the several other methods for breast cancer identification, stands for the fact that is the first examination after the clinical breast examination performed by doctors. Mammography is one of the most effective techniques in medical imaging, designed for breast cancer detection, at an early stage, i.e. when it is more treatable, or the detection in general of other breast abnormalities and their diagnosis as malignant or benign [16]. The ultimate goal of that technique is to support early treatment for breast cancer patients, to enhance their survival rate, and to avoid an aggressive treatment, like mastectomy, especially in the modern era where there exist better treatment options [17, 18]. It is a cheap and safe tool, performed with the use of a dedicated X-ray unit for the visualization of breast tissue. It consists of the recording of two views (also known as projections) for each breast: the mediolateral oblique (MLO) view, which is a side view of the breast, and the craniocaudal (CC) view, which is respectively, a top-to-bottom view of the breast. Some of the most common findings that medical experts observe in mammograms include calcifications, masses, architectural distortion of the breast's tissue, and breast asymmetries [19]. A mammogram can be acquired either in screen-film techniques (X-ray film) or in a digital format. The latter approach (digital mammography) has been proven to have various advantages in comparison to the former (film-screen or computer radiography) and thus should be preferred. These advantages include the wide dynamic range, the possibility for contrast enhancement through image post-processing and computer-aided detection, the feasibility of extending digital X-ray imaging of the breast into tomographic and three-dimensional imaging approaches (i.e. tomosynthesis) and computed tomography [20, 21, 22, 23].

In more detail, the radiographic technique that is used for the mammography process includes the compression of the breast for a small duration (around 5-10 seconds), delivering a low dose of radiation, and obtaining high-quality images. It is noted that during the aforementioned breast squeezing, some women have reported that a slight discomfort or even a little pain is felt [20]. The whole bilateral standard procedure, including the preparation that needs to be done before the image acquisition, ranges from 5 to 10 min. It is vital to note that no diagnostic technique is perfect and can detect all the tumors in the breast area. Specifically, mammographic screening may have some limitations like overdiagnosis and overtreatment. Thus, it must be highlighted that based on [24, 25], approximately 28% of cancers might be missing in a mammogram, due to various reasons. Especially in the cases of women who are either during pre-menopausal or who have dense breasts, the possibility of missing a cancer is significantly higher. Mammography can be conducted either as a screening or as a diagnostic setting.

Screening mammography is performed periodically, every 1, 2, or 3 years from the age of 40 – 50 years until around 70 – 75, depending on the screening programs of each nation or region, in order to detect small cancers before they are found either through self-palpation, which all women are advised to perform regularly, or clinical breast examination. It is performed by one specialist, named radiographer, and the resulting images are usually analyzed by two radiologists, independently, in separate sessions. The European guidelines recommend that women between the ages of 50 and 70 years old, to be checked with a screening mammography every 2 years. However, women who have a family history incident with breast cancer, are highly advised to start earlier the periodic checking [20].

Diagnostic mammography, also known as clinical mammography, is performed in cases that present clinical symptoms and its aim is to diagnose if the patient has breast cancer or not. These symptoms can be a palpable lump, skin thickening, nipple discharge, and nipple retraction etc. Similarly to screening mammography, this setting is also performed by a radiographer, and the acquisition of the images includes the two standard views for each breast (CC and MLO). However, it differs from the previous setting in the fact that the images are immediately available for evaluation and that either before, or after the image acquisition, the radiologist performs a full clinical examination in the breasts. This examination has paramount importance, especially in the cases where there is not available a recently done examination of the patient from another doctor. Furthermore, a radiologist may proceed to highlight with a marker any abnormalities or scars, based on his judgment and experience. The examination is always followed by a written report from the radiologist, which notes all the conclusions and recommendations for the next steps of the patient [20]. It has many advantages compared to screening setting, including the lower dose of radiation in women, higher image quality, availability in further post-processing, digital archiving, image transmission, and no chemical pollution [21].

Finally, it must be noted that the likelihood of developing breast cancer because of the mammography procedure is extremely low. Several estimations with models of multiple factors have been developed, and the results reveal that the risk of developing breast cancer from the radiation of screening mammography is at least 100 times lower than the probability of preventing breast cancer death [20, 26]. As previously mentioned, especially after the introduction of digital mammography (FFDM) the radiation dose of mammographic examinations has decreased, and thus also the risk of radiation-induced breast cancer [21].

Based on research [27, 28, 29], the tiring, repetitive, and challenging nature of detecting suspicious findings in the breast area for radiologists results in an undetected lesion rate of approximately 10-30%. To solve this issue, there have been developed several computer-aided detection and diagnosis (CAD) systems, which help medical experts in the interpretation of medical images for more accurate results [30, 31]. The last two decades have seen a surge in the interest in medical artificial intelligence. Given the fact that there are many advantages of artificial intelligence and the affordability and fundamental role of mammography in the diagnostic process of breast cancer, it is intriguing to explore the combination of these two.

Artificial Intelligence

Artificial Intelligence (AI) is a field of computer science and engineering that goals to create agents capable of mimicking human behavior for certain tasks. It is a broader term for the ability of machines to perform intelligent tasks, with minimal human intervention. AI as a term has been closely associated with the invention of robots, which derives from the Czech word “robota”. In Czech it means “forced labor” and it was firstly introduced by the writer Karel Capek in his 1921 play, Rossum's Universal Robots (R.U.R.). Isaac Asimov, the famous American writer, and professor of biochemistry at Boston University, also inspired by the same term, immortalized the common to today word “robot” in many of his science fiction stories, during the middle 20th century [32].

Two prominent figures whose names have been associated with the early development, or put differently, the creation of AI are John McCarthy and Alan Turing. John McCarthy, among other contributions to computer science, coined the term “Artificial Intelligence” in 1955, defining it as “the science and engineering of making intelligent machines”. Besides that, he invented LISP, a programming language used to solve problems in AI, was a great teacher, and played a vital role in the creation of the two prestigious schools in Artificial Intelligence, one at MIT and the other at Stanford [33]. Respectively, Alan Turing, often called the father of modern computing, wrote many foundational papers in the area of AI, including the Turing Test, which is a test designed in a specific way to determine whether a computer machine exhibits human-like intelligence or not [34].

AI has attracted huge interest across a wide range of domains because it can automate tasks that were previously or are currently being performed and require human intervention. Nowadays AI methodologies, among other tasks, are commonly utilized in speech recognition, computer vision, and natural language processing (NLP) [35]. A characteristic of AI systems is that they can analyze data, recognize patterns, make decisions, and improve their performance over time. In the last decades, there has been observed a rapid growth in AI, with huge steps and scientific discoveries each year, due to the practical successes in machine learning (ML). Specifically in the healthcare section, professionals and medical experts utilize AI programs in all parts of their work, such as diagnosis, decision-making in the treatment, and prediction of outcome. These programs include artificial neural networks (ANNs), fuzzy expert systems, evolutionary computation, and hybrid intelligent systems, which are all different approaches within the broader field of AI [36].

In the real world, there is a wide utilization in the applications of ANNs, due to their ability in classification and pattern recognition, with high accuracy. Thus, many researchers have tried to use them in the solving of various problems. The way artificial neural networks are constructed is inspired by the behavior of neurons in biological neural networks, i.e. human brains. A significant advantage of ANNs is that they are able to accommodate interactions as well as nonlinear associations, without user specification. However, an important disadvantage of them is that they lack transparency. That happens due to the existence of the many hidden layers, which make the relationships between the input and output layers challenging [37].

In clinical situations all the different problems that medical experts face (diagnosis, treatment, and predicting outcome) are dependent on many factors including the history of the patient, the symptoms, clinical, biological, and pathological variables etc. Hence, the need for ANNs that can analyze all the aforementioned variables and their weights and intricate the relationships between them is more than necessary [36]. One of the first researchers to utilize an ANN and explore its many potentials was William G. Baxt. More analytically, he developed a nonlinear artificial neural network trained by backpropagation, with the aim to diagnose acute myocardial infarction (coronary occlusion) in patients presenting to the emergency department with acute anterior chest pain [37]. From that moment onwards, along with the the rapid development of computer hardware and software applications, AI and ANNs have been applied in almost every field of medicine with new techniques and approaches being found constantly. Moreover, this revolutionizing technology has helped in the healthcare domain in data digitization, through the constant need for more developed computational models that use AI systems for the extraction of information from the provided data. [35]

Machine Learning (ML) is a subset of AI, which aims in the development of programs and statistical models that allows computers to learn and improve their performance for a specific task, based on the experience they gained throughout training. ML algorithms make predictions or take decisions, without being directly programmed. In 1959, Arthur Lee Samuel, a pioneer in the fields of artificial intelligence and computer gaming, coined the term machine learning and he described it as a “field of study that gives computers the ability to learn without being explicitly programmed” [38]. The significant progress that ML has shown in the past years is based on the new statistical learning algorithms along with the abundance of large datasets and the low computation cost [39]. Lately, a very utilized method is deep learning (DL), due to its ability to solve more complex tasks and the significantly high results that can be achieved, close to the human level performance [40]. The goal of a ML program is to predict with the best accuracy possible future events or scenarios, that are unknown to the computer, based on the dataset that has been selected and the learning method that the system has undergone. Machine learning algorithms can be classified into six categories, including supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, transductive learning, and inductive inference [41].

In this study, is used a supervised machine learning network. The reason for this selection relies on the fact that the ultimate goal of it is to train the model with specific characteristics and features that have been annotated by medical experts. Supervised learning is one of the most, if not the most important methodology in the area of machine learning and holds significant importance in the processing of multimedia data. It involves learning a mapping between a set of input variables X and an output variable Y, utilizing this mapping to predict outputs for new, unseen data. Supervised learning techniques can be characterized more powerful when compared to unsupervised learning techniques, due to their availability of labeled training data, which offers clear criteria for model optimization [42].

Related Work

In recent years, there has been a significant number of attempts for the identification of malignant areas and the classification and segmentation of tumors in the breast area. Researchers have explored various methods, focusing on utilizing primarily state-of-the-art deep learning architectures, specifically Convolutional Neural Networks (CNNs) for both detecting and classifying breast cancer. Some such studies are presented below to provide a comprehensive overview of the existing landscape in the field, as well as to highlight some of the evolving strategies.

In reference [43], the researchers introduce a new CAD system for classifying benign and malignant tumors from mammogram samples, with two distinct approaches being employed. The first approach involves the manual segmentation of the Region of Interest (ROI), while the second approach utilizes the technique of threshold and region based. The deep convolutional neural network (DCNN) named AlexNet is used for feature extraction, whereas the last fully connected (fc) layer of it is connected to a support vector machine (SVM), with the aim to achieve better classification results. The selected datasets of the study, wherein the model was tested are the Digital Database for Screening Mammography (DDSM) and the Curated Breast Imaging Subset of DDSM (CBIS-DDSM).

The novelty of this study relies on the way the ROI is extracted from the mammograms, which is done by two distinct techniques, and that the last fully connected layer of the DCNN architecture is replaced by an SVM. The proposed CAD is divided into various steps, including image enhancement, image segmentation, feature extraction, feature classification, and lastly, an evaluation for the classifier. The block diagram of the study is depicted in the Figure 1 below. For the first step, image enhancement, is employed the contrast-limited adaptive histogram equalization (CLAHE), which is an image contrast improvement method that redistributes its intensity values. CLAHE was preferred over other adaptive histogram equalization (AHE) methods because it introduces a clip level to constraint the local histogram to control the amount of contrast enhancement for each pixel. For the second step, as previously explained there are two different approaches, but it is noted that both are only performed for the DDSM dataset, since the CBIS-DDSM already provides the segmentation by medical experts. Therefore, the first is about the manual determination of circular contours by examining the pixel values of the tumor and using them to crop the region of ROI manually from the dataset. On the other side, in the second approach the ROI is determined by a threshold and the region-based segmentation method. More specifically, the suitable threshold for the study was established to be 76 for all the images of the dataset, and the region with the biggest area within this threshold was determined and cropped automatically. Concerning the third step of the study, the DCNN, is noted that AlexNet is pre-trained with the ImageNet dataset, which includes 1.2 million images, and there are 1000 different classes, but for this research it underwent a fine-tunning, in which the last fully connected layer of it got replaced with a new layer, in order to classify two classes, benign and malignant masses. The last step of the model before the evaluation of the results is the SVM, where the extracted ROIs are classified in one of the two classes with the use of a classifier technique. In the literature there are several classifiers, but the SVM algorithm was selected, since it was the one with the best classification results. For better visualization, the DCNN – SVM fine-tuned AlexNet architecture is shown in the Figure 1, where are depicted analytically the five convolutional layers and the five ReLU (Rectified Linear Unit) activations, the three pooling layers, and finally the three fully connected (fc) layers. As described above, the last fully connected layer is connected to an SVM classifier, with the goal of maximizing, as much as possible, the accuracy of the model.

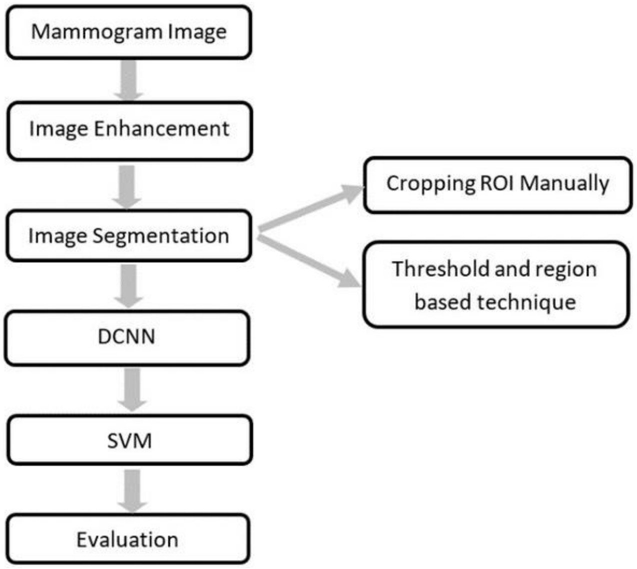


Figure 1. The pipeline (or else block diagram) of the CAD system of the study. Source: [43].

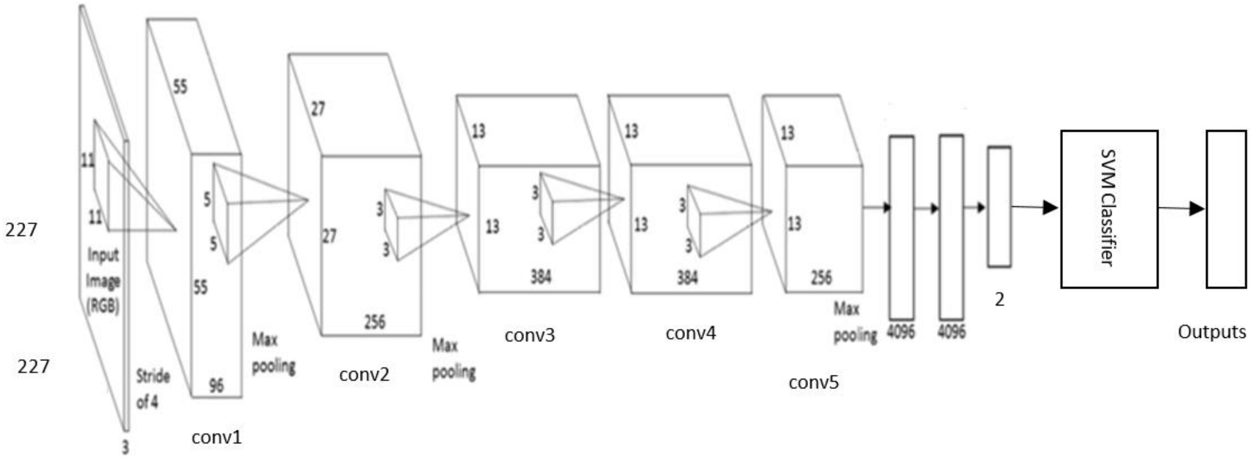


Figure 2. The DCNN –SVM fine-tuned AlexNet architecture. Source: [43].

Concerning the data augmentation, the only method that was applied in the two datasets was the rotation of the images at 0, 90, 180 and 270 degrees. The data was split into two subsets, 70% for the training set and the remainder 30% for the testing set. Additionally, some parameters and configurations of the model were set for the maximization of its performance in this specific task, which is the medical breast cancer diagnosis. Specifically, the iteration number, the primary learning rate, the momentum, and the weight decay are set to 10^4, 10^{-3}, 0.9, and 5×10^{-4}, respectively. Moreover, the model was trained for 20 epochs in each experiment. The selected evaluation metrics by the researchers of this study include the confusion matrix (or else known as error matrix), the accuracy, the receiver-operating curve (ROC), the area under the ROC curve (AUC), the precision, and the F1 score.

For the DDSM dataset, the DCNN for the two segmentation approaches achieved 71.01% and 69.2% accuracy, respectively. In the case where the DCNN got attached to the SVM in order to improve the performance of the model, the first approach managed to improve its accuracy percentage from 71.01% to 79%, while in the second segmentation approach the DCNN with the SVM classifier managed to achieve accuracy equal to 80.9%. Moreover, of the six SVM different kernel functions, the one with the best performance proved to be the linear SVM kernel function for both approaches. The other five kernel functions include the quadratic, the cubic, the fine gaussian, the medium gaussian and the coarse gaussian. More analytically, for the first segmentation approach the linear SVM kernel function achieved an accuracy of 79%, AUC of 88%, sensitivity of 76.3%, specificity of 82.2%, precision of 85% and F1 score of 80%, while the results of the second approach were an accuracy of 80.5%, AUC of 88%, sensitivity of 77.4%, specificity of 84.2%, precision of 86%, and F1 score of 81.5%. It is also noted that the testing error for the first and second segmentation approaches was equal to 30.17% and 30.43%, respectively. Similarly, for the CBIS-DDSM dataset, the simple DCNN for feature extraction and classification achieved an accuracy of 73.6%, while in the case where the SVM classifier was attached to it the percentage reached up to 87.2%. In this dataset the most accurate SVM kernel function among the six previously mentioned, proved to be the medium gaussian, since it achieved the highest percentage in all metrics. Specifically, its results are an accuracy of 87.2%, AUC of 94%, sensitivity of 86.2%, specificity of 87.7%, precision of 88% and F1 score of 87.1%. Hence, it is easily understood that in all experiments, meaning in both approaches of the DDSM dataset and the one on in the CBIS-DDSM dataset, when the last fully connected layer of the AlexNet is replaced by an SVM classifier the model's performance is significantly better than when the model is without it. Additionally, it is noted that all the metrics on the CBIS-DDSM dataset are higher than the DDSM dataset, and that is because the mammograms on the former dataset have already been segmented by specialists. Lastly, it is highlighted that when the proposed model was compared with several state-of-the-art models (both based on the AlexNet architecture and other DCNN such as the GoogLeNet and the

VGG) it managed to achieve the best performance amongst them, with 87.2% accuracy and 0.94% AUC in the CBIS-DDSM dataset.

When comparing the above work with the proposed model of the study, it is noted that the work of [43] has a slightly better performance. However, an important disadvantage of is its computational and training time, given the fact that in the above work, there is an integration of two different types of classifiers (CNN and SVM). Moreover, the model of the reference [43] is a significantly more complicated structure.

In the reference [44], is presented a novel framework for segmentation and classification of breast cancer images, where pre-trained modified U-Net model and various deep learning models are employed. More analytically, this study relies on Mediolateral Oblique (MLO) view and Craniocaudal (CC) view to enhance the system performance, by utilizing five deep learning models for breast cancer diagnosis, which are applied to classify three different mammography datasets into benign and malignant tumors. The pre-trained deep learning models that were selected are: InceptionV3, DenseNet121, ResNet50, VGG16 and MobileNetV2 and were preferred over training from scratch, which can lead to over-fitting, being more time consuming and needing an excessive computing power. On the other hand, the three datasets are: Mammographic Image Analysis Society (MIAS), Digital Database for Screening Mammography (DDSM) and the Curated Breast Imaging Subset of DDSM (CBIS-DDSM). The deep learning models use digitized mammograms with high accuracy and low computational time, while the datasets encompass the distinct cases of breast thickness and size, and the patient’s age, sourced from the (MIAS), (DDSM) and (CBIS-DDSM) databases.

The idea of this research is divided into two phases, with the first being the classification based on different deep models, and the second the segmentation before classification. Initially, the five pre-trained deep learning models are employed to classify the MIAS, DDSM and CBIS-DDSM images into benign or malignant. Subsequently, the segmentation phase follows, where a modified segmentation of the U-Net model is applied to extract the region of interest (breast region) and remove any undesired areas. Finally, after the segmentation phase, the different pre-trained deep learning models are applied to the segmented images in order to classify them into benign or malignant. On the two following figures (Figure 3 and Figure 4) are depicted the proposed framework of the study (Data augmentation + modified U-Net model + Classifier Networks). It is worth mentioning that the data augmentation techniques were applied on DDSM and MIAS datasets to resolve the limited dataset resources, while transfer learning is used with the aim of minimizing both the consuming time and computing resources of the model.

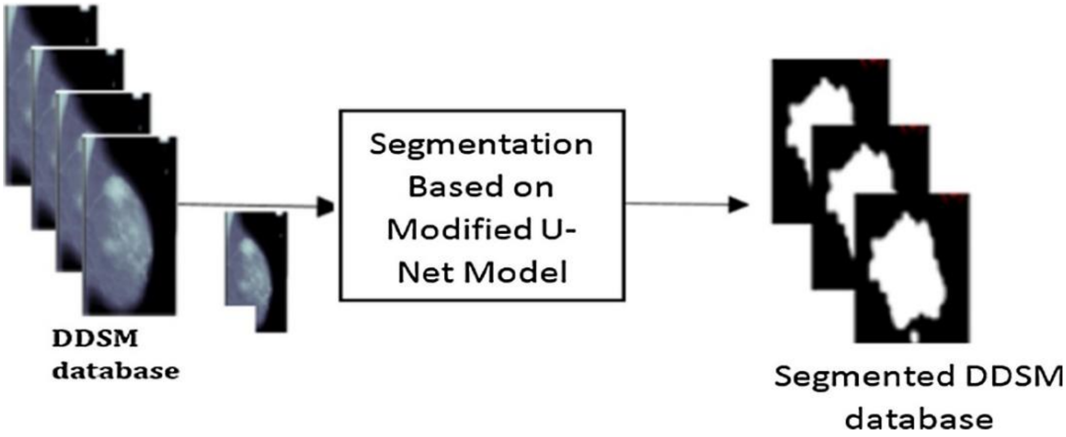


Figure 3. The proposed segmentation framework of the study. Source: [44].

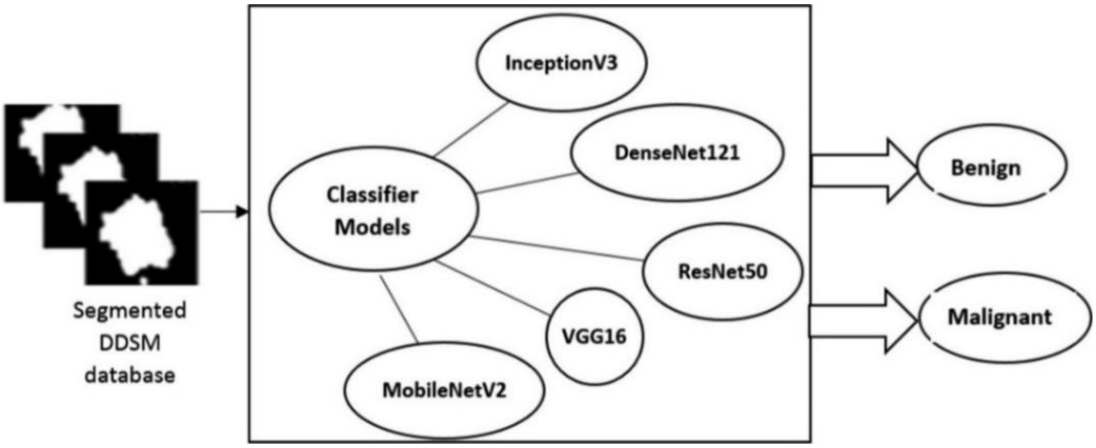


Figure 4. The proposed classification framework of the study. Source: [44].

The size of DDSM, MIAS and CBIS-DDSM is 534, 302 and 300 cases respectively, and all have both MLO and CC view of each breast. The latter dataset was only used to test the model's performance, while the first two were split to be used for training and testing. Concerning the data augmentation technique that was applied in the initial data (DDSM and MIAS datasets), the rotation of the images at 0, 90, 180 and 270 degrees was selected, with the augmented number of images being 2136 and 1208 images, respectively. The evaluation metrics used for the results of the research include intersect over union (IoU), dice coefficients (DC), accuracy, sensitivity (or recall), precision, F1 score, area under the curve (AUC) and computational time.

The findings of this specific research include, among others, that the best segmentation results were achieved on DDSM dataset, followed by MIAS and CBIS-DDSM datasets, all based on MLO view, where the IoU and DC metrics were used. Secondly, it is easily noticeable that the selected data augmentation technique proved successful. This is evident across all evaluation metrics at each stage (classification for the five different pre-trained deep learning models for the DDSM, MIAS and CBIS-DDSM based on the MLO view, segmentation with classification for the different models for the three datasets based on the MLO view and segmentation with classification for the different models for the three datasets based on the MLO and CC views), since the performance is significantly better with data augmentation compared to without, in all three datasets. The evaluation metrics that were employed to assess the performance of the model in all these steps were accuracy, sensitivity, precision, F1 score and AUC.

Of all the five classifier networks (InceptionV3, DenseNet121, ResNet50, VGG16 and MobileNetV2) used for this study, the InceptionV3 model achieved the highest performance of all, since it had in each evaluation metric the highest percentage. Between the two different segmentation approaches of the study (the three datasets based on MLO view and based on MLO and CC views); the best one was proved to be the latter. More specifically, the (Data augmentation + modified segmentation U-Net model + InceptionV3) achieves the best performance among the other classifier networks. The performance of it was: 98.87% accuracy, 98.98% sensitivity, 98.79% precision, 97.99% F1 score, 98.88% AUC and computational time 1.2134 s on DDSM datasets, while the proposed framework when utilizing both MLO and CC view achieves an even better performance, where the metrics are enhanced to: 99.43% accuracy, 99.22% AUC, 99.12% sensitivity, 98.99% precision, 98.98% F1 score. In conclusion, the results of the study reveal that the proposed models are able to achieve a better performance, up to 10% more, than the ones that were used in the literature of the paper. Lastly, the conclusion is that there is no necessity for human intervention with pre- or post-processing or hand-crafted features.

Comparing the proposed model of this study with the [44], it is evident that the latter achieves significantly better segmentation results, but similarly to the previous work (reference [43]), it has a considerably more complex structure. A potential disadvantage of this work, that the proposed model does not have, is the fact that it depends on pre-trained deep learning models, which although leads to very accurate results in the datasets used, but at the same time there is the possibility of low adaptability to other dataset with different characteristics.

In the research [45], is presented a Computer-Aided Detection (CAD) method that integrates the advances of image processing, deep learning, and image-to-image translation for a biomedical application, in order to assist medical professionals in identifying the potential presence of breast cancer. The selected dataset for the study is the UCHC DigiMammo (UCHCDM) database, which is a collection of private datasets from the University of Connecticut Center (UCHC). It includes screening mammograms of 230 different patients, where each one of these cases has a Prior exam (initial screening) and a Current exam (a second follow-up screening between 1 to 6 years after the initial one). The idea is a fusion model that utilizes the You-Only-Look-Once (YOLO) deep learning network, which detects suspicious lesions on Current mammograms and classifies them into Mass, Calcification or Architectural Distortion. In addition, this model will be retrospectively applied to synthetic mammograms, which were created by the Prior mammograms, based on the image-to-image translation models (CycleGAN and Pix2Pix), with the aim of predicting early cancer cases.

CycleGAN and Pix2Pix are two state-of-the-art techniques for image-to-image translation. The main difference between these two models is that Pix2Pix requires paired datasets, while CycleGAN works with unpaired. In more detail, Pix2Pix takes one image from the source domain (A), corrects, and updates its training based on the corresponding image from the target domain (B), while CycleGAN accepts simultaneously two images and performs a cyclic translation (hence the name of the model) across domains, resulting in the the generation of new synthetic images. YOLO architecture is used for simultaneous medical image detection and classification and the main reason that is particularly suitable for CAD applications is its ability for low memory dependence and rapid results. Is a deep learning network where a single CNN architecture model simultaneously identifies and localizes the bounding boxes of objects within entire images and assigns class labels to these objects (classifies them). Based on different studies that utilized the YOLO model and comparing its results to some state-of-the-art methods in mammography detection and classification, the YOLO model achieves a relatively higher accuracy.

Initially, the basic YOLO model was trained under different configurations, involving variations in target class labels. Following this, each one of the experiments was evaluated by identifying the optimal predicted bounding boxes from all the augmented images, including both the original and the rotated images, based on the highest confidence score. The above technique demonstrated its effectiveness for determining the best representative images, to ensure an accurate detection and classification of breast lesions in each mammogram. As shown in the figure below the YOLO model was implemented with the aim of having improved final prediction results. More analytically, in Figure 5 there are two different kinds of YOLO-based models, with the names Model1 and Model2. Model1 was trained and configured for a single class, either mass, calcification, or architectural distortion, while Model2 was configured for multi-class training, encompassing all three classes concurrently. Lastly, the fusion Model denotes the combined evaluation of Model1 and Model2, which was utilized to improve the overall detection performance. The final model is designed in such a way that selects predictions not within the single class predictions, based on a threshold of 0.5, a parameter that yields satisfactory results.

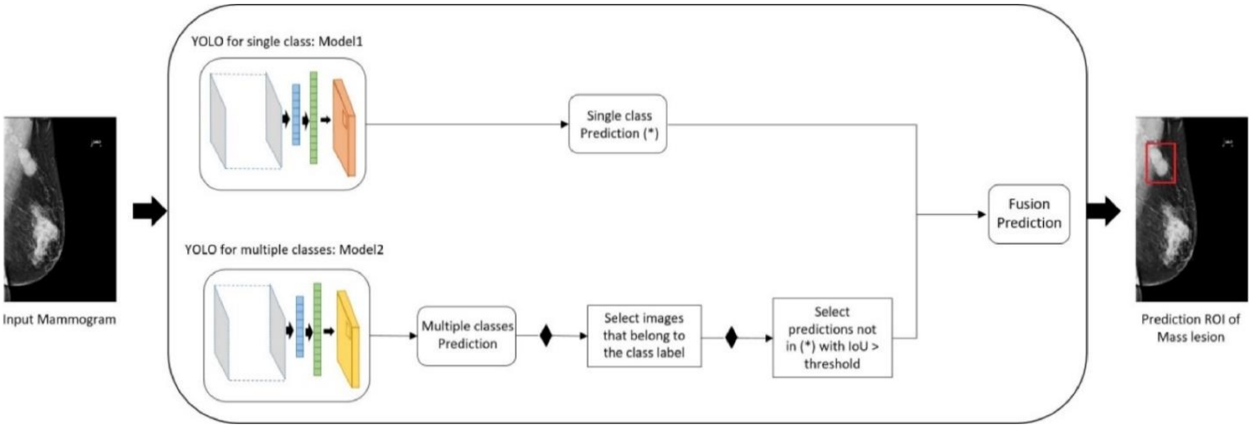

YOLO for single class: Model1
Single class Prediction (*)
Fusion Prediction
YOLO for multiple classes: Model2
Multiple classes Prediction
Select images that belong to the class label
Select predictions not in (*) with IoU > threshold
Input Mammogram
Prediction ROI of Mass lesion

Figure 5. YOLO-based fusion model. Indicative example of a mammogram with Mass lesion. Source: [45].

The evaluation metrics used for the results of the research are intersect over union (IoU), detection accuracy, area under the curve (AUC), sensitivity, precision and recall. Concerning the YOLO-based model on Current mammograms, results reveal the advantages of the adapted fusion Model in comparison to just the Model1 or Model2 and confirm its highest performance for each class label. The overall score of it was 92%, while the score for Model1 and Model2 was 78% and 82%, respectively.

The pipeline of the research is depicted in Figure 6. Firstly, the YOLO-based fusion model is applied to the Current mammograms, aiming to detect breast lesions and classify them into one of the 3 classes (mass, calcification, or architectural distortion) and the rest to normal. Subsequently, the two image-to-image techniques (Pix2Pix and CycleGAN) are employed to establish a mapping between Current mammograms and their corresponding Prior mammograms. This process results in the generation of new Synthetic Prior mammograms that purpose to address the misalignment between the screenings that arose from texture and temporal changes. Furthermore, the two trained models (Model1 and Model2) are used to predict the location and type of breast lesions on the Synthetic Prior mammograms. Given the fact that accurate predictions of the bounding boxes for suspicious lesions of "future cancers" in Prior mammograms are particularly challenging, all the diagnostic information is consolidated into one framework that delves into possible evidence of invisible patterns that might signify the risk of "future cancer". Subsequently, the inference models are directly deployed on translated Prior mammograms and the evaluation is conducted by using the actual positions of the bounding boxes and the class labels of their corresponding Current mammograms.

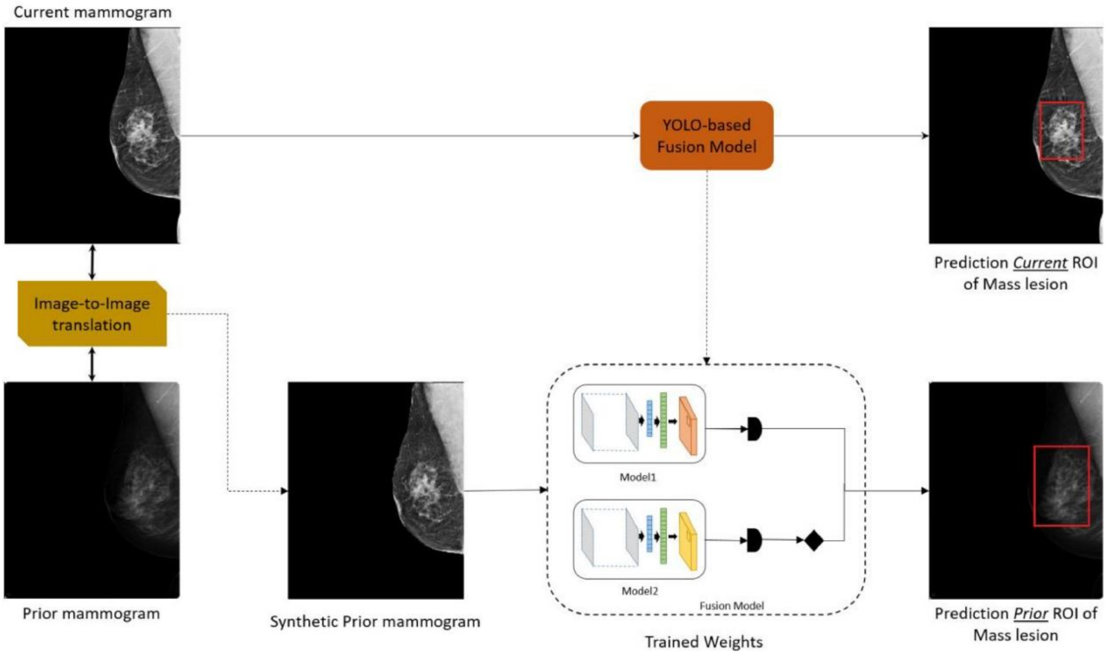

Current mammogram
YOLO-based Fusion Model
Prediction Current ROI of Mass lesion
Image-to-Image translation
Prior mammogram
Synthetic Prior mammogram
Model1
Model2
Fusion Model
Trained Weights
Prediction Prior ROI of Mass lesion

Figure 6. Pipeline of the study (early detection and classification on Prior mammograms). Indicative example of a case with a Prior Exam with normal diagnosis and a Current Exam with mass lesions (visible in the red bounding boxes). Source: [45].

It is important to emphasize that all the experiments conducted with the YOLO-based model shared the same experimental parameters, which included a learning rate of 0.001, a batch size of 8, and a fixed number of epochs set at 100. Additionally, in the second half of the iterations, specifically from the 50th iteration to the 100th, the learning rate was dynamically reduced by 10% every 10 epochs, in case of constant loss function value. This method is also known as “early stopping method”. In the second phase of the study, the emphasis was placed on the use of pairs of screening exams, including Current and their Prior mammograms, aiming to provide an early detection and classification of tumors on the Prior mammograms. More analytically, this was tested with three different approaches on test sets of original Prior mammograms, including only the YOLO fusion model, the YOLO fusion model using CycleGAN technique and the YOLO fusion model using Pix2Pix technique and was found that the highest results were reported by the third approach. In fact, the YOLO-based model that was inferred on synthetic Prior mammograms by Pix2Pix, had a (37% ± 0.1) of the total 124 mammograms, while the YOLO fusion model using CycleGAN had (27% ± 0.07) and the simple YOLO fusion model (28% ± 0.06), comparing all to the expert’s predictions. Subsequently, Pix2Pix emerges as the most effective technique for image-to-image translation of mammograms from Prior to Current appearance, with an overall true prediction rate of 37%, in order to increase the number of correct detection and categorization of breast lesions at the t=0 years.

Additionally, the proposed work was compared with other researches for mass detection, which were evaluated on the public datasets CBIS-DDSM, INbreast, and MIAS, and it is revealed that achieved overall better accuracy rate and inference time. This originates from the fact that the other works were conducted by different configurations and preprocessing techniques, which may achieve different performance on public datasets. Specifically, the proposed methodology showed detection accuracy of 92.09% and 0.62 s for inference time per image. Finally, a comparison with two similar works in early detection of breast lesions showcased the proposed methodology's superiority, which achieved a 36% accuracy of early detection compared to the 20% and 27% of the other studies.

Comparing the proposed model of this study with the [45], it is evident that the latter achieves significantly better segmentation results. However, a possible disadvantage of it is that it uses the UCHC DigiMammo dataset, which is a private. That might result in poor generalization of the model with other open datasets, such as CBIS-DDSM, INbreast, and MIAS, given the fact that there is possible for the private dataset to have different characteristics. Thus, further validation on different, preferably open datasets is needed, in order to check the model’s robustness and reliability. The above is not a problem for the proposed work of this study, which used the INbreast open dataset and in order to overcome the problem of its limited size, implemented various data augmentation techniques. Additionally, it is highlighted that in comparison to the model of this study the model of the reference [45] has a significantly more complex structure.

In the reference [46], the researchers introduce a Rank-R Feedforward Neural Network (FNN), a tensor-based nonlinear learning model that utilizes canonical polyadic decomposition (CPD) on its parameters, that efficiently detects abnormalities on digital mammograms. The proposed model offers notable advantages over the typical ML methods, with the most important being a) that it significantly decreases the number of trainable parameters, which results in a particularly capable approach for problems with small datasets, and b) that it views inputs as multi-dimensional arrays, which signifies no need for vectorization and enabling the comprehensive utilization of structural information across all data dimensions. The selected dataset for the study is the INbreast, a collection of 410 mammograms corresponding to 115 different cases.

The idea of this research is divided into three phases. The first phase is depicted in Figure 7 below, where a peripheral cropping is applied on the initial mammogram, followed by the employment of nine independent filters in the cropped image resulting in a multichannel object that encompasses the raw image. Thus, the input mammogram is converted into a three-dimensional object that contains useful additional information for the next steps of the work. In more detail, the nine basic low-level filters include sobel in combination with different threshold values, the canny edge detector, gaussian difference, gamma correction, histogram normalization, and gabor, and aim to exploit any potential features associated with the ROIs.

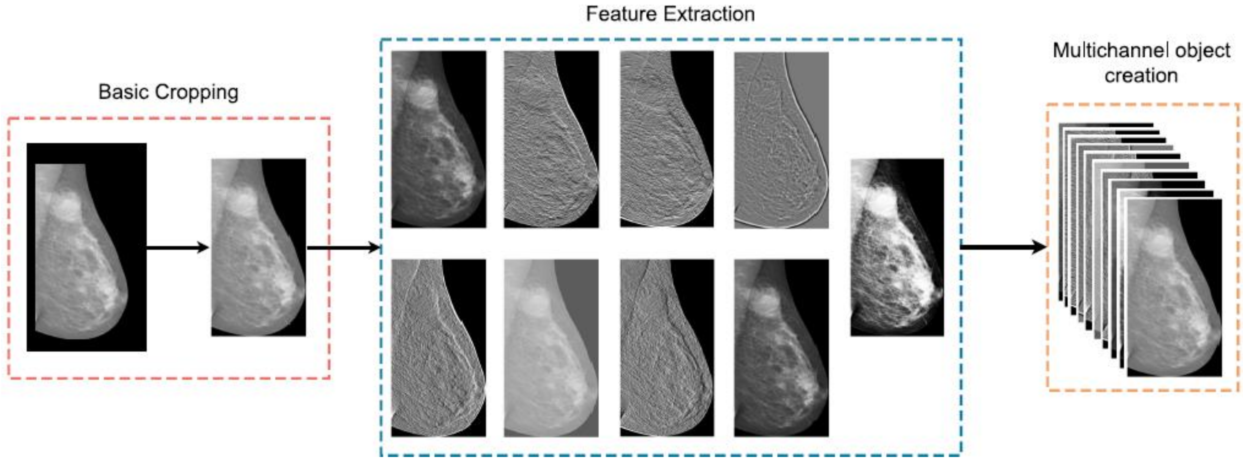


Figure 7. First stage of the framework, where a basic cropping procedure is employed on the input image and low-level features are extracted by digital filters. Source: [46].

In the second phase, as illustrated in the Figure 8, follows the patch extraction, a crucial step due to the extraneous and unwanted information on the mammograms, that remains even after the basic cropping of the first stage. More analytically, the multichannel image undergoes a horizontal traversal using a predefined step, and only patches that have all the three selected criteria are extracted and stored for further processing. These criteria include a) the predefined number of patches to be extracted by a single image, b) the coverage of breast tissue inside a patch to be over 90%, and c) the inclusion of ROIs or part of them in the patch, in case of the existence of any type of lesions in the image. The size of the patches, automatically extracted from each multichannel object, is set to 64 x 64 x 10 pixels.

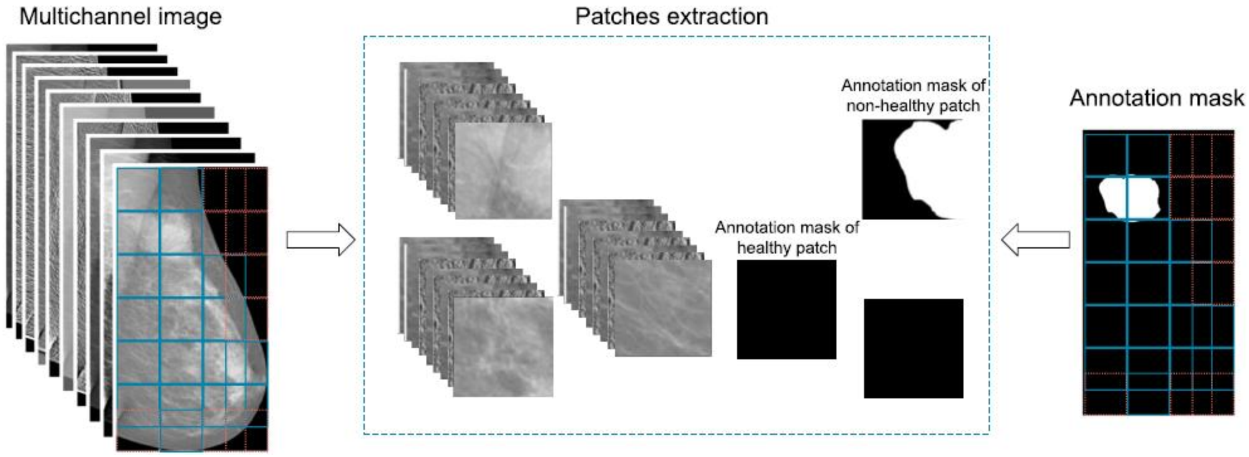


Figure 8. Second stage of the framework, where patches are obtained from the enriched, multichannel image object. This involves extracting relevant areas from the corresponding annotation mask. Source: [46].

Lastly, the third phase involves the tensorization procedure. Tensorization is a process that analyzes a given patch of the image and generates a tensor object for each pixel. That tensor object is referred to as the dominant pixel, its size has the form TWS x TWS x 10 and depends on the tensor window size (TWS) hyper-parameter. The tensor's class is the same as the dominant's pixel, as illustrated in the annotation mask of the corresponding patch. Furthermore, after the end of the tensorization process, all the generated tensor objects are stored in a temporary list. The selection of the samples that will be used for the construction of the training set of the model, is based on the samples per class (SPC) hyper-parameter, while the remaining samples are stored in the training set. In more detail, Figure 8 illustrates the process of splitting 64 x 64 x 10 patches into tensors with size 21 x 21 x 10. Each class is sufficiently represented in the final dataset (based on the SPC hyper-parameter), and the tensor samples are depicted with distinct colors (yellow for healthy tissue, orange for tumor, and green for calcification). On that figure, is also visible the list creation step that followed the tensorization, the splitting of training and testing sets, and their respective purpose, which is training for the model in the former and performance evaluation in the latter.

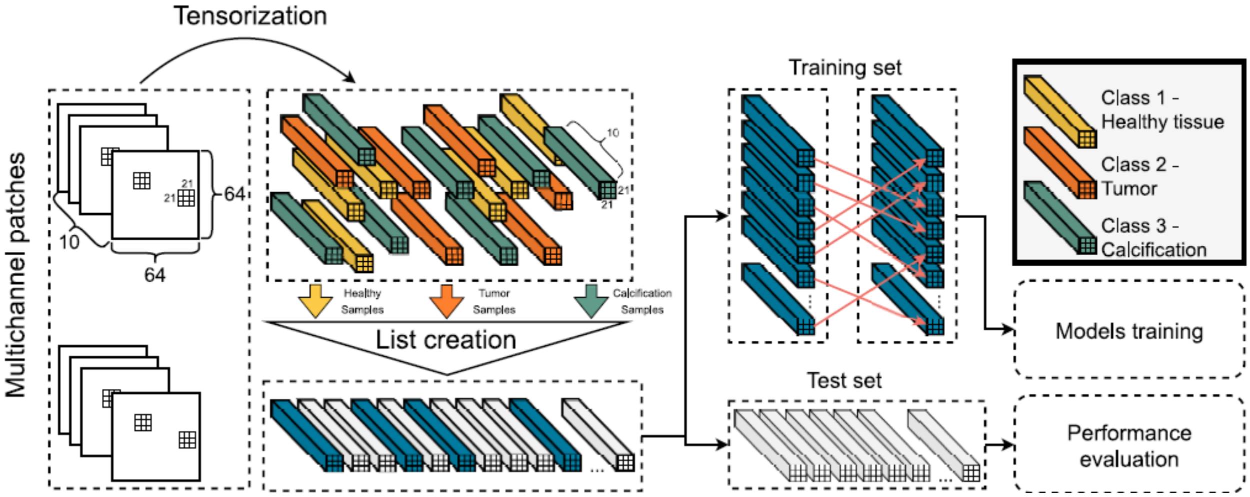


Figure 9. Third stage of the framework, where the selected patches are transformed into tensor objects, which are subsequently stored in a temporary list. Following a sampling process, the training and the testing sets are constructed. A permutation is employed to the former and both sets are the input into the AI models. Source: [46].

The evaluation metrics used are the F1 score and accuracy, followed by the 95% confidence interval. Concerning the experimental procedure of the study, the proposed model was compared against some state-of-the-art deep learning models for detecting abnormalities on digital mammograms. It is important to note that all these state-of-the-art deep learning models were first adapted to fit the INbreast dataset, as well as fine-tuned to achieve a higher performance. All experiments conducted were trained in seventy epochs, with the validation process being applied on the testing test every 10 epochs. In more detail, the conducted experiments are in total six and differ in the experimental parameters SPC and TWS. Specifically, the experimental pairs used are (10 samples, 35 pixels), (40 samples, 35 pixels), (60 samples, 35 pixels), (10 samples, 21 pixels), (40 samples, 21 pixels), and (60 samples, 35 pixels) for SPC and TWS, respectively. Moreover, it should be emphasized that the tensors are randomly selected each time the experiment is repeated, due to random permutation. Thus, all the experiments were conducted several times, with the aim of making sure that all methods are evaluated on most of the information of the original INbreast dataset.

For all models, among 21 or 35 pixels for TWS, is noticed that the latter had a significantly better performance, both in accuracy and F1 score. On the other hand, among 10, 40 and 60 samples for the SPC, the former proved to be the one with the worst performance in all models and metrics, while the other two exhibited relatively comparable results, with the 60-pixel setting showing a slightly better performance. Hence, in almost all the models, both in the proposed as well as the comparing ones, the best accuracy and F1 score was achieved with with the pair SPC and TWS of (40 samples, 35 pixels) and (60 samples, 35 pixels). In those two experimental configurations, the accuracy and the F1 score of the Rank-R FNN model in is significantly higher than what was achieved in the compared models, as well as the deviation of the proposed model is the lowest of all. More analytically, in the aforementioned experiments the proposed tensor-based model achieved accuracy of 88% ± 5% and 90% ± 4% and F1 score of 84% ± 5% and 83% ± 9%, respectively. Furthermore, the proposed model showed a better performance among the several classes, while the model under comparison (AlexNet architecture) showed a higher confusion, specifically for the first two. In general, the proposed method proved to perform better in the experiments with TWS = 35, even for the low values of SPC, with very few parts of overlapping with the AlexNet. On the contrary, with TWS = 21 the model does not achieve a good performance, with a dense overlap.

In conclusion, the presented tensor-based learning model demonstrated a better mean performance than the compared state-of-the-art models, in most of the cases, since it achieved higher accuracy and F1 score. Additionally, the smaller range of the 95% confidence intervals (lower deviation) in the work of the study affirms that the model based on tensors tends to have more robustness and stability in contrast to the comparing researches. On top of that, the proposed model needs a smaller number of epochs for training in most of the experimental tests, comparing to the literature that presented a more unstable training. As shown in the paper, even in the worst scenario the Rank-R FNN model overpasses the literature by approximately 2–5% in accuracy, while in the best-case scenario that percentage can be up to 20%. Thus, the proposed Rank-R FNN model is recommended to be applied for medical data, since it can demonstrate accurate results in cases with limited data, which is a common challenge in the medical datasets.

Comparing the proposed model of this study with the [46], it is evident that the latter achieves better segmentation results. However, a possible drawback is the fact that the Rank-R FNN model is remarkably more complex, which results in difficult interpretability, which in a network like that is essential for overall clinical trust and acceptance. Additionally, it is highlighted that in comparison to the model of this study the model of the reference [46] has a more complicated preprocessing procedure.

In [47], a novel deep feed-forward neural network model with four activation functions (AFs) has been proposed for breast cancer classification: hidden layer 1: Swish, hidden layer, 2:-LeakyReLU, hidden layer 3: ReLU, and final output layer: naturally Sigmoidal. The selected dataset for the study is the Wisconsin Diagnostic Breast Cancer (WDBC) dataset and includes a total of 569 samples and 30 real-valued numerical features computed by a digitalized image of a fine needle aspiration (FNA) of a breast lesion.

An AF is responsible for transforming the result of the neuron's summation function into the final output. Put differently, AFs are mathematical expressions used to determine the output of a neuron (activated or not), based on the weighted sum, and to introduce non-linear characteristics to its output. Different AF can have significant variations in the accuracy and computational performance of a DNN model. Hence, the selection of the most accurate AFs is a challenge. One part of the conclusions of this specific research implies insights into the performance of all four AFs and highlights the preferred AF, based on its performance for classification tasks.

The main goal of the study is to construct a better multiple AFs scheme for the deep feed-forward network, with the aim of improving the accuracy of breast cancer classification. In Figure 10, is depicted the proposed model, where the first hidden layer is the Swish AF, the second hidden layer is the LeakyReLU AF, the third hidden layer is the ReLU AF, and finally the output layer is the Sigmoid AF. As presented in the figure below, the first two hidden layers are followed by dropout layers, in order to ignore some units in the deep neural network (DNN). Thus, the dropout layers aim to avoid any overfitting of the model.

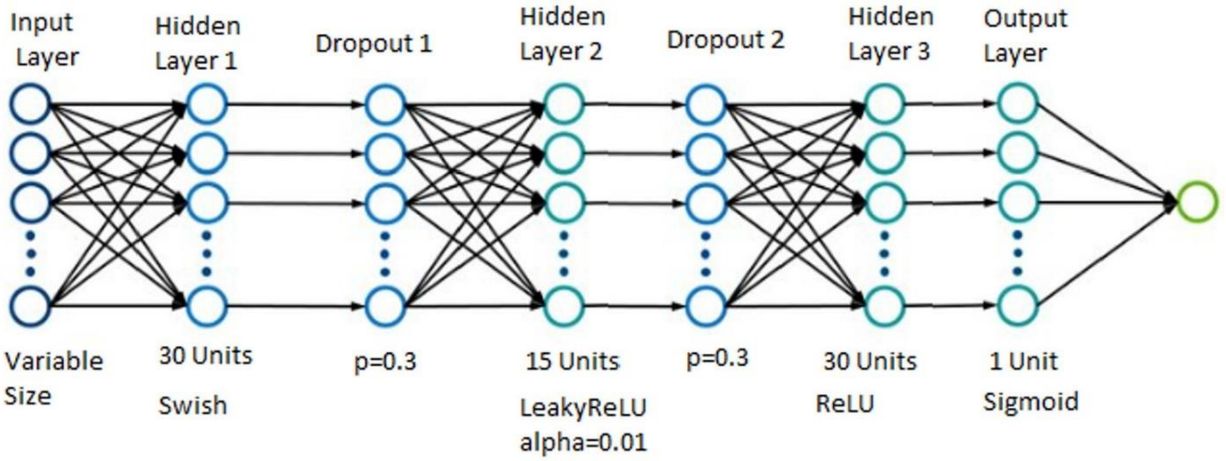


Figure 10. Proposed DNN model, with multiple activation functions. Source: [47].

For comparison purposes, there were used four more DNNs with similar architecture with the proposed model above, but with different AFs. These DNN models include the Sigmoid DNN, Swish DNN, ReLU DNN, and LeakyReLU DNN. Additionally, the proposed model was also compared with many state-of-the-art classifiers, such as decision trees (DTs), support vector machines (SVMs), k-nearest neighbors (KNNs), etc.

Each one of the five DNNs, both the proposed model and the four compared DNNs, were selected to be trained for 50 epochs, using seven different optimizers, including SGD, Adam, Adamax, Adagrad, Adadelta, RMSprop, and Nadam with their default settings, resulting in a total of 35 different models. It is highlighted that for the measurement and comparison of the performance of the 35 models, there was used a repeated stratified 10-fold CV method. The evaluation metrics used for the results of the research include accuracy, F1 score (also known as F-measure), Matthews Correlation Coefficient (MCC), sensitivity, specificity, precision, and recall.

The results of the research showed that the proposed multiple activation deep neural network obtained the best performance of all the other networks, since in all metrics had the highest percentage except specificity and precision, where it came second followed by Coarse Gaussian SVM with 99.72% and 0.995%. In more detail, the work of the study achieved 98.21%, sensitivity 96.65%, specificity 99.13%, precision 0.985%, recall 0.967%, F1-score 0.976%, and MCC 0.962%.

In conclusion, the presented deep feed-forward neural network model with four AFs outperformed the four compared DNN models (Sigmoid DNN, Swish DNN, ReLU DNN, and LeakyReLU DNN), as well as the established state-of-the-art models. Moreover, it became evident that all other DNN models (Swish DNN, ReLU DNN, and LeakyReLU DNN) and many traditional and standard classifiers achieved a better performance than the Sigmoid DNN, and hence it follows that it is not a very good choice for breast cancer classification tasks on the WDBC dataset, since other AF DNNs are significantly better. In contrast, the best results for this type of medical dataset were achieved by ReLU DNNs (ReLU DNN, and LeakyReLU DNN), given the fact that their performance was relatively close and slightly better than the Swish DNN. It is also noted that based on the results of the study, the best optimizer for this classification is the Adagrad optimizer. This is easily understood by the observation that, across the five different DNNs (the proposed one and the four others for comparison), Adagrad managed to outperform the six other optimizers in four out of the five DNNs.

Comparing the proposed model of this study with the [47], it is evident that the latter achieves significantly better segmentation results. However, it is highlighted that is logical and expected, given the fact in comparison to the model of this study the model of the reference [47] has a significantly more complex structure, which results in a better performance.

In the research [48], is presented how CNNs can be utilized for direct classification of pre-segmented breast masses in mammograms as either benign or malignant, through the combination of transfer learning, specific and careful pre-processing procedure and data augmentation techniques, in order to overcome the challenge of insufficient training datasets. The selected dataset for the study is the Digital Database for Screening Mammography (DDSM), a collaboratively maintained public dataset at the University of South Florida. It includes around 2500 cases, each containing both MLO and CC views of both breasts.

The idea of the study is that three different CNN models are trained for breast cancer classification, and their results are compared with the aim to find the best approach. The three networks are a shallow CNN (the baseline model), an AlexNet and a GoogLeNet. For the last two there is used the same base architecture as the original work, but without including the last fully-connected (FC) layer to output two classes. Additionally, there are missing also the two auxiliary classifiers from the GoogLeNet, since it was proved that they impaired the training in practice. The inspiration for the architecture of the shallow CNN is taken by the early layers of the AlexNet network. It consists of three convolutional blocks composed of 3 x 3 convolutions layers, followed by firstly batch normalization, secondly ReLU, and lastly max pooling layers, with 32, 32 and 64 filters each. Moreover, there are also 3 FC layers with a size of 128, 64 and 2, respectively. The final layer of the model is a soft-max, used for the binary classification. It is also noted that, there are used Xavier weight initialization, and the Adam update rule with a base learning rate of 10^{-3} and batch size of 64 pixels. It is important to note that AlexNet model was initialized with convolutional layers with pre-trained weights and a small learning rate multiplier of 0.1, as well as there were randomly initialized 3 FC layers. Correspondingly, for the GoogLeNet network there was utilized the same weight structure; a learning rate of 0.1 for the layers before the Inception_5a module, 1 for both the Inception_5a and Inception_5b modules, and 10 for the last FC layer, in order to achieve a more aggressive learning. Moreover, for the training of the AlexeNet model there were selected the Adam optimizer with a base learning rate of 10^{-3} and dropout rate equal to 0.5, while for the GoogLeNet model there was used the Vanilla SGD optimizer with a base learning rate of 10^{-2} and dropout 0.2.

The area that surrounds a mass can provide useful information for the diagnosis. Thus, it was decided by the researchers to utilize two approaches for providing the network with that specific context. The first includes an input to the network with a region that contains 50 pixels of fixed padding around the mass, aiming to provide a context size that does not depend on the dimensions of the mass (Small Context). On the other hand, the second approach uses proportional padding through the extraction of a region with double the size of the mass bounding box (Large Context). As known, medical image datasets are often relatively small, which is a challenge for researchers that have to overcome. The solution to that is data augmentation. In the study there were used various augmentation techniques, such as rotation, cropping, and mirroring transformations, that aim to enlarge the dataset.

It was tested that a fine-tuned AlexNet achieves significantly better results than the baseline model, with 0.90 and 0.66 percentage of accuracy, respectively. Concerning the two approaches on the size of the context, Small and Large Context, the results of the study reveal that the latter has a higher validation accuracy than the former. In more detail, the AlexNet model without augmentation and the Large Context achieved 0.71 of accuracy, while the with the Small it achieved only 0.64. As presented in the paper, the selected augmentation techniques helped remedy the scarcity of data, and hence helped the network achieve better performance. Moreover, the evaluation metrics used for the results of the research include accuracy, precision, and recall. The best approach of the study includes the Large Context and the augmented dataset. The best model of the three proved to be GoogLeNet, since it managed to to outperform the other two with a significant margin. In more detail, the performance on the testing set that GoogLeNet achieved is an accuracy of 0.929, precision 0.924 and recall 0.934, followed by AlexNet, which had an accuracy of 0.890, precision of 0.908 and recall equal to 0.868. The baseline model proved to be the worst, with an accuracy of 0.604, precision of 0.587 and recall 0.703. The conclusion of the study is that the best approach of the study (GoogLeNet model with the Large Context and the augmented dataset) outperformed medical experts (radiologists), since it managed to achieve recall equal to 0.934 and precision equal to 0.924, while human performance it has been shown that is between 0.745 and 0.923, based on the study [48].

Concerning the comparing of proposed model of this study with the one of the [48], it is evident that similarly to the previous works, the latter achieves a better segmentation performance. This is logical and expected, since the model of this study is not a complex, but a typical U-net architecture, and thus the difference in the results ins justified.

Methodology

In this study, two distinct approaches were utilized, in order to gain a more complete understanding of the U-net's performance, strengths, and weaknesses. The first approach involves a dataset that includes the same number of samples (images and mammograms) as the initial dataset, while the second one utilizes a larger, more diverse, and more robust dataset, which was created by three data augmentation techniques. The first step of the process, which is common for both approaches, includes the preprocessing procedure of the initial dataset, where the mammograms and masks are cropped to isolate the breast area and get resized all in the same dimensions, which are matrixes of 256 x 256 pixels. The pipeline of this research is depicted in the Figure 11 below.

After the preprocessing stage, follows the point where the two approaches are differentiated. The first approach involves the use of the preprocessed dataset in its raw state, without any modifications, and is divided into three subsets. This phase is termed the splitting phase, and specifically, these three subsets are the training, validation, and testing sets, which have 70%, 20%, and 10%, respectively. Subsequently, the training of the neural network is done with the use of the training set, the validation set is employed to assess the training and lastly, the testing set is used to evaluate the model’s performance with previously unseen data.

The second approach followed a similar logic, differing only on the dataset that goes through the U-net. After preprocessing the data augmentation step was implemented to enlarge the dataset's size. In total, three augmentation techniques were employed, which are, respectively: histogram equalization (HE), gamma filtering, and a 180-degree rotation. In more detail, the initial dataset underwent the first augmentation technique, resulting in the first augmented dataset, with a size equal to the initial one (410 images). Likewise, the second augmented dataset, which also consists of 410 images, emanated from the utilization of the second augmentation technique on the initial dataset. In this step, these two datasets, along with the initial dataset, were all merged to create a temporary dataset with the size of 410x3=1230 images. On this temporary dataset comes the third and last augmentation technique to be applied, producing the third augmented dataset, which also contained 1230 images. Finally, the temporary dataset and the third augmented dataset were added together to form the final dataset, which is the one that gets split into the training, validation, and testing sets. Similarly to the first approach, the percentages allocated to these three subsets are 70%, 20%, and 10%, respectively. Once more, the training set was used to train the neural network, the validation set is to validate the results of the training and lastly, the testing set was used to evaluate the performance of the U-net with a new batch of data, that the network has not seen again.

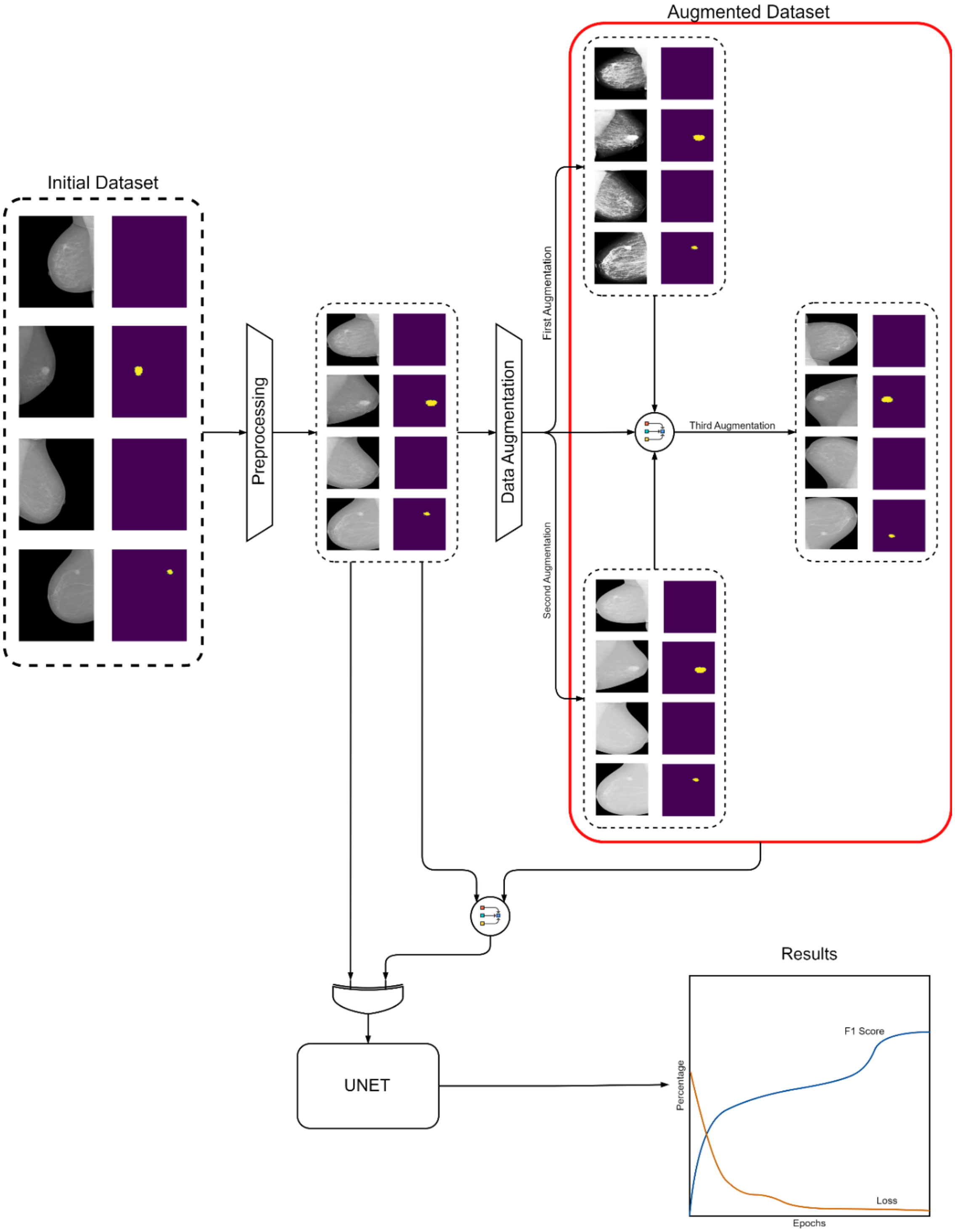


Figure 11. The pipeline of the study. Firstly, the initial dataset goes through the preprocessing process, which involves the cropping and resizing of the mammograms and masks. Subsequently, the study follows two distinct approaches. The first one entails taking the preprocessed dataset and passing it through the U-net model to obtain results, such as the F1 score and loss diagrams. In the second approach, an important step precedes the U-net phase. This step is called data augmentation, where three different augmentation techniques are used, that effectively enlarge the dataset.

Consequently, a final dataset is created by merging the initial and the augmented datasets. This combined dataset is then the input into the model, that similarly to the first approach generates the desired results (F1 score and loss diagrams).

Problem Formulation

In order to ensure a clear understanding of the objectives of this work, it is crucial to formulate the problem of interest and design a robust solution. Towards this direction, is required to mathematically formulate the various components that constitute the problem, borrowing the notation and approach presented in [49].

Let I be denote the final, augmented dataset containing all the mammograms that will be utilized for the purposes of this thesis. Assuming these single-channel images, they can be defined in the two-dimensional domain $\Omega \subset R^2$. Additionally, let $I_0 \in I$ be the observed image and P_1, P_2 two propositional functions identifying the different regions of interest; $P_1(A) = True$ if all $x \in A \subseteq \Omega$ satisfy $P_1(A)$ representing the pixels correspond to lesion, $P_2(A) = True$ if all $x \in A \subseteq \Omega$ satisfy $P_2(A)$ denoting the area of healthy tissue. In a nutshell, the Function 1 that follows is about the propositional functions the P_1 and P_2, which are utilized for identifying different regions of interest, where I^* a proper estimation of I_0.

$$P_1(A) = \{I^*(x) = 1, \forall x \in A\}, P_2(A) = \{I^*(x) = 0, \forall x \in A\}$$

Function 1. Propositional functions for identifying regions of interest P_1 and P_2.

From the formulation of the image segmentation problem, it is obvious that a crucial part is missing: a function f needs to be found that, given an input image I_0, will yield a proper estimated segmentation I^*. For this reason, machine learning techniques are employed in an attempt to form the f such that $I^* = f(I_0)$. To achieve this, a U-net model is proposed that attempts to minimize the cross-entropy loss aiming to calculate a set Θ of parameters θ that determine the form of f. Thus, the goal is to find f_θ such that $I^* = f_\theta(I_0)$ or $I^* = f(I_0, \theta^*)$, with $f(\cdot, \theta^*): I \to \Sigma$, where Σ is the possible segmentations of images in I. Therefore, the training set consist of multiple pairs in $X \times Y \subset I \times \Sigma$.

By using this, the model can be trained by minimizing the loss function as depicted in the Function 2 below, on which L_{CE} represents the cross-entropy loss.

$$\theta^* = arg \min_{\theta \in \Theta} L_{CE}(X, Y, \theta)$$

Function 2. Training Parameter Optimization.

Following the aforementioned procedure, a well-formed function can be defined that produces the desired predictions for the case study. It is noted that image segmentation is a very important and challenging task in computer vision, that aims to separate a digital image into different parts that share the same features and properties with each other. Its primary objective is the fact that it can transform and represent an image in a meaningful and easily analyzable format. Various fields utilize this technique, with some of them being medical imaging, scene understanding, object detection and recognition tasks, robotic perception, etc. [50]. An image segmentation task can be categorized into three types, semantic segmentation, which associates each pixel of the image with a specific class label, instance segmentation, which masks each instance of an object that is presented in the image independently, and panoptic segmentation, which can be described as the combination of the two other groups [51]. Nowadays, convolutional neural networks are one of the most common approaches for image segmentation tasks, since they provide exceptional results [50, 51].

Dataset Description

For the purpose of this current study, the INbreast database [19] was utilized. INbreast is an open dataset that was originally collected from the Breast Center in Centro Hospitalar de S. Joao (CHSJ), in Porto, under the permission of both the National Committee of Data Protection and the Hospital's Ethics Committee. INbreast database collects data from April 2008 to July 2010, and the equipment used for their acquisition was MammoNovation Siemens FFDM, with an amorphous selenium solid state detector, 70 μm (microns) pixel size, and 14-bit contrast resolution. The dimensions of the images are either a matrix of 3328 x 4084 or 2560 x 3328-pixels, and that depends on the compression plate that was used during the acquisition, which obviously depends directly on the breast size of the patient. The format in which the images are saved is the Digital Imaging and Communications in Medicine (DICOM). It is important to highlight that all medical information about the patient of each mammogram was intentionally erased from the DICOM file, with the aim of the database remaining anonymous and confidential.

The dataset includes 115 different cases with a total of 410 images. Among these 115 cases, 90 are women with both MLO and CC views on each breast, and the remaining 25 cases are from mastectomy patients. Thus, from the former type of patients, there are included 4 images per case, resulting in 360 images in total, and from the latter only two views of only one breast per case, resulting in 50 images. It is also noted that 8 cases from the group of patients that have both MLO and CC views were acquired with a follow-up, i.e. in different times and not all in the same visit in the hospital. In the database, there is recorded a variety of different types of mammograms including mammograms with the presence of mass, calcification, architectural distortions, asymmetries, and multiple findings at once. Also, a big part of the dataset belongs to normal mammograms, meaning healthy breasts. In Figure 12 is depicted the distribution of the aforementioned six different types of mammograms presented in the INbreast dataset. It is easily understood that more than half of the mammograms of the dataset consists of mammograms with calcification, followed by mammograms with multiple findings, and normal ones.

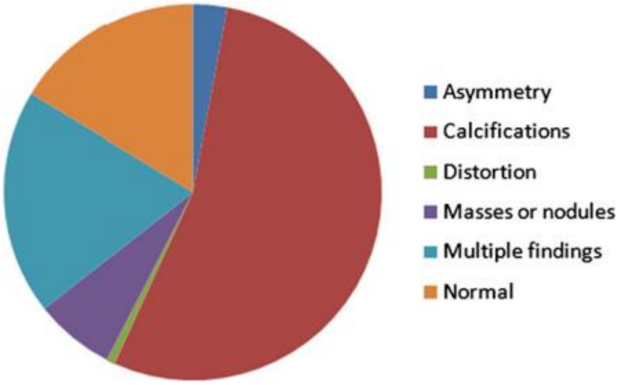


Figure 12. Pie chart with the different types of mammograms found in the INbreast database. Source: [19].

A main feature of the INbreast dataset is that each mammogram has carefully been associated with the ground truth (GT) annotation. These annotations were first precisely done by a medical expert in the breast cancer field, and after were validated by a second specialist, between April 2010 and December 2010. In the situations where the two specialists did not agree on the diagnosis, the case was remained under discussion until a joint decision made. The application that was used for the creation of these annotations is the open-source picture archiving and communication system (PACS) workstation named OsiriX, which operates on a Macintosh platform. Each one of the findings got assigned a label, which identified the type of lesion both experts agreed that was present on each mammogram. In total, the medical experts observed seven different types of annotations in the 410 mammograms, including asymmetry, calcification, mass, distortion, pectoral muscle (only in the MLO view), cluster (of MCCs), and spiculated region. For the first five types (asymmetry, calcification, mass, distortion, and pectoral muscle) a precise contour of the finding was created by the specialists, while for the clusters they drew an ellipse, which encloses the entire finding. Lastly, for the cases that were observed in the last type (spiculated region) the experts combined the contour line of the denser region of the mass with an ellipse that encloses all the spicules of the breast. These annotations are available through an XML format, which is constructed as described below:

- A standard header indicating the XML version and type and encoding information.
- The tag <key>NumberOfROIs</key>, followed by an integer number, which shows the number of the present annotations in the corresponding mammogram.
- For each region of interest (ROI), exist three tags. Firstly, there is the tag <key>Area</key>, which is followed by the area value that the current ROI covers. Additionally, there is the tag <key>Center</key>, which is followed by the coordinates of the center point of that specific ROI, and lastly, there is the tag <key>Name</key>, which is followed by the category of finding (mass, calcification, distortion, spiculated region), on which the experts have agreed on, and also some other general information that concern that ROI.

- Subsequently, following the general information for each ROI, there is a list of contour points between the two tags <array> and </array>.

Additional information that concerns the age of the patient, the family history (FX), the exact time that the mammogram was taken, the ACR breast density annotation and the Breast Imaging Reporting and Data System (BI-RADS) is also provided. BI-RADS is a classification system, proposed by the American College of Radiology (ACR) in 1986, with the original report released in 1993 [52]. The latest edition of that system is called BI-RADS 5 (2013), and it contains six classifications for the breast lesions [53, 54]. It is noted that the BI-RADS classes 1 and 2 are considered benign masses, and thus a biopsy was not needed. In contrast, for the rest four BI-RADS classes (3, 4, 5, 6), in some cases, there was selected to be done a biopsy testing, therefore in those cases the biopsy result is also included. Therefore, the biopsy procedure was in total done in 56 cases, which in only 11 the result of it was benign, while in the remaining 45 cases it was found that the tumor is malignant. The distribution of benign and malignant cases is depicted in Figure 13. In total, there can be found 116 masses in the whole dataset, among 107 mammograms that have a tumor, resulting in ≈1.1 masses per image. It is also highlighted that one of the most important breast characteristics is the density, since it can affect the ease of analysis in a mammogram. In fact, specialties find bigger difficulties with the analysis of denser breasts compared to the nondenser ones. The INbreast database, for each mammogram includes its density in the ACR standard scale.

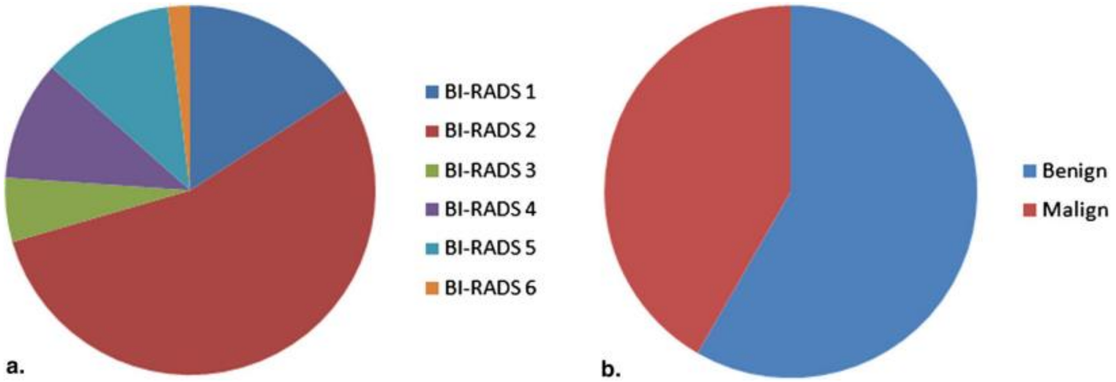


Figure 13. Pie chart (a) depicts the mammogram distribution in the BI-RADS classification and (b) the distribution of the tumor cases, between benign and malignant. Source: [19].

The advantages of the INbreast database lie in the use of full-field digital mammograms (as opposed to digitized mammograms that other datasets have used) and its inclusion of a diverse range of cases. Additionally, the careful and detailed annotations by the specialists and the fact that it is publicly accessible make the dataset very adequate for future works centered or related to breast cancer imaging. Below is followed the example Figure 14 that presents two mammograms (a), their respective masks (b), and their combination with a transparency of 0.35 in the mask layer (c) for a clearer visualization. Specifically, the presentation includes one mammograph without abnormalities (the first row of the figure) and one with abnormalities (the second row).

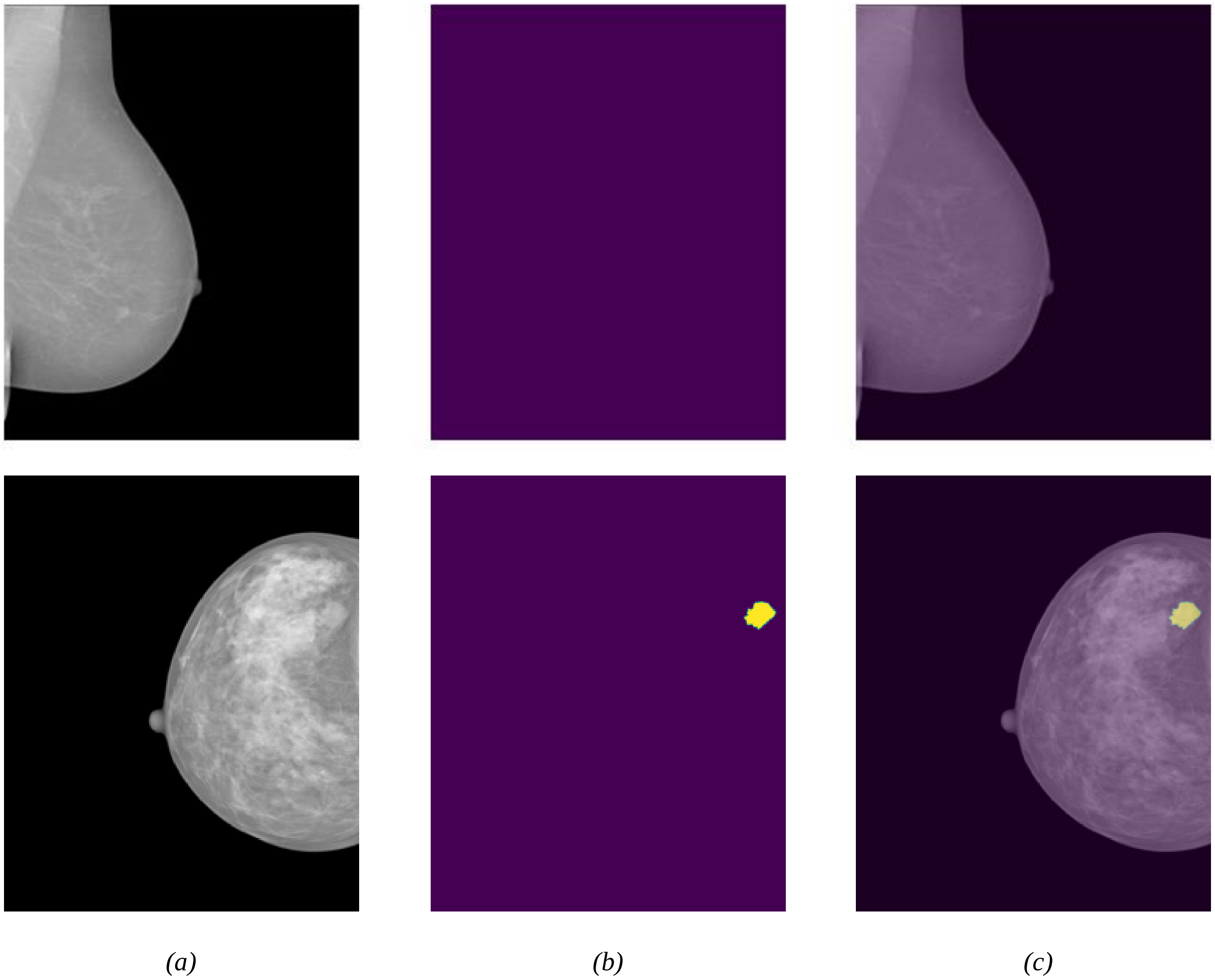


Figure 14. Indicative presentation of (a) the two mammograms, (b) their respective masks, and (c) their combination with a transparency of 0.35 in the mask layer, from the initial dataset INbreast.

Data Preprocessing

In most cases, data preprocessing (DP) has a fundamental role in building robust and accurate ML networks. It has been shown to enhance model results, especially in generalization performance [55, 56]. Through data preprocessing the initial (raw) data are transformed into a format simpler and more effective for the model's next steps [57]. DP techniques can be categorized into three groups: data reduction, missing-data treatment, and data projection. In the first group, there are techniques that intend to decrease the size of a dataset, either through means of feature selection or case selection. Respectively, the second group, missing-data treatments, are methods that erase missing values from the images, are replace them with the estimates, and lastly, missing-data treatments are methods that aim to transform the appearance of the images, such as scaling and resizing [55]. In this study, image preprocessing was deemed necessary to improve data compatibility in the model and to make it more efficient. The preprocessing techniques that were selected to be applied to the dataset are three and include cropping, resizing, and normalization of the images.

Cropping is a popular method that aims for the better visualization of an image. There are several different techniques of cropping that can be divided into two categories: attention-based approach and aesthetics-based approach. The former group emphasizes in the recognition of the main region within the scene based on attention scores computed across the image, while the latter focuses on the overall visual appeal of the cropped image by centering on composition-related properties [58]. The selected cropping method belongs to the first group and employs a bounding box to isolate the ROI of the image and eliminate extraneous information in both mammographs and their corresponding masks. Given the fact that in the INbreast dataset, each mammogram contains a large empty (black) area on one or more sides (either on the right, left, top, bottom side, or even a combination of them), that is not valuable for the model, but on the contrary, it can decrease the performance of it, due to more unwanted information presented in each image, it was considered appropriate to be eliminated. In more detail, cropping focused solely on the relevant and essential area, the breast. Therefore, the selected cropping technique sets the bounders of the bounding box of each mammogram based on the presence of non-zero-pixel values (non-black). The same cropping process is also applied to the corresponding mask, in order to ensure consistency between the cropped mammogram and its associated mask.

As outlined in the corresponding chapter (Dataset Description), the INbreast dataset comprises images with dimensions of either 3328 x 4084 or 2560 x 3328 pixels, which are considered to be particularly large for the model, resulting in a very slow learning process. However, these were not the dimensions of the images at that stage of the research, since the cropping technique has been implemented, resulting into smaller image dimensions. Drawing from prior research and understanding the trade-offs between model accuracy and computational efficiency, it was determined that smaller image dimensions could be used without significant loss in accuracy but with considerable gains in computational speed. This is crucial given the time-intensive nature of machine learning models Moreover, the U-net architecture utilized in this study necessitates uniform image dimensions for proper input, with consistency required across both the x and y axes. The above means that all images have to be the exact dimensions and to have a squared shape. Based on all the above, the selected dimensions for that resizing process are set to be a 256 x 256-pixel matrix. Resizing is a basic and widespread pre-processing procedure in the area of computer vision, since machine and deep learning models are trained quickly on datasets with small images [59]. By resizing an image, the information presented on it gets simplified, which effectively alters the approach to the analysis. This change aims to improve the learning time and by that the training time for the architecture in the following stages. Spatial resizing is mainly performed for at least of one of the following three reasons: 1) a model constructed with mini-batch learning, through gradient descent, that requires the same dimensions (resolutions) for all images in each batch, 2) memory limitations, that do not allow CNNs to enable the inputting of images with high resolutions, and 3) image with significantly large sizes that lead to a slower training process of the model. It is widely known that every system has a limit on its memory budget, and thus there is inevitable to not be a trade-off between the memory occupied by the spatial resolution and the batch size. The selected trade-off can have substantial effects on the CNN's performance [60]. A resize of an image can be either upscaling or downscaling. The former type enlarges the image by extrapolation, while the latter reduces it through interpolation. The most common of the two is the second, since images are typically larger than the input dimension of learning models and thus image-scaling attacks focus on downscaling. Popular imaging libraries, such as OpenCV (which is the one who got used in the code's study) and Pillow, perform downscaling by first resizing images horizontally and then vertically [59]. For a more complete understating, the preprocessing steps that were selected to applied into the initial dataset are presented in the Figure 15.

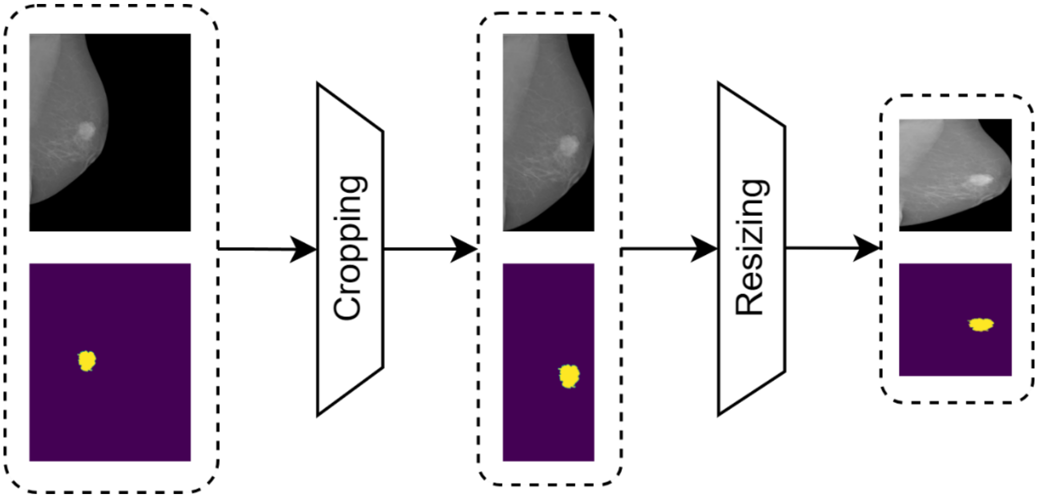


Figure 15. The preprocessing procedure of the dataset. Initially a cropping technique is applied to eliminate extraneous information from the images, followed by a resizing process which sets the size of them all to 256 x 256 pixels.

In the next step, is followed the normalization of the data. However, it is noted that normalization is not a, strictly speaking, form of preprocessing, given the fact that it is not applied directly to the input vectors, but can be seen as a kernel interpretation of the preprocessing [61]. The term normalization was coined by computer scientist E. F. Codd in 1972 and is a common scaling or mapping technique that is used to create a new range based on the existing one of the features [62, 61, 63]. Put differently, normalization can be described as a "scaling down" transformation of the features of an image that reduces the training time of the model as the magnitudes of values are scaled down to significantly lower levels. This happens because after the normalization, the features share a common scale, i.e. in a range between 0 and 1, while before, they could have big differences in the maximum and minimum values [56, 57, 61]. Normalization has been proven to be significantly beneficial for prediction purposes in ANNs and k-Nearest Neighborhood algorithms [56]. The three most popular techniques of normalization are min-max, z-score and decimal scaling [56, 62].

In this study, the min-max normalization was selected, which is also known as feature scaling. Min-max normalization involves a linear transformation of the original data range into a pre-defined new interval (New_{min}, New_{max}). In other words, it is a simple normalization technique that fits the data in a specified range, typically between 0 and 1. Moreover, min-max ensures that all relationships of the data values are preserved, without introducing any potential bias to the data [62, 61]. The min-max method is divided into two steps. The first step includes the identification of the extrema, viz. determining the minimum (X_{min}) and the maximum (X_{max}) values within the dataset that is undergoing the normalization process, while the second is about the calculation of the range of values by using the extreme values from the first step (Range = $X_{max} – X_{min}$). Function 3 presents the min-max normalization formula, where X_{norm} corresponds to the result of normalization, X is the initial value before it, and X_{min} and X_{max} represent the minimum and maximum values of each feature in the image, respectively [57].

$$X_{norm} = \frac{(X - X_{min})}{(X_{max} - X_{min})} = \frac{(X - X_{min})}{Range}$$

Function 3. Min-Max Formula.

Lastly, there was applied a random permutation to the dataset. This method was implemented to shuffle the dataset, aiming to ensure that the order of samples is randomized and that the model will be exposed to different patterns in each experiment, meaning each time the model is trained. By randomizing the dataset's order, the model is less likely to learn, or even better to say “memorize” specific patterns or sequences that might not generalize well to new, unseen data and thus making the model shallow [64]. In the assumption of existing a dataset A with n elements and all are numbered in a certain order by 1, 2, ... to n. The random permutation is defined as the method of reordering these elements in a way that each element gets assigned one and only random number (the match is 1-1) from the range of 1 to n. Put differently, a random permutation is a one-to-one map from the dataset to itself, that is, a map that assigns to every element from the dataset exactly one other element [63]. In practical terms, after each experiment (training session), the entire dataset is randomly shuffled, providing the model with a different data order. Hence, it is ensured that the model's performance evaluation is based on a variety of data arrangements, avoiding any bias introduced by the original order of the dataset. Consequently, accurate and unbiased information about the model's accuracy is obtained, and there is no repetition of the same experiment, contributing to a more comprehensive and complete understanding of the model's strengths and weaknesses. Random permutation techniques are also widely utilized for privacy-preserving purposes in several fields, such as data analysis, linear programming, support vector machine, and clustering [64, 65]. The security of random permutation method relies on the number of possibilities, which grow exponentially with the number of elements that the dataset contains [64].

Figure 16 displays the output of the six images ((a) the two mammograms, (b) their respective masks, and (c) their combination with a transparency of 0.35 in the mask layer) of the Figure 14 after being subjected to the preprocessing procedure.

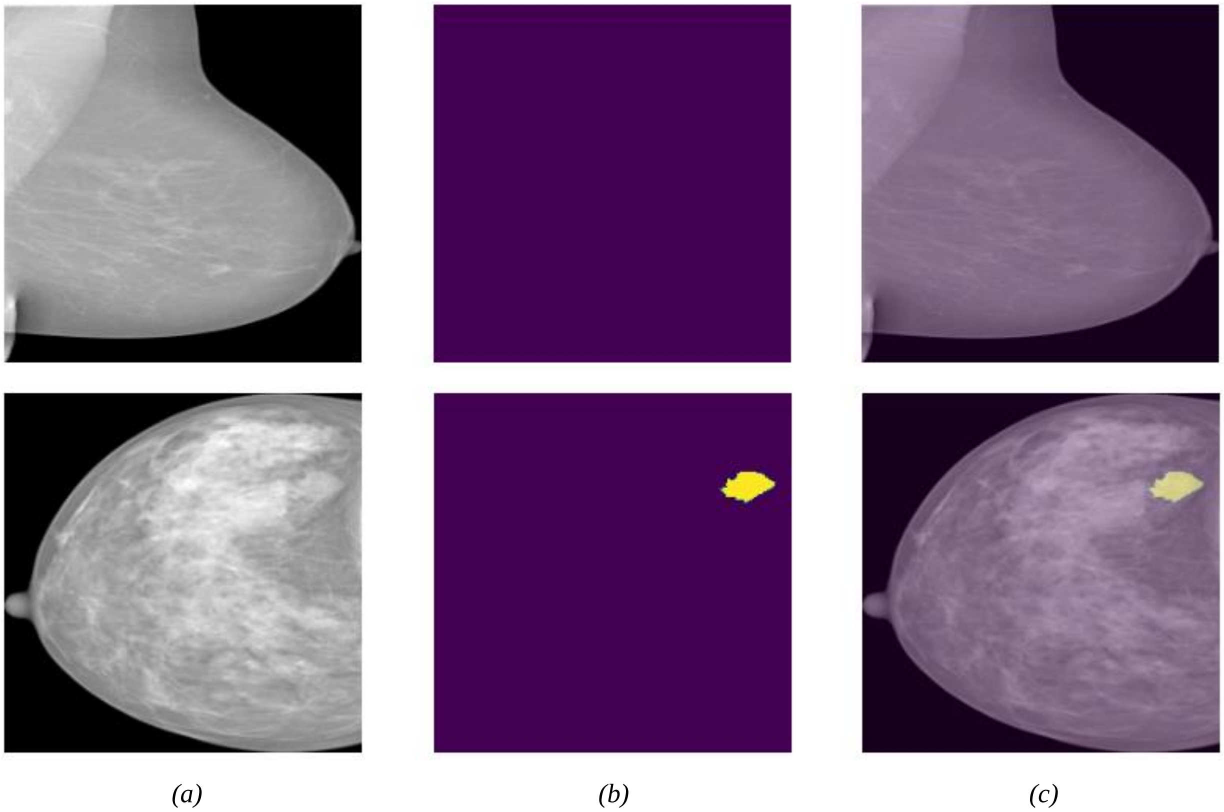


Figure 16. The output of a) the two mammograms, (b) their respective masks, and (c) their combination with a transparency of 0.35 in the mask layer.

Data Augmentation

For the second approach of the study, there was required a larger amount of data, at least more than 1000 samples. Due to the limited number of initial data, which consisted of only 410 images, the technique of data augmentation was employed. Data augmentation is the process of artificially increasing the amount of data by applying various transformation techniques to deform the existing dataset [66, 67]. Additionally, it is important to note that the newly generated data do not lose the conceptual meaning of each record [68]. The purpose of using the data augmentation techniques is to create additional, synthetic training data for the cases where the available data is deemed insufficient [69]. Based on the research [67], a data augmentation technique falls into one of three groups: spatial transformation, color distortion, and information dropping. The first group, spatial transformation, includes basic data augmentation techniques, including random scale, cropping, flipping random rotation, etc., which are all extensively utilized in model training. The color distortion group includes alterations in brightness, hue, etc., and is also widely used in many models. The ultimate goal of these two groups is to transform the training data effectively, mimicking real-world scenarios by modifying specific information channels. The last category, information deletion, is a very popular approach in the area of data augmentation recently, due to its efficient results. It contains techniques like random erasing, cutout, hide-and-seek, etc. It is widely acknowledged that eliminating certain information in the image allows the CNNs to focus on the learning of less sensitive and more crucial information. That results in an increase in the model's perception field and robustness [67].

The importance of the deployment of data augmentation techniques for learning invariance has been proven by Dosovitskiy et al. in the area of unsupervised feature learning [70]. Proper data augmentation is significantly important while aiming for the best segmentation results, particularly in the cases where the training dataset is small. It is widely known that in practical semantic segmentation problems, collecting and annotating sufficient data for the networks is really challenging, given the fact that it can be an expensive and time-consuming task, or even impossible in some cases. For this reason, data augmentation techniques come into play to address the above issue. The goal of data augmentation is to increase the generalizability of the model, given the fact that the CNN will constantly see new, slightly modified versions of the input data, which will make it able to learn more robust features during the training process. Moreover, data augmentation is considered to be a very effective regularization method, that in comparison to others, has several benefits. For instance, in a data augmentation technique the operation concerns solely the training set (input), instead of alternating the network's structure. In general, data augmentation proves easy to apply across various tasks, whereas alternative loss- or label-based methods might require additional design considerations [67]. It is also noteworthy to mention that the data augmentation process helps the model avoid overfitting, which leads to poor generalization. That assistance stems from from the fact that data augmentation manipulates the training dataset, which is the the root cause of a potential overfitting. [71, 72]. This happens under the assumption that additional valuable information will be derived from the original dataset, through the augmentation techniques, which artificially increase the size of the training dataset, either by oversampling or data warping augmentations. The former augmentations (oversampling) generate synthetic instances from the existing images, while the latter (data warping) alter the information presented in the existing images without changing their label. The data warping group contains several augmentation techniques including geometric and color transformations, adversarial training, random erasing, and neural style transfer. In both cases, the generated images are attached to the training set [71, 72]. Lastly, it is noteworthy to add that data augmentation techniques have been proven to be more efficient in small datasets, compared to big datasets, where the effect is less seen [67].

In this study, as shown in the Figure 17, three distinct augmentation techniques were chosen: histogram equalization (HE), gamma filter, and 180-degree rotation. The first two techniques were applied to the initial dataset, generating an additional 410 images for each method. Subsequently, a temporary dataset was created, comprising the original dataset along with the first two augmented datasets (resulting from histogram equalization (HE) and the gamma filter). On this dataset there was applied the third augmentation technique, a 180-degree rotation (or it can be described as vertical flipping), which resulted in the same dataset size but rotated. Finally, the temporary dataset and the third augmented dataset were added together, resulting in the final one, which has a length of 2460 images. The selected augmentations are considered justified, given the fact that masses have no inherent orientation, and thus their diagnosis is invariant to these transformations.

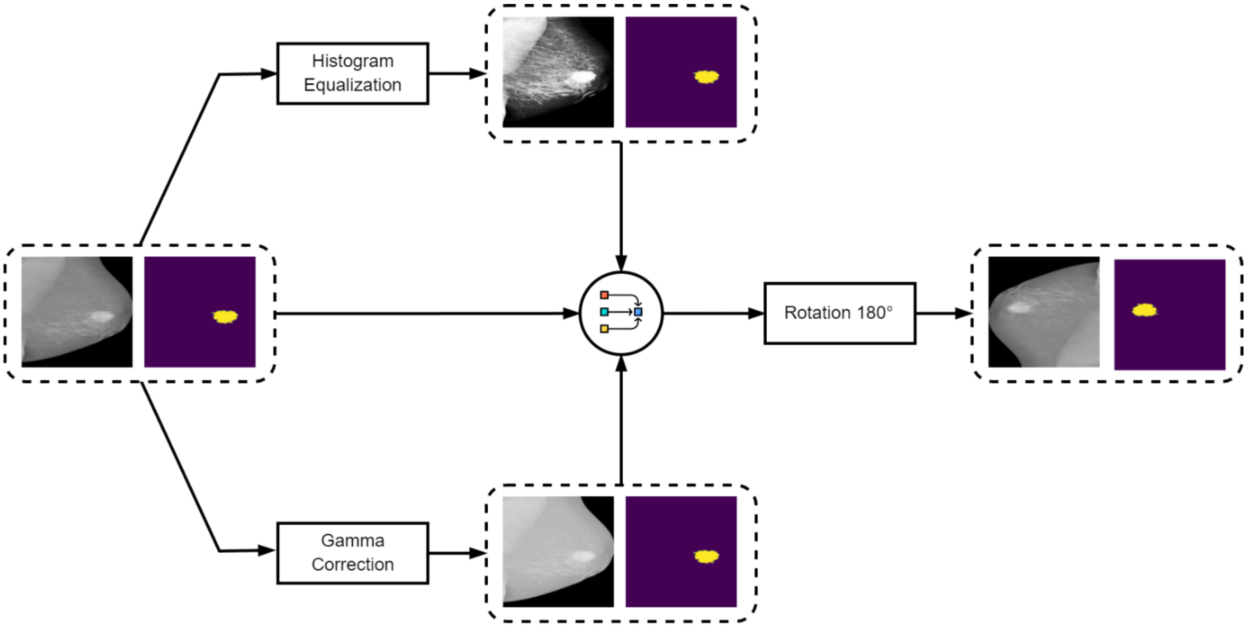
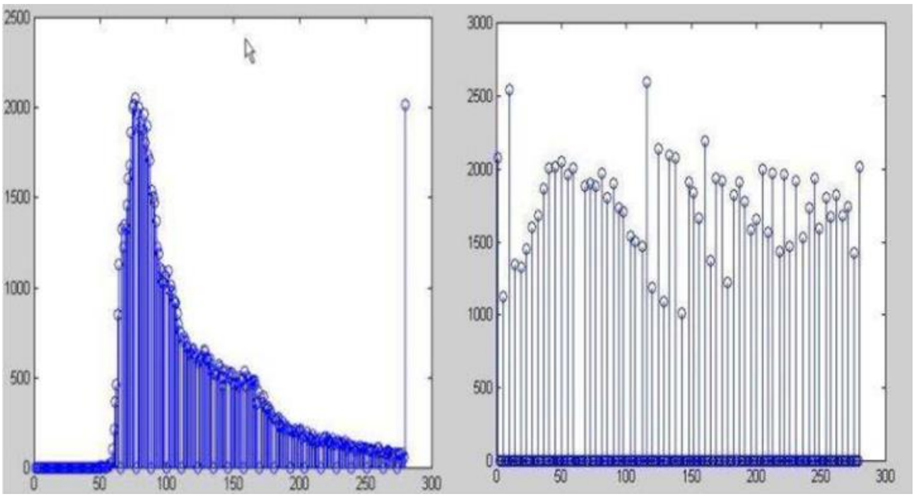


Figure 17. Presentation of the three augmentation techniques (histogram equalization (HE), gamma filter and 180-degree rotation) used for the enlargement of the initial dataset.

Numerous image enhancement techniques aim to reveal the hidden details of an image or to increase its contrast. To put it differently, these techniques make the images more useful by allowing the most important features of them to be visible through the expansion of their dynamic range. It is important to emphasize that image enhancement techniques solely improve the quality of an image without any raise at the information content within it [73]. The enhancement techniques are classified into two major categories, according to the data domain in which they are applied. These two groups include the spatial domain methods and the frequency domain methods. In the spatial domain methods, the enhancement operation is performed directly on the pixels of the image, while in the frequency domain methods modify the Fourier transform of an image [74].

Histogram equalization (HE) is one of the most important and common enhancement techniques in the spatial domain group, due to its simplicity and comparatively better performance on almost all different types of images [75, 76, 77]. HE is an effective and easy-to-implement computer image processing method that enhances the contrast in the input image, especially in cases where the information of the image is represented by close contrast values, by using its histogram [77, 78, 79]. The histogram of an image is a plot that displays the frequency of each gray-level appearance and provides information about its pixel distribution. These gray levels range between 0 (black) to 255 (white) [80]. Put differently, HE accomplishes that contrast enhancement by modifying the gray levels of an image based on its probability distribution function and stretching the dynamic range of their distribution, with the aim to improve visual effects on the output image [79, 81]. Therefore, HE enables the lower contrast regions to gain higher contrast, by spreading out the most prevalent intensity values, and thus the distribution of intensities that are presented on the histogram can be improved [79]. Figure 18 depicts the change of the histogram of a random image, before and after the HE technique. However, in some specific cases, there are some disadvantages to using the HE technique, due to the tendency to over enhance inside the situations and cause unwanted noise existing [77]. The HE technique has optimal results when used in images that have either very bright or very dark backgrounds. It is noted that HE may generate an unrealistic result in photographs, but it has been proven a very useful technique for scientific images, such as thermal, satellite, or x-ray images, etc. [79]. Figure 19 shows an indicative example of the application of the technique.

A histogram equalization technique can be categorized into one of two groups: local histogram equalization techniques and global histogram equalization techniques. Local HE methods tend to have better enhancing effects but lack processing speed in comparison to global HE techniques [81]. There are several different histogram equalization (HE) techniques with their own advantages and disadvantages, with some of them being classical histogram equalization (CHE), dualistic sub image histogram equalization (DSIHE), background brightness preserving histogram equalization (BBPHE), and dynamic histogram equalization (DHE). The selection of the suitable HE is contingent upon both the specific task and the nature of the dataset. For this study, the selected one was the CHE, which is a global operation, implying that the equalization is applied to the whole image.



(a) *(b)*

Figure 18. Indicative presentation of the original histogram of a random image (a) and its histogram after the HE technique (b). Source: [82].

Function 4 shows the probability density function $p(X_k)$ for an image X, where $k = 0, 1, \dots, L-1$ and represents the intensity level, n^k the number of times that the intensity level (X_k) appears in the input image X, n the total number of samples/pixels in the input image, and X_k the value of intensity level k.

$$p(X_k) = \frac{n^k}{n}, \text{ where } k = 0, 1, \dots, L-1$$

Function 4. Probability density function $p(X_k)$.

Based on the function above, the cumulative density function $c(X)$ is presented in the Function 5 below, where $X_k = x$, for $k = 0, 1, \dots, L-1$.

$$c(X) = \sum_{j=0}^{k} p(x_j)$$

Function 5. Cumulative density function $c(X)$.

It is important to be noted that $c(X_{L-1}) = 1$ by definition. HE is a method that utilizes the cumulative density function $c(X)$ as the transform function and through that maps the input image into the entire dynamic range (X_0, X_{L-1}). Function 6 presents the transform function f(x).

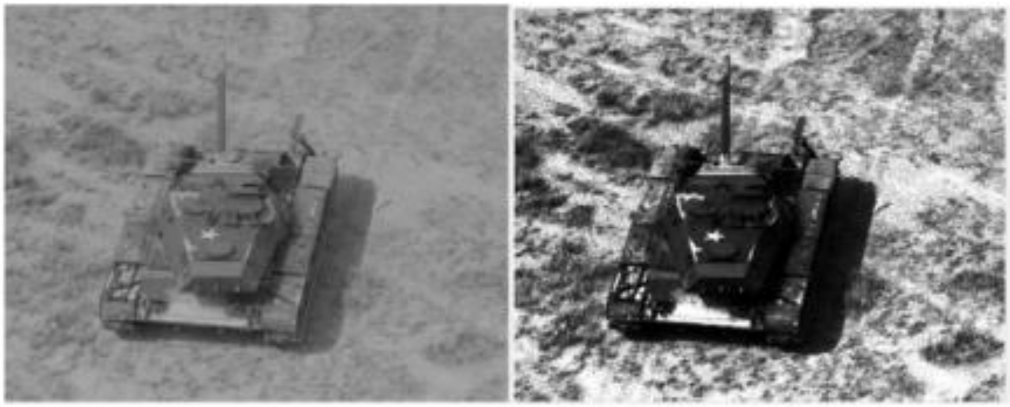
$$Y(i,j) = X_0 + (X_{L-1} - X_0)C(X_k)$$

Function 6.Transform function f(x).

Finally, the output image of the histogram equalization, Y = {Y (i, j)} can be formulated as shown in the Function 7 below:

$$Y = f(X) = \{f(X\ (i,j)|\forall\ X\ (i,j) \in X\}$$

Function 7. The function Y gives the output image of the HE.



(a) *(b)*

Figure 19. Indicative example of the Histogram Equalization technique, where (a) depicts the original image and (b) the HE image. Source: [73].

Gamma correction, also known as gamma filtering, is a widely popular and cost-effective technique for image enhancement, which changes the contrast level of an image by classifying its intensity into bright and dark [83, 84, 85]. Put differently, it controls the overall brightness of an image [86]. The result of the gamma correction method differs based on the γ coefficient, which can take on a range of positive real values. Theoretically, these values vary from 0 to infinity, but in the majority of the cases they range between 0.1 and 5. More analytically, values below 1 ($\gamma < 1$) result in a brightened image, values above 1 ($\gamma < 1$) in a darkened image, and in the cases where the γ is equal to 1 there is no change on the image [87, 88]. After experimenting with the gamma value, 0.5 was chosen for this research, since it was decided that the augmented images with this gamma value suited more with both the structure of the model and the nature of the dataset. The Function 8 computes the output image after the gamma correction (Si,j) at the (i,j) pixel location of the image. On that function, the h represents the pixel value of the original image at the (i,j) location, and γ the gamma correction value, which is determined according to the need of each case.

$$S_{i,j} = 255\left(\frac{h_{i,j}}{255}\right)^{1/\gamma}$$

Function 8. The result of gamma correction ($S_{i,j}$) at the (i,j) pixel location of the image.

Rotation is one of the most common and simple geometric data augmentation techniques, that rotates the original image either clockwise or counterclockwise. In other words, the rotation of an image can be done either right or left on an axis, between 1 and 359 degrees [89]. There are two types of rotation techniques, where the first sets a specified degree to produce different images from the original ones, while in the second method, the degrees for augmentation are randomly selected between a defined range [87]. Rotation is a special type of affine transformation [90]. The safety of rotation augmentations is strongly connected with the rotation degree parameter, which means that small rotations (between 1 and 20 degrees) tend to keep the label of the data, but in bigger rotations (more than 20 degrees) the label may no longer be preserved after the transformation [71]. The rotation selected belongs to the first category of rotation techniques, with the specific degree of the rotation being 180, which could be also described as a vertical flipping of the images. Flipping is another basic geometric transformation technique. It is a very widely used method in data augmentation and can be either horizontal or vertical, with the former being more used, because it is more realistic. However, in this study, the data are not direction sensitive (tumors can have weird and asymmetric shapes), so, vertical flipping (or rotation of 180 degrees) is considered a suitable choice [91]. In the Function 9 below, are presented the two functions (for both x and y coordinates) of the rotation operator, where (x_0, y_0) are the coordinates of the center of rotation in the original image, (x_1, y_1) are the coordinates of a presented element in the original image, (x_2, y_2) the coordinates of the same element in the output image and θ the angle of the rotation (in the clockwise rotations the angles are positive numbers, while in the counterclockwise rotations are is negative).

$$x_2 = cos(\theta)(x_1 - x_0) - sin(\theta)(y_1 - y_2) + x_0$$

$$y_2 = sin(\theta)(x_1 - x_0) - cos(\theta)(y_1 - y_2) + y_0$$

Function 9. The two transformations of the rotation operator.

Data Splitting

An important step in the development of an ANN that follows the data augmentations if needed, is the partitioning of the final dataset into smaller subsets. This process is called data splitting, and in other words, it divides the whole dataset into smaller portions, for cross-validatory purposes [92]. In that way, it is achieved a rigorous testing of the trained model’s generalization ability, which leads to a better final performance [93]. These sets are typically three (3:1 fashion); training set, validation set, the testing set, which are briefly explained below. Although, there are also data splitting approaches with different configurations, such as 2:1 and 4:1 fashion, but in this study the initial splitting (3:1) was selected [94]. The goal of data splitting into the aforementioned categories is to avoid overfitting the model, which is when it gives very accurate predictions on the training data during training, but not on new-unseen data [95].

- The training set is the portion of the whole dataset responsible for the model's training or in other words, is the dataset utilized for model fitting, allowing the model to observe, learn, and optimize its parameters [93]. The classification accuracy of the model is closely related to the size of the training set for various classifiers [96].
- The validation set is the sample of the whole dataset that assesses the performance of the task-specific predictor, while tuning the hyperparameters of the model [97]. Put differently, it provides an unbiased estimation of the performance that the model would potentially give if it were deployed for actual predictions in real-world situations, by using unseen data [98]. It is utilized to evaluate the given model and checks if the neural network has memorized the training data (which of course is not the desired outcome) or has actually learned some meaningful aspects of them, so that the model can be later used to an unseen, held-out testing set. Thus, the validation set aims to make the model generalized. Furthermore, is noted that the validation set is also known as dev set or development set, which makes sense, because as mentioned before this sample of the dataset plays an assistant role during the model's "development" phase. It is important to highlight that in the cases where the training and testing sets are small, the need for a robust evaluation is higher and plays a more vital role in the model’s performance [99, 100].
- The testing set is the portion of the whole dataset that is used to impartially assess the performance of a fully trained model that has undergone training and validation phases. Put another way, it is used for cross-validation during training to avoid over-fitting [93].

Given the fact that there does not seem to be clear guidance on which ratio is optimal, the exact ratio of the data splitting depends on characteristics of the dataset of each research, and mainly on the size of it. Some of the most commonly used ratios are 70:30 and 80:20, with the first number being the data used for training and the second for the evaluation and testing of the model [95, 101]. A very typical ratio for relatively small datasets tends to be 70:20:10 for training, validation, and testing sets, respectively. Hence, it was decided as the splitting ratio in both approaches (with the initial dataset and the augmented dataset). For the first approach, it is easily understood that the 70:20:10 ratio was selected due to the limited dataset, while the second approach followed the same splitting in order to maintain consistency and avoid introducing an additional variable. This decision aimed to achieve a more accurate and meaningful comparison between the two approaches. In Figure 20 is presented an indicative data splitting of the dataset in the three sets.

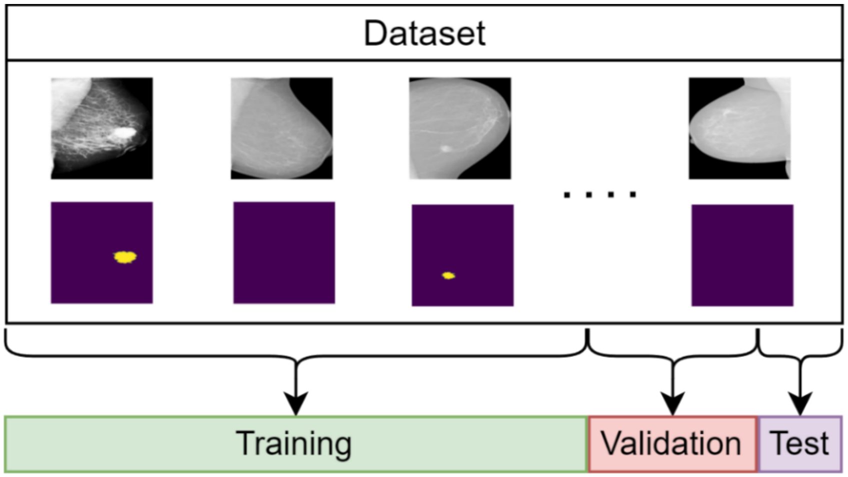


Figure 20. Indicative presentation of the data splitting.

Loss Function

Loss functions, also known as error functions or even cost functions, have a fundamental role in every ANN pipeline since they evaluate how effectively the algorithm is modeling the featured dataset. Put differently, a loss function helps the network determine how "inaccurate" it is and based on that information improve itself. Hence, a loss function is a measure of error, and the ultimate goal is to minimize it by iteratively adjusting the model's parameters. In the segmentation tasks, the role of a loss function is to assess how well the outcome of the model (predicted segmentation) matches the ground truth (masks given by medical experts) [102]. The choice of the loss function depends on the nature of the problem and the architecture of the neural network in use. In the existing literature, there have been proposed many different loss functions for semantic segmentation problems. These functions are divided into four types, which are distribution-based loss, region-based loss, boundary-based loss, and compounded loss [103].

Dice Coefficient (DC) alongside cross-entropy (CE) are considered the two most basic metrics in semantic segmentation tasks and they are typically used when training neural networks [103]. In this study, the former metric was selected to evaluate the model's performance, since it has been proved that in most cases it outperforms the CE in unbalanced medical image segmentation tasks, like the INbreast dataset [104]. It is also noted that the dice coefficient is a quantity that ranges from 0 to 1 and the goal is to have the highest value. The dice loss function, proposed by Milletari et al. [105], is based on the DC, given the fact that it can directly optimize it. It is a measure of overlap between the ground truth and the proposed segmentation of the network, considering each pixel of the images independently. It is commonly utilized to assess segmentation performance when the ground truth is available [106]. It belongs in the category of region-based loss functions and is a widely used metric in the computer vision community that calculates and evaluates the similarity that two images share [103, 104]. Region-based functions endeavor to decrease the mismatch (or in other words to increase the overlap regions) between ground truth and predicted segmentation, with the fundamental function being dice loss [102]. In Function 10 that follows, is presented the 2-class variant of the dice loss, denoted as DL_2. It is highlighted that the number ϵ is added in the numerator and denominator in order to avoid the function being undefined in edge case scenarios, which simply means to avoid division by 0 [107].

$$DL_2 = 1 - \frac{\sum_{n=1}^{N} p_n r_n + \epsilon}{\Sigma_{n=1}^{N} p_n + r_n + \epsilon} - \frac{\sum_{n=1}^{N} (1 - p_n)\,(1 - r_n) + \epsilon}{\Sigma_{n=1}^{N} 2 - p_n - r_n + \epsilon}$$

Function 10. Dice loss function, where r_n represents the ground truth binary mask and p_n the predicted by the neural network binary mask.

U-net

In 2015, researchers at the University of Freiburg, Germany, led by Olaf Ronneberger, developed a convolutional neural network architecture called U-Net for segmenting biomedical images [66]. CNNs are a subset of artificial neural networks designed in a way to process one or more dimensional data, mainly deployed for image analysis tasks, but they can be applied to various types of data. The name of this specific type of network is derived from the convolution layers, which are the fundamental part of it. While an ANN enlarges the volume of trainable parameters, a CNN focuses on the extraction of the key features from the data through the convolution operation. The advantage of a CNN stands in its capacity to concurrently learn numerous features and generate a feature map. This is possible through an element-wise multiplication (dot product) of the input with a kernel (filter), and then applying a summation of the results to a single value (scalar product) [70]. Today, the U-Net is widely recognized as one of the most effective approaches to semantic segmentation tasks. Unlike some other neural network models, U-Net is designed in a way that can learn from a smaller number of training samples. In the Figure 21 below is depicted the architecture of the U-net model, where each blue box is a multi-channel feature map with the number of its channels indicated at the top of the box, the dimensions (x-y size) of these feature maps are specified at the lower left corner of each box, white boxes symbolize duplicated feature maps, and lastly, the arrows illustrate various operations.

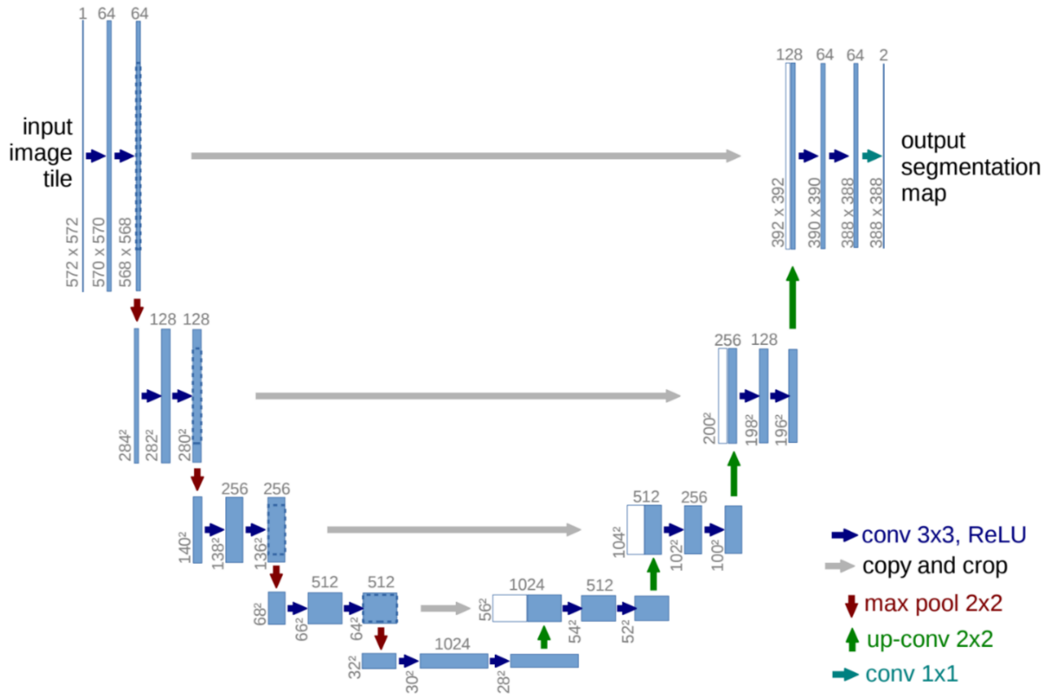


Figure 21. Architecture of the U-net model (example of 572x572 pixels and 1 channel of the input image tile). Source [66].

The name of this unique neural network is U-net, which derives from its U-shaped architecture. It consists of convolutional layers and is divided into two networks: the encoder network, comprising four encoder blocks, and the decoder network, comprising four decoder blocks. These two networks are connected via a bridge, forming the distinctive U-shaped structure of the model. What sets U-net apart from any other CNN and makes it particularly effective for image segmentation tasks is the combination of the skip connections and the detection network.

The encoder network, often referred to as the contracting network (left side of the figure above), employs a sequence of four encoder blocks to extract an abstract representation of the input image and endeavor to understand the features and details contained within the image. Each one of these blocks contains two 3x3 convolutional layers (also called unpadded convolutions), followed by a ReLU activation function. That function introduces non-linear characteristics within the network, resulting in an improved generalization of the training data. Subsequent to that, the output of the ReLU is subjected to a max pooling layer using a kernel size of 2x2. This pooling operation results in a scaling down of the spatial dimensions (height and width) in the feature maps, by half. This feature provides a notable advantage to the model as it reduces the computational cost during training, leading to a decrease in the number of trainable parameters. Thus, at each down-sampling step, the feature channels are doubled. Furthermore, the result of the ReLU function serves as a skip connection for the corresponding decoder block of the expansive network in the subsequent stages of the process, as described below.

The skip connections are visually represented by gray arrows in the model's architecture in the figure above, and they play a vital role in the model by providing essential features to the decoder network, that might have been lost due to the depth of the network. The skip connections result in the generation of more accurate semantic features and act as shortcut connections that help the indirect flow of gradients to the earlier layers with zero degradation. In other words, the existence of these skip connections ensures a better flow of gradients during backpropagation, which aids in a better representation by the network. Between the encoder and decoder network exists a bottleneck layer, that is also called a bridge. It involves two 3x3 convolutions, where a ReLU activation function follows each of them. The output of the bridge is the final feature map representation.

The decoder network, also known as the expansive network (right side of the figure above), is used to take the abstract representation from the bottleneck layer and generate a semantic segmentation mask with the contribution of skip connections. Specifically, the decoder block starts with transpose convolution (or else known as up-convolution) of 2x2 and is followed by the corresponding skip connection feature map from the encoder network. In addition, after that, there are two 3x3 convolutions, where a ReLU activation function follows each of them. Finally, the last decoder's output undergoes a 1x1 convolution with sigmoid activation. That sigmoid activation gives the segmentation mask the pixel-wise classification. In total, the U-net model consists of 23 convolutional layers, including eighteen convolutions 3x3, four transpose convolutions 2x2, and one convolution 1x1.

Optimization

In machine learning tasks, a model acquires knowledge from given data, by learning how to minimize a specified loss function. This function, in most cases, is derived from the difference between the ground truth (true value) and the result (output value) computed by the ANN [108, 109, 110]. Consequently, the specific form of the loss function can be influenced by various factors, such as the volume of training data, the existence of non-linear activation function in ANN or not, and the overall structure of ANN. The aforementioned choices lead to both a singularity and a local minimum, in the lost function used for the network. A common feature that all the different functions in the literature (either singularity or local minimum) share is that their first derivative value is equal to zero [108].

Optimization is considered a challenge for the majority of machine learning tasks, as well as in several other fields, including operations research, engineering, and statistics [111]. The fundamental process of machine learning revolves around solving optimization problems. In a ML network, its weight parameters and configurations are initialized and optimized by an optimization algorithm, until the selected loss function approaches a minimum value, or the accuracy of the model approaches a maximum value [112]. It is therefore a vitally important process, as it maximizes the model's performance and minimizes the misclassification errors. Therefore, in order to achieve the model being as effective as possible, a suitable optimizer for each specific network has to be chosen.

Based on the level of derivative information used and the underlying principles of the optimization approach, an optimization algorithm is categorized in a group of algorithms with similar features. Such groups are the first-order algorithms, the second-order algorithms, and the meta-heuristic algorithms [113, 114]. Given the fact that the optimization algorithm selected for this research belongs to the first category, this is the one that has been analyzed in depth. However, indicatively is noted that the second-order algorithms use both the first derivative, also known as gradient and the second-order derivative, also called Hessian, in order to minimize the Loss function [114, 115, 116]. The Hessian represents a matrix of second-order partial derivatives [117]. Respectively, meta-heuristic algorithms are not strictly tied to the derivatives of the objective function. They are often population-based, or trajectory-based and tend to explore the solution space more globally [118].

First-order optimization algorithms have been utilized for the training of several machine learning networks. These algorithms use only the first derivative (gradient) information of the objective function [119]. The aim of first-order optimization algorithms focuses on utilizing information related to function values and gradients or subgradients, excluding Hessians [120, 121]. In the literature, there are different categories of optimizers, including gradient descent-based learning algorithms, adaptive gradient-based learning algorithms, momentum adaptive gradient-based learning algorithms, etc. [122].

One of the first-order introduced in the literature optimization algorithms is the gradient descent (GD). GD is an optimization method that uses a fixed-point method to drive the first derivative of the loss function to the minimum value, i.e. zero. This method can be reliable in simple ANNs, but not in complex ones, where it often finds several difficulties [108]. GD is applied to minimize some functions by iteratively moving in the direction of the steepest descent as defined by the negative of the gradient. Various types of GD exist, with some of them being the stochastic gradient batch (vanilla), gradient descent, and mini-batch gradient descent. A more rapid and memory-efficient approach to the GD method is the SGD [108, 122]. SGD is an iterative method for learning model parameters by using the negative gradient of the loss function with a mini-batch extracted from data. It surpasses the GD method because instead of using the entire dataset, it processes individual training examples in each iteration. Based on the SGD method several new adaptive methods have been found by researchers and proposed in the scientific society. A characteristic of an adaptive method is that it has the ability to use different learning rates for each one of the parameters [118, 122].

Adagrad, Adadelta, and RMSProp belong in the category of adaptive gradient-based learning algorithms since they use adaptive learning rates in order to update their variables [122]. AdaGrad was the starter of adaptive methods. It is a stochastic optimization method that permits the learning rate to adapt depending on the parameters. Hence, it performs small updates for frequent parameters and big updates for infrequent parameters. It is noted that Adagrad has been proven to work well with sparse gradients [122, 123, 124, 125]. Adagrad adjusts the learning rate for each parameter individually at each time step based on the previously calculated gradients for that specific parameter. The main advantage of AdaGrad is its ability to eliminate the necessity for manual tuning of the learning rate, with the majority of the cases being set at the default value, viz. of 0.01 [123, 126]. However, a fundamental drawback of this algorithm relies in the accumulation of the squared gradients in the denominator. Given the fact that every added term is positive, the cumulative sum continually grows during training, causing the learning rate to shrink and finally become insignificantly small [123, 124, 126, 127]. Following Adagrad, which as noted previously was the first adaptive method, several other adaptive methods have been presented to solve the problems and fill the gaps that are created from the weaknesses of the previously proposed methods. AdaDelta and RMSProp have converted the sum of gradient vectors from AdaGrad into an average [128]. More analytically, the former can be described as an expansion of the AdaGrad method that leans towards eliminating the decay caused by the learning rate [122]. The Adadelta approach limits the window of collected past gradients to a few settled estimate weights instead of pressing the entire squared first-order derivative by using a decay average [122, 129, 130]. Respectively, the latter can be described as an extension of the AdaGrad method, which uses a moving average of the partial derivatives instead of the sum in the calculation of the η for each instance [114, 122]. RMSProp is capable of computing learning rates for every parameter. Additionally, independent calculations and packaging for each parameter are possible. It is noted that RMSProp performs well in on-line and non-stationary settings [122, 126].

The Adam optimizer is a method for efficient stochastic optimization that only requires first-order gradients with little memory requirement, and it belongs in the category of momentum and adaptive gradient-based learning algorithms [131]. Adam can be defined as another approach that calculates the adaptive learning rates for every parameter [108, 122]. It is designed in a way that combines the advantages of the two previously explained algorithms, AdaGrad and RMSProp [122, 131]. Its name is derived from the adaptive moment estimation, and it is the most widely adopted approach based on the GD method and the momentum method. More specifically, it is an SGD algorithm that utilizes the momentum concept to efficiently reach the global minimum of the loss function. This assists in effectively adjusting the learning rate for each parameter, hastening the time needed for convergence to the minimum [108]. Adam is designed for machine learning tasks with large datasets or high-dimensional parameter spaces and is considered to be a robust and suitable choice for various non-convex optimization problems in the area of machine learning [131]. Adam is still one of the most commonly used algorithms for the training of deep neural networks, because of its adaptive learning rate and excellent performance [131, 128]. It provides several benefits, including the magnitudes of parameter updates are invariant to rescaling of the gradient, its stepsizes are approximately bounded by the stepsize hyperparameter, independence from a stationary objective, compatibility with sparse gradients, and the natural incorporation of a form of step size annealing [131]. Something challenging in the Adam method is the learning rate, which is selected by the researcher. If the selected learning rate is too small, there will be zero or almost zero progress, while if it is too big the solution will fluctuate, or it may even diverge (in the extreme scenarios).

Overall, the Adam optimizer is known for its simplicity in the implementation, as well as the low memory requirements [131]. The Adam method computes individual adaptive learning rates for various parameters from estimates of the first and second-moment values (gradients) [128]. Put differently, Adam utilizes the sum of the gradient values multiplied by the weights calculated in the past, which is also known as the first momentum. Respectively, the sum of squares of the gradient is calculated in a similar way and the result is known as second momentum. Finally, the ratio between the first and second momentum values is computed, and the minimum value is searched for, according to the ratio [108].

Below, in the Function 11 and Function 12 is presented how the first and second momentum is obtained, respectively. Additionally, in the Function 13 is presented the weight update of the Adam method.

$$m_i = \beta_1 m_{i-1} + (1 - \beta_1)\frac{\partial C}{\partial w}$$

Function 11. Adam update for the first momentum estimate (gradient).

In the Function 12 that follows, is presented the second momentum.

$$v_i = \beta_2 v_{i-1} + (1 - \beta_2)\left(\frac{\partial C}{\partial w}\right)^2$$

Function 12. Adam update for the second momentum estimate (squared gradient).

$$w_{i+1} = w_i - \eta \frac{\widehat{m}_i}{\sqrt{\hat{v}_i + \epsilon}}$$

Function 13. Adam final step of the weight update w.

where $\widehat{m}_i = m_i/(1 - \beta_1)$ and $\hat{v}_i = v_i/(1 - \beta_2)$.

Metrics used

Evaluation metrics play a vital role in estimating the performance of an ANN, by helping the researchers measure how effectively the model performs its intended task [132]. The evaluation metrics available are plentiful and the appropriate selection of one is contingent upon several factors, including the distinct characteristics of the neural network in use, the inherent properties of the dataset, the problem/task, etc. Indicatively some of them are accuracy, precision, recall, F1 score, Matthews correlation coefficient (MCC), and mean absolute error (MAE).

Some researchers consider that accuracy is the most adequate evaluation metric. Accuracy, as defined by the ratio of the correctly classified samples to the total number of samples, is reliable in both cases with two labels and in multiclass cases, viz., when labels are more than two [133, 135]. However, it cannot be considered a reliable evaluation metric in problems where the dataset in use is unbalanced, which means that the number of samples in one class is significantly higher than others. That happens because accuracy provides an over-optimistic estimation of the classifier's ability on the majority class [134, 136, 137]. In these cases, is preferred the Matthews correlation coefficient (MCC), which is an effective metric in cases with an unbalanced dataset, that stems from the definition of the phi coefficient (ϕ) [135, 138]. However, there are some extreme cases where the MCC is not a reliable metric, given the fact that it might display large fluctuations, because of the imbalanced outcomes in the classification [135, 139].

The harmonic mean of precision and recall was introduced in the literature of statistical ecology by Dice [140] and Sørensen [141] independently in 1948, and later rediscovered in the 1970s in the information theory by van Rijsbergen [142, 143]. However, the current notation of the F1 measure was for the first time introduced in 1992 [143]. In fact, during the decade of 1990, F1 gained a lot of popularity in the machine learning community, to such a point that it was re-introduced in the literature as a novel evaluation metric [133, 135]. Nowadays, it is a widely used metric in several areas of machine learning, both in binary and multiclass cases. Specifically, the F1 score finds extensive utility in classification problems, information retrieval, and NLP tasks.

Dubey and Tatar state that the F1 score and MCC provide more realistic estimates of real-world model performance [145]. Undoubtedly, these two metrics are some of the most common and reliable performance measures, but they differ in mainly two features. The first is that the F1 score gets affected by class redefining (asymmetric between the classes), while the MCC is invariant if the negative class is renamed as positive, and vice versa. However, it is highlighted that this problem for the F1 score can be solved by simply extending the macro/micro averaging procedure to the binary case itself, which is invariant for class redefining and its behavior reminds the MCC. Therefore, the F1 score would be defined in both positive and negative classes and then average the two values, for the macro, and using the average sensitivity and average precision values, for the micro. The second is that the F1 score is independent of the correctly classified negative samples (also named true negatives). However, the micro/macro average F1 remains biased, since it has not yet been accepted by the scientific community as a standard practice [133].

For the evaluation of the model's performance, the F1 score was preferred as the most suitable metric. The F1 score is the harmonic mean of precision and recall. This implies the penalization of the extreme values of its components (precision and recall), if any. It is asymmetric between the classes, which is translated in the dependence on which class is defined as the positive and which is the negative [132]. Put differently, if given complemented predictions and true labels, the F1 measure differs significantly. An example of that would be a large positive class and a classifier biased towards this majority. In this situation, the F1 score would be high, given the fact that it is proportional to true positives. However, in the redefining of the class labels, the outcome would be completely different, with neither the data nor the relative class distribution having changed. More specifically, if the class labels changed, resulting in the negative class being the majority of the data, and the classifier is biased towards the negative class, the F1 score would be low. Moreover, the F1 score is a quantity that ranges from 0 to 1, with the highest value of 1 indicating maximum precision and recall values, while the lowest value of 0 represents that one of the two (or both) precision or recall is zero [146]. It is worth mentioning that the F1 score tends to be closer to the lowest number of its components (precision and recall). Therefore, a high F1 score requires high values in both, in order to ensure a balanced performance. The F1 score is alternatively defined as a function of counts of true positives (TP), false positives (FP), and false negatives (FN). In Function 14 is presented the calculation of the F1 score, with both definitions [135].

$$F_1 = 2 * \frac{Precision * Recall}{Presicion + Recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

Function 14. Calculation of the F1 score.

It is easily visible that the second definition does not present the number of true negatives (TN) and that the true positives are considered as twice important as the rest [135, 147]. In this specific study that focuses solely on positive outputs, F1 is considered a suitable metric for the measurement of performance and that is why it got selected.

Experimental Procedure

As explained in the methodology section, there are two distinct approaches to this study. The first approach was implemented with only the initial dataset, while the second one was with a larger dataset that occurred through the data augmentation phase. It is reminded that the first approach includes 410 images (mammograms and masks), while the second includes 2460.

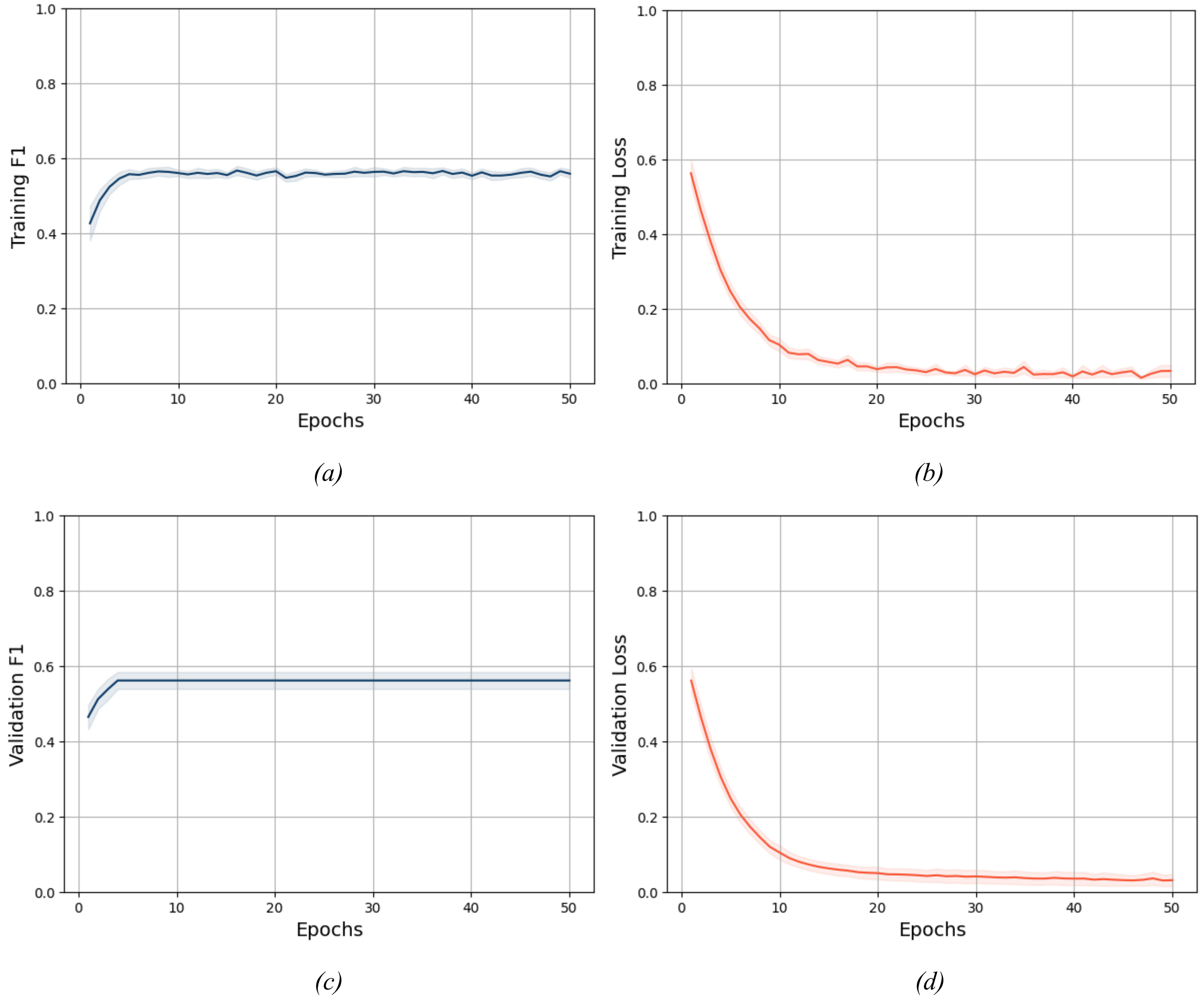
The training of the model on each one of the approaches has been decided to be divided into four experimental categories, each comprising 20 experimenter repetitions. The four experimental categories differ in the epochs (two have 50 and two have 100 epochs) and at the learning rate of the optimizer (two have lr=0.01 and two have lr=0.001). The experimental categories have the same experimental conditions in both approaches. This division is crucial for gaining insights into the weak points of the network. In essence, this study consists of a total of 2 approaches x 4 experimental categories (or simply experiments) x 20 experimenter repetitions = 160 times that the model was trained, with the goal of a more robust and complete understanding of the neural network's weaknesses or constraints across different scenarios as well as finding the combination that has the best performance.

Overall, the above experimental design allows for a deep comprehensive exploration of the model's performance under different training conditions and dataset, which goals into identifying its strengths and weaknesses under varying levels of training duration and learning rates. It is noted that, this is a very common practice in machine learning to conduct such experiments to fine-tune model parameters and understand their behavior with the aim of better understanding the model to improve it.

First Approach

Below are presented the four figures of the first approach, depicting the model's results in the form of diagrams, including F1 score (diagrams at the left) and loss function (diagrams at the right), for both the training and validation phases of each of the four different experiments conducted, as described above. Following this, the diagrams will be briefly analyzed, and conclusions will be drawn regarding the model. These findings will guide subsequent actions to enhance the model in the future. Additionally, the F1 score for the testing set is presented for each one of the experiments.

Firstly, there are presented in the Figure 22 the results of the experiment with a learning rate of 0.001 and 50 epochs are presented. This experiment had a testing **F1 score** equal to **0.5899519274632136**.



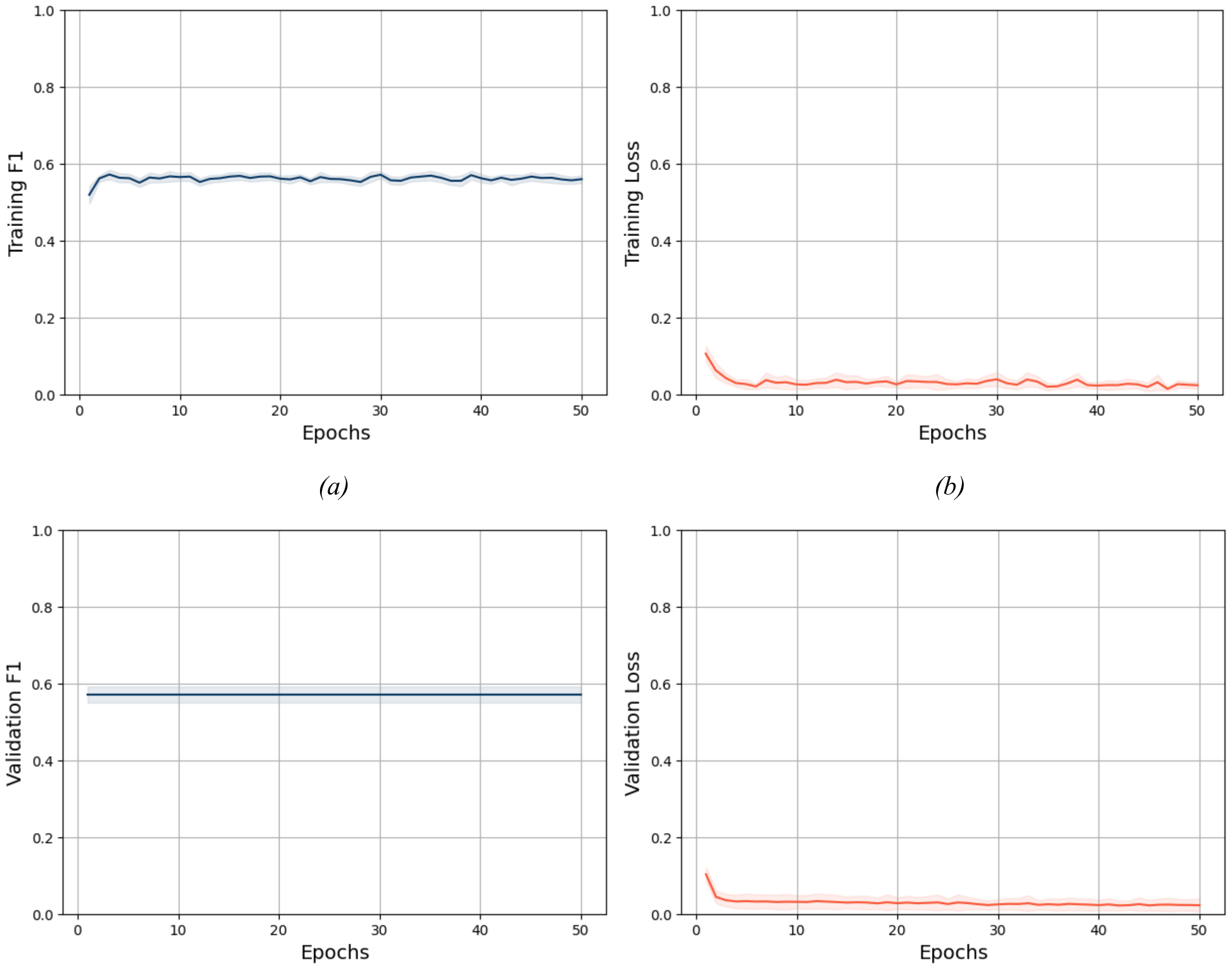
(a) *(b)* *(c)* *(d)*

Figure 22. F1 score and Loss of the first experiment (learning rate = 0.001 and 50 epochs). (a) Shows the F1 score of the training set, (b) the Loss of the training set, (c) the F1 score of the validation set and (d) the loss of the validation set.

Observing Figure 22, various conclusions for the model can be drawn regarding the first experiment conducted. Specifically, it becomes evident through the (a) chart that the model showed a small improvement in the F1 score for the training dataset during the first five epochs. However, from the 6th epoch onwards until the last (50th), it did not show any substantial variation. A similar pattern was also followed at the F1 score of the validation set, (c) chart, with the notable difference being that the learning process came to a complete halt after the 4th epoch of the experiment, as the F1 score line remained flat thereafter. It is worth noting that in both cases, the F1 score commenced at approximately 0.43-0.47 and reached its maximum just below 0.60. Finally, it is noteworthy that the variance of chart (c) is significantly greater than the chart's (a).

Regarding charts (b) and (d), a notable resemblance between them is also discernible. In both cases, they start from a value of approximately 0.57, experience a quick decline to 0.1 within the initial 10 epochs, and ultimately converge at a value of 0.04 by the experiment's conclusion (50th epoch). The only distinctions observed between them are that in the training set loss chart, the variance is considerably smaller than the loss chart of the validation set. Additionally, chart (d) exhibits a smoother line, devoid of abrupt fluctuations between consecutive epochs.

Subsequently, there are depicted in the Figure 23 outcomes of the experiment conducted with a learning rate of 0.01 and 50 epochs. This experiment achieved a testing **F1 score** of **0.5360531901319822**.



(a) *(b)*

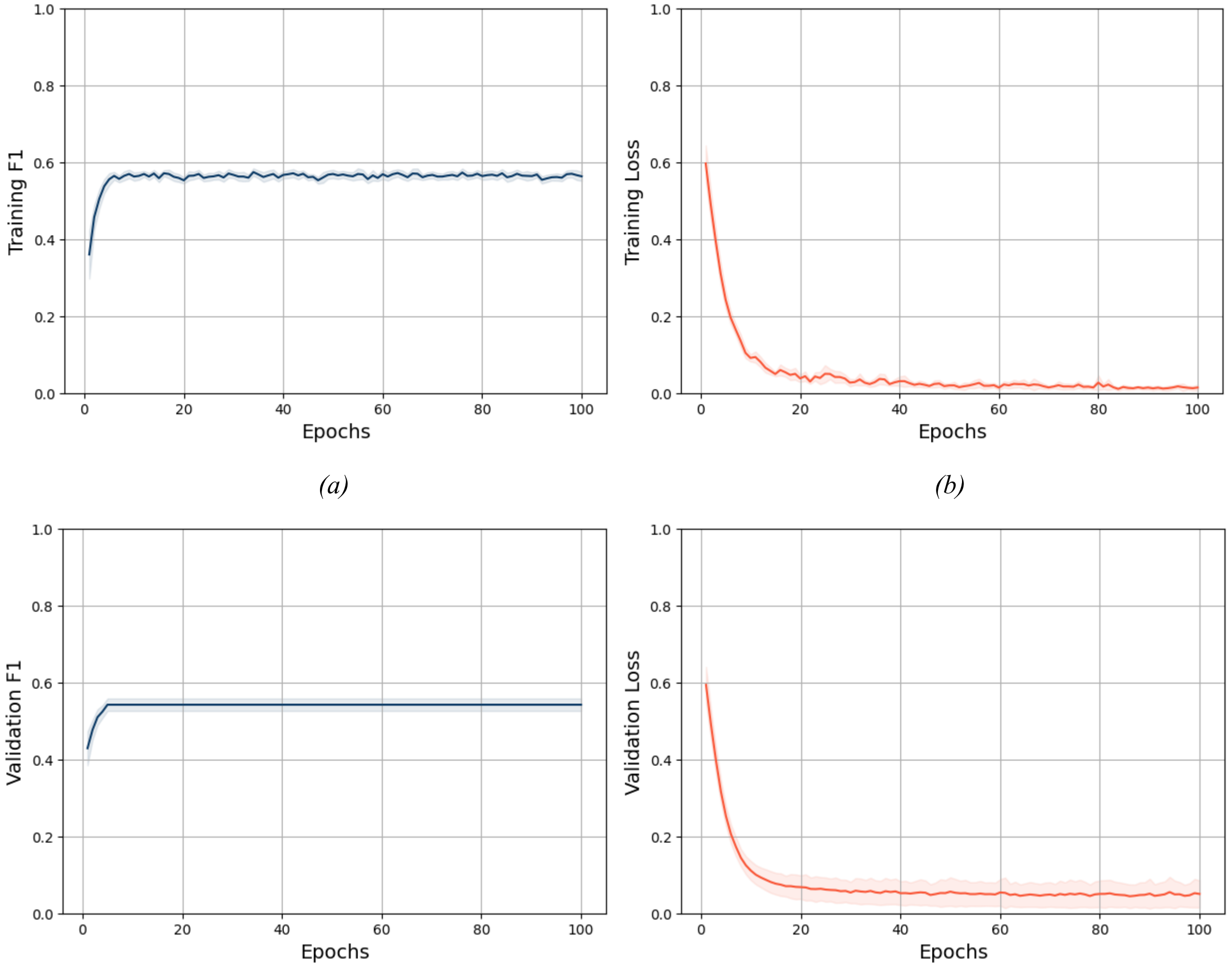
(c) *(d)*

Figure 23. F1 score and Loss of the second experiment (learning rate = 0.01 and 50 epochs). (a) Shows the F1 score of the training set, (b) the Loss of the training set, (c) the F1 score of the validation set and (d) the loss of the validation set.

Observing and analyzing the above figure, many crucial insights about the model can be deduced concerning the second experiment. More analytically, it becomes clear by the (a) chart that the model showed a slight increase at the F1 score of the training set during the first two epochs, as it initiated at approximately 0.54 and reached 0.57. However, starting from the 3rd epoch and continuing until the final (50th), it did not demonstrate any further learning and it stayed relatively constant throughout the remainder of the experiment. Regarding the F1 score of the validation set, as indicated in chart (c), it remained consistently flat across all epochs, signifying that the learning process did not yield positive results in this experiment. The value hovered around 0.58 throughout the entire duration of the experiment. Similar to the first experiment, the variance of chart (c) is visibly greater than that of chart (a).

When considering charts (b) and (d), a significant similarity is evident. In both cases, they initiate at a value of approximately 0.11, experience a quick decrease to 0.07 for chart (b) and 0.05 for chart (d) within the first 2 epochs, and ultimately converge at a value of 0.03 by the conclusion of the experiment (50th epoch). Similarly to the first experiment, notable distinctions between these two charts, include the smaller variance at the loss chart in the training set compared to the loss chart of the validation set, as well as the smoother curve exhibited in chart (d), which lacks abrupt fluctuations between consecutive epochs.

Following this there are presented in the Figure 24 the findings from the experiment involving a learning rate of 0.001 and 100 epochs. In this experiment, the achieved testing **F1 score** was **0.5440800420939922**.



(a) *(b)*

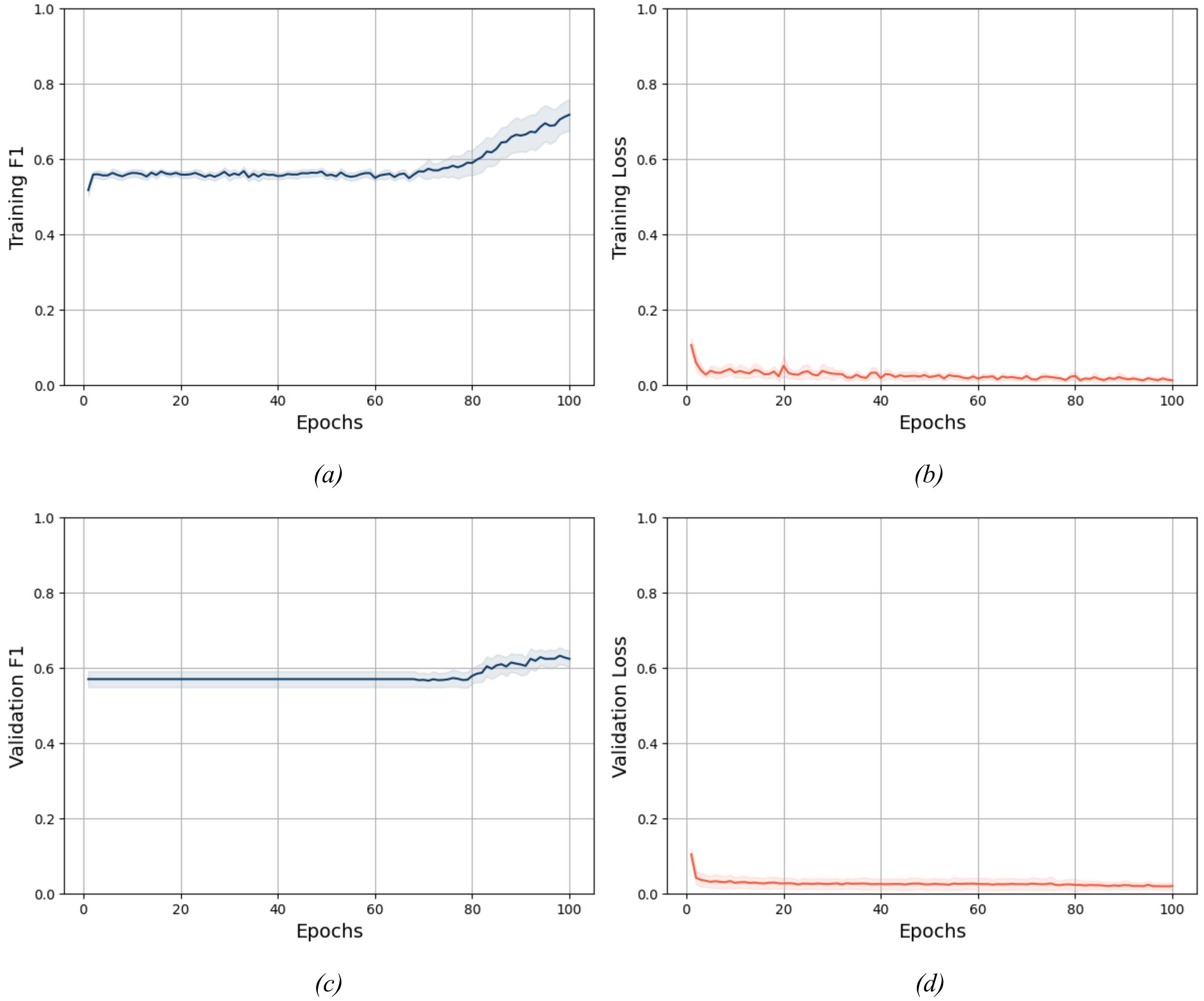
(c) *(d)*

Figure 24. F1 score and Loss of the third experiment (learning rate = 0.001 and 100 epochs). (a) Shows the F1 score of the training set, (b) the Loss of the training set, (c) the F1 score of the validation set and (d) the loss of the validation set.

Observing Figure 24, numerous crucial observations regarding the third experiment of this research become apparent. Before delving deeper into the analysis, it is noteworthy to mention that the charts in this experiment share a similar structure as those in the first experiment (Figure 22). Specifically, the F1 score at the training set, chart (a), exhibits a remarkable improvement during the initial six epochs. Nevertheless, commencing from the 6th epoch and continuing until the culmination of the experiment (100th), it did not display significant progress. More analytically, it began at approximately 0.35-36 and reached the values of 0.57-0.58 in the sixth epoch, remaining relatively stable throughout the rest of the experiment, with minimal variation. A similar pattern was also followed at the F1 score of the validation set, chart (c), with the notable difference being that the learning process failed to exhibit any discernible progress after the fifth epoch of the experiment, as the F1 score line remained flat thereafter. The F1 score started at approximately 0.43 and after the fourth epoch remained consistently around the area of 0.55 for the entire duration of the experiment. Once again, the variance of chart (c) is greater than that of chart (a).

Regarding charts (b) and (d), a notable resemblance between them is also discernible. In both cases, they start from a value of 0.60, experience a quick decline to 0.1 within the initial 10 epochs, and ultimately converge at a value of 0.02 for chart (b) and 0.06 for chart (d) by the experiment's conclusion (50th epoch). Following a pattern akin to the first experiment, distinguishing factors between the two charts (b) and (d), are the smaller variance at the loss chart in the training set compared to the loss chart of the validation set, as well as the smoother curve exhibited in chart (d), which lacks abrupt fluctuations between consecutive epochs.

Lastly, there are presented in the Figure 25 the results of the experiment that was implemented with a learning rate of 0.01 and 100 epochs, which achieved a testing **F1 score** of **0.6429447333017986**.



(a) *(b)*

(c) *(d)*

Figure 25. F1 score and Loss of the fourth experiment (learning rate = 0.01 and 100 epochs). (a) Shows the F1 score of the training set, (b) the Loss of the training set, (c) the F1 score of the validation set and (d) the loss of the validation set.

Considering the figure above and conducting an analysis, numerous critical observations concerning the final experiment within this research emerge. It is readily apparent that charts (b) and (d) exhibit a similar pattern to those of the second experiment (Figure 23), whereas, in contrast, charts (a) and (c) are entirely distinct from what was previously discussed regarding the first three experiments in the study. More analytically, the F1 score at the training set, chart (a), exhibits a slight rise at the initial two epochs. Following that, commencing from the 3rd and continuing until the 68th epoch, it failed to exhibit any discernible progress. Subsequently though, it is observed a significant improvement of the F1 score of the model, extending until the experiment's conclusion (100th epoch). It started at around 0.52 and reached the values of 0.56-0.57 in the third epoch, remaining relatively stable until the 68th epoch, with minimal variation. Thereafter, it exhibited a substantial rise until the value of approximately 0.72. A similar pattern was also observed in the F1 score of the validation set, chart (c) but the improvement was smaller, which is a sign of overfitting. Some of the differences include the fact that the learning process showed no progress in the initial sixty-eight epochs of the experiment, as the F1 score remained flat. Subsequently, the variance in chart (a) is greater than that the chart (c), contrary to what was observed in the charts of the F1 score of the previous three experiments. Particularly, the starting value commenced at approximately 0.57 and maintained that value until the 68th epoch of the experiment, where there was a notable increase, which reached as high as 0.63 at the end of the experiment (100th epoch).

Regarding charts (b) and (d), a notable resemblance between them is also discernible. Both charts, initiated at a value of approximately 0.11, experience a quick decrease to 0.03 within the first 4 epochs, and ultimately converge at a value of 0.02 for the chart (b) and 0.03 for the chart (d) by the conclusion of the experiment (50th epoch). In a manner consistent with the first experiment, the line exhibited in chart (d) is way smoother than the line in chart (b), which exhibits sharp fluctuations between consecutive epochs. Interestingly though, unlike the other experiments, it is observed that the loss chart of the validation set exhibits a smaller variance compared to the training's set chart.

First Approach Results - Analysis

The conclusions drawn from the analysis of the above diagrams, which represent the results of the first approach experiments conducted of the model, are particularly important for the next steps of the study, given the fact that based on them came the idea of the second approach. Thus, these conclusions serve as the basis for making changes and conducting new experiments aimed at improving the F1 score and reducing the model's error.

It is reported that, in order to evaluate which learning rate of the two is more suitable for the model, the four experiments have to be compared between each other in a way that have the rest conditions and configurations same, such as epochs and dataset. As the dataset remains consistent across all first approach experiments, comparisons are made between two pairs: the first-second experiment and the third-fourth experiment, both having the same number of epochs. Consequently, it is observed that the small (lr = 0.001) and big (lr = 0.01) learning rates do not yield consistent results in terms of F1 score and loss value at these two pairs. More specifically, in the first pair (50-epoch experiments) the small learning rate achieved the highest F1 score at the testing set, while the opposite occurred at the second pair (100-epoch experiments) where the highest F1 score at the testing set was achieved by the big learning rate. Concerning the loss values, in the first pair, both the lowest error value of the training and validation set are observed in the second experiment, viz. when the model was trained with the big learning rate. On the contrary, in the second pair, the lowest error value of the two sets was not achieved at the same experiment. The lowest error value of the training set was achieved from both the third and fourth experiments, while the lowest loss value of the validation set was achieved in the fourth experiment, in which the model was trained with the big learning rate. This observation highlights the sensitivity of the model's performance to the choice of learning rate and suggests that the optimal learning rate may change over the course of training, depending on the number of epochs. Hence, given the observed inconsistencies in performance, it remains unclear which learning rate is more suitable for the model. Further investigation into the dynamics of learning rates and their impact on model convergence is necessary to effectively fine-tune the training process. Additionally, it is worth noting that the big learning rate was significantly faster compared to the small one during the model’s training.

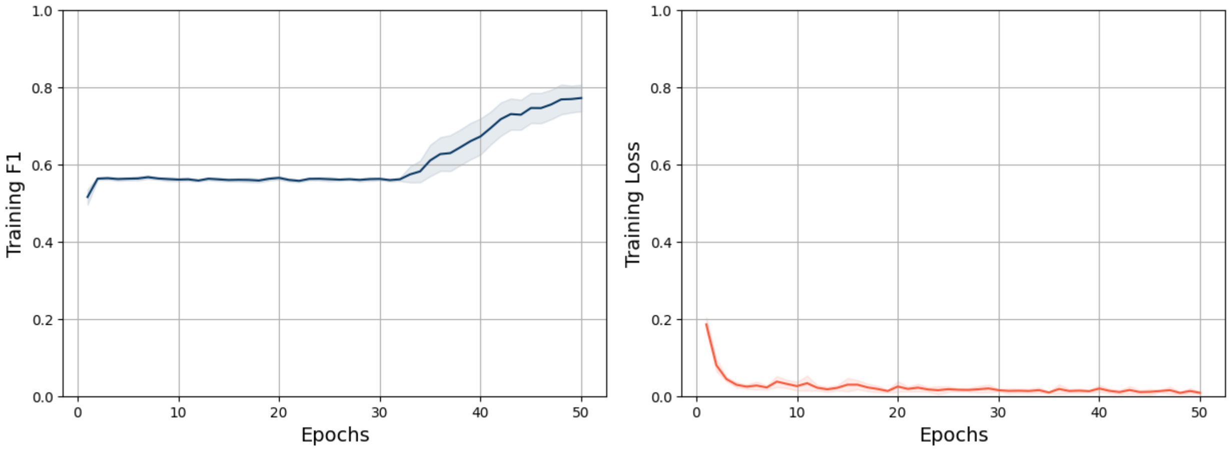
In general, the number of epochs in an ANN is a critical parameter. In this first approach, it is proved that the proposed network performs better in the fourth experiment, which is the 100-epoch experiment with the big learning rate, while on the contrary, in the rest three experiments the model did not achieve to learn from the data. To be more specific, the F1 score of the testing set in the first three experiments ranges from approximately 0.54 to 0.59, while in the fourth experiment, it rises to 0.64, indicating a small but important difference in performance. The F1 score of the testing and validation sets is not really beneficial for analysis, given the fact that in three out of four experiments the F1 line remained mainly flat with slight fluctuations, if any, around the area of 0.55-0.58. However, it is mentioned that in the most successful experiment of the four, the F1 score in the training and validation sets managed to reach up to 0.72 and 0.63, respectively. Thus, this suggests that the 100 epochs might be preferable for future experiments, since the model continues to "learn" beyond the initial 50 epochs, provided that the learning rate is correctly chosen. Additionally, it is noted that the improvement, or better to say the increase in the F1 score, was achieved after the 60th epoch, which indicates that under these configurations the model fits the dataset, but very late in the training procedure, resulting in a low performance. That suggests that the training data might not be enough to train the model for effective generalization, which points out the way to implementation of data augmentation techniques. Additionally, it noted that as expected the training of the 100-epoch experiments is slower than the 50-epoch ones.

The augmentation of the dataset is considered necessary for the subsequent stages of the study, since it is believed that the model's challenge in achieving better performance, among other factors, is primarily attributed to the relatively small dataset, which only consists of 410 samples. Therefore, the dataset’s size can either increase the likelihood of detecting patterns that do not genuinely exist or increase the possibility of overfitting the model, which is visible in the three out of four experiments. In more detail, the model failed to improve and to surpass the 0.60 in the F1 score in the first three experiments. The only experiment that managed to learn some useful information from the dataset and to slightly get generalized is the fourth, which consists of 100 epochs and the big learning rate. As mentioned earlier, the choice of 100 epochs seems to be the preferred over the 50 epochs, since it gives the model the enough number of iterations to learn from the data, and the big learning rate corresponds to a faster learning, meaning bigger but less accurate steps during training. Thus, the model managed to show a small improvement, but its performance is not the desired one, and for that the implementation of data augmentation techniques is inevitable in order to make it more generalized and robust. Therefore, implementing experiments with a larger volume of data is considered of utmost importance. Hence, in the second approach that follows is showed how the proposed model responds to a larger dataset, within the same experimental configurations.

Second Approach

Similarly to the first approach, below are presented the four figures of the second approach, which depict the model's performance in the form of diagrams, including the F1 score (left diagrams) and the loss function (right diagrams), for both the training and validation phases of each of the four different experiment categories conducted. Subsequently, follows a brief analysis of these diagrams, from which various conclusions will be drawn regarding the model. These findings will guide subsequent actions to enhance the model in the future. Additionally, the F1 score for the testing set is presented for each one of the experiments.

Firstly, there are presented in the Figure 26 results of the experiment with a learning rate of 0.001 and 50 epochs are presented. This experiment had a testing **F1 score** equal to **0.7196611072868108.**



(a)

(b)

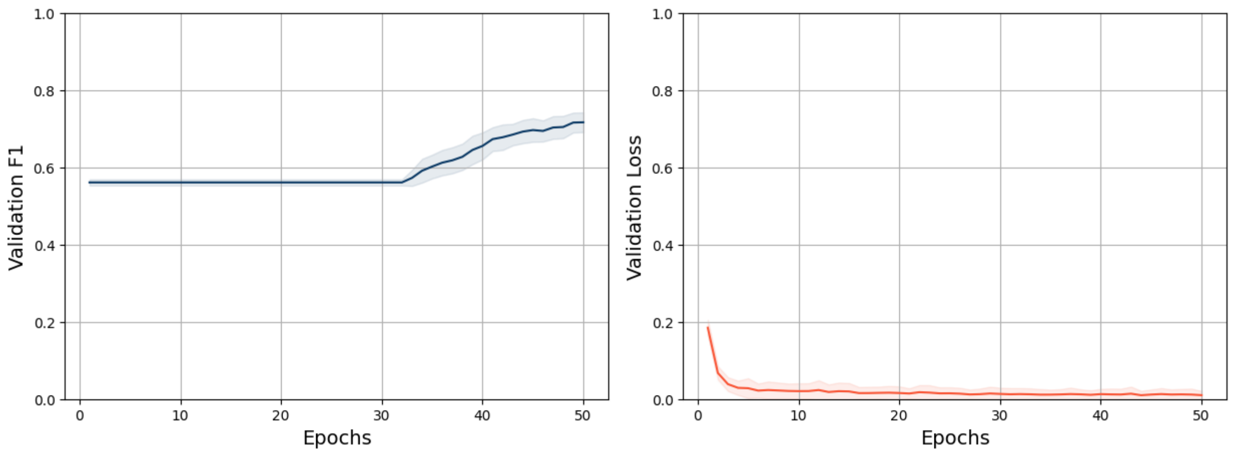
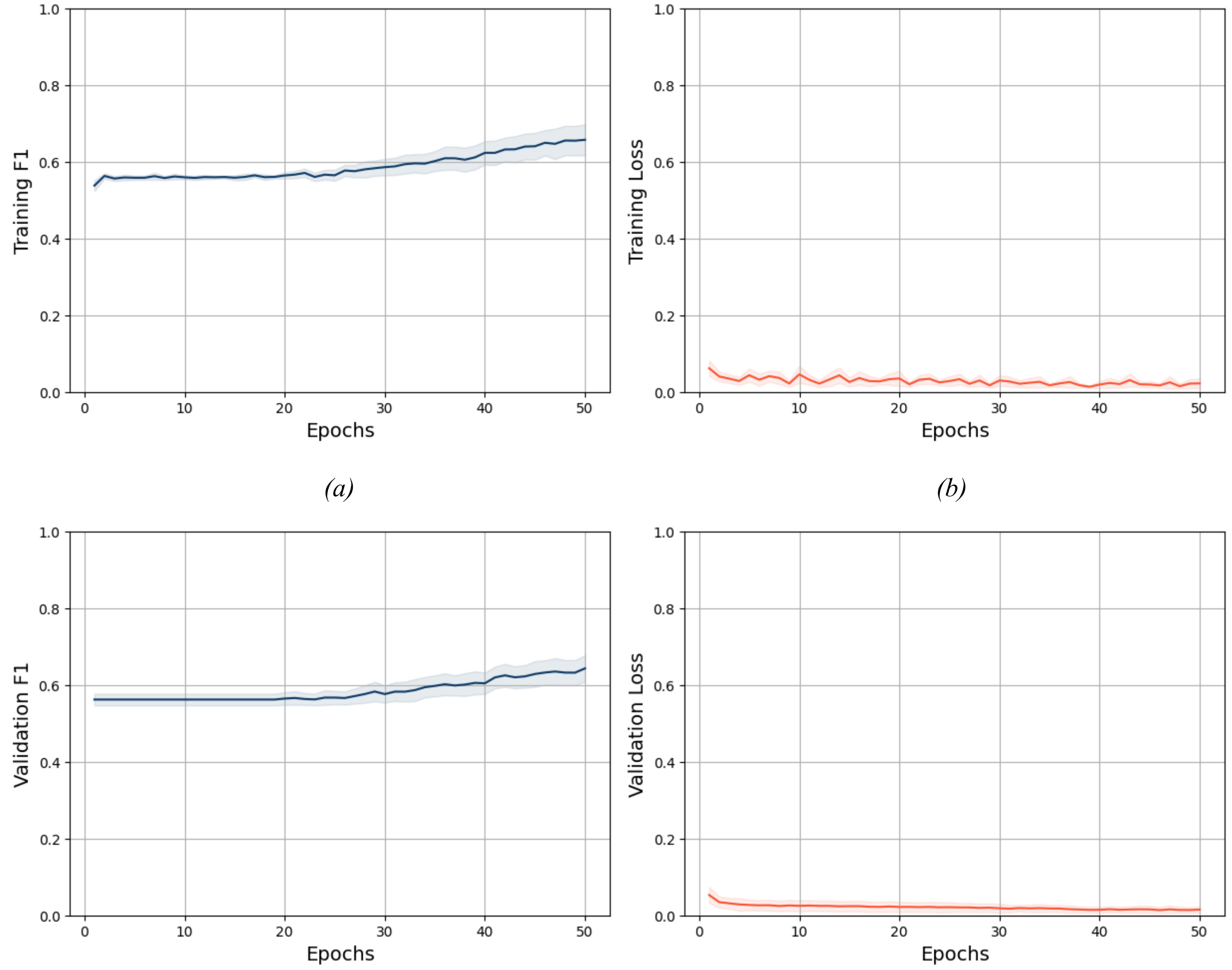


Figure 26. F1 score and Loss of the first experiment (learning rate = 0.001 and 50 epochs). (a) Shows the F1 score of the training set, (b) the Loss of the training set, (c) the F1 score of the validation set and (d) the loss of the validation set.

Through the observation of the Figure 26, various conclusions for the model can be drawn regarding the fifth experiment of the study. Specifically, following the same pattern as the Figure 25, it becomes evident by the (a) chart, that the model showed an improvement in the F1 score for the training set during the first two epochs. However, from the 3rd onwards until the 34th epoch, it did not show any substantial improvement, being consistently around 0.57. Although, from the 35th epoch until the last (50th), it showed a significant improvement, reaching up to 0.78. A similar pattern was also followed at the F1 score of the validation set, (c) chart, with the notable difference being that the learning process being completely zero in the initial 32 epochs, since the F1 score line was flat. After that point, the model started "learning", showing repeated improvements and F1 score line reached approximately 0.72 at the last epoch. It is worth noting that the variance in both charts is very small in the first 32 epochs and from that point onwards significantly big. When comparing the variances of those two charts, it is observed that (a) has a slightly greater variance.

Regarding charts (b) and (d), a notable resemblance between them is also visible. In both cases, they start from a value of approximately 0.2, experience a quick decline below 0.1 within the initial 3 epochs, and ultimately converge at a value of 0.05 by the 5th epoch. From that moment, they show a constant decrease until the end of the experiment, where finally the loss value is approximately 0.02 in both charts. The only difference between those two charts is the considerably smaller variance in the loss chart of the training set. Furthermore, chart (d) exhibits a smoother line, with way less abrupt fluctuations between consecutive epochs.

Subsequently, there are depicted in the Figure 27 the outcomes of the experiment conducted with a learning rate of 0.01 and 50 epochs. This experiment achieved a testing **F1 score of 0.6446228600210615.**



(a) *(b)*

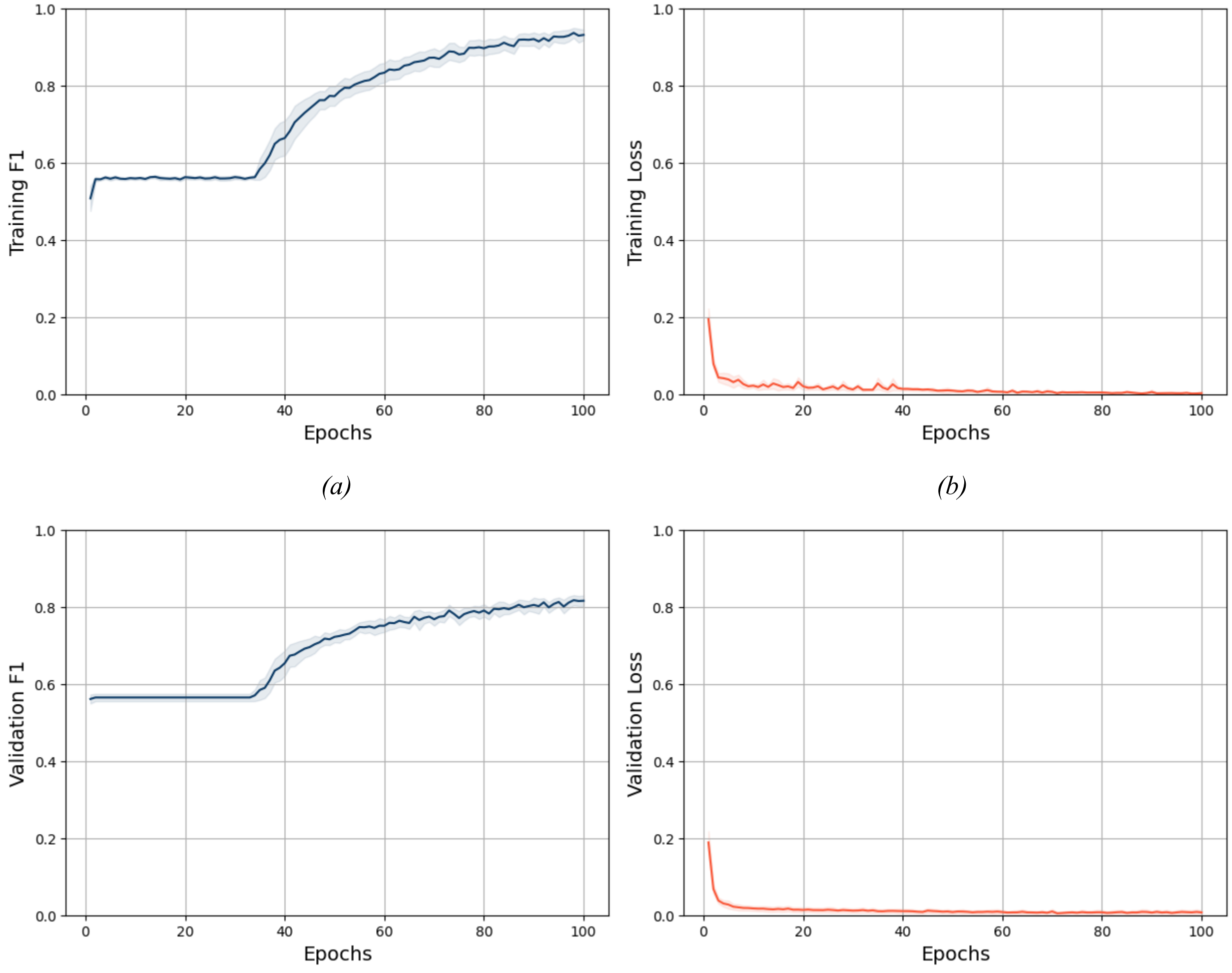
(c) *(d)*

Figure 27. F1 score and Loss of the second experiment (learning rate = 0.01 and 50 epochs). (a) Shows the F1 score of the training set, (b) the Loss of the training set, (c) the F1 score of the validation set and (d) the loss of the validation set.

Observing the results of the third experiment of the second approach of the research, various interesting conclusions concerning the model, become evident. More analytically, at the chart (a) of Figure 27 the model showed a quick increase at the F1 score of the training set during the first two epochs, as it initiated at approximately 0.54 and reached up to 0.57. From that point onwards until the middle of the experiment, it showed almost zero improvement in the learning process, but from the 25th epoch until the end it demonstrated some improvement, with the F1 score line ending in a value of approximately 0.65. Regarding the F1 score of the validation set, as indicated in chart (c), it remained consistently flat across the first 20 epochs and from that point onwards it managed to show a slight improvement. More specifically, it initiated from the area of 0.57, remained still until the 20th epoch and reached approximately 0.65 at the 50th epoch. Concerning the variation of those two charts, it is easily visible that in the first 20 epochs the chart (c) has the highest, given the fact that in the (a) chart the variation is very low, but from the 20th epoch until the last both charts have way higher variation comparing to the first part, with the chart's (a) to be the highest.

Regarding charts (b) and (d), a notable resemblance between them is also discernible. In both cases, they start from a value of approximately 0.07 and experience a small decline within the initial 3 epochs. From that moment onwards continue a constant decline and until the end of the experiment, where finally the loss value is approximately 0.03 and 0.02, for the chart (b) and (d), respectively. Moreover, the variance of the two charts is relatively equal. Thus, the only difference between them is the significantly smoother line exhibited in chart (d), which lacks abrupt fluctuations between consecutive epochs compared to to the line of chart (c).

Following this there are presented in the Figure 28 findings from the experiment involving a learning rate of 0.001 and 100 epochs. In this experiment, the achieved testing **F1 score** was **0.8142842385917902**.



(a) *(b)*

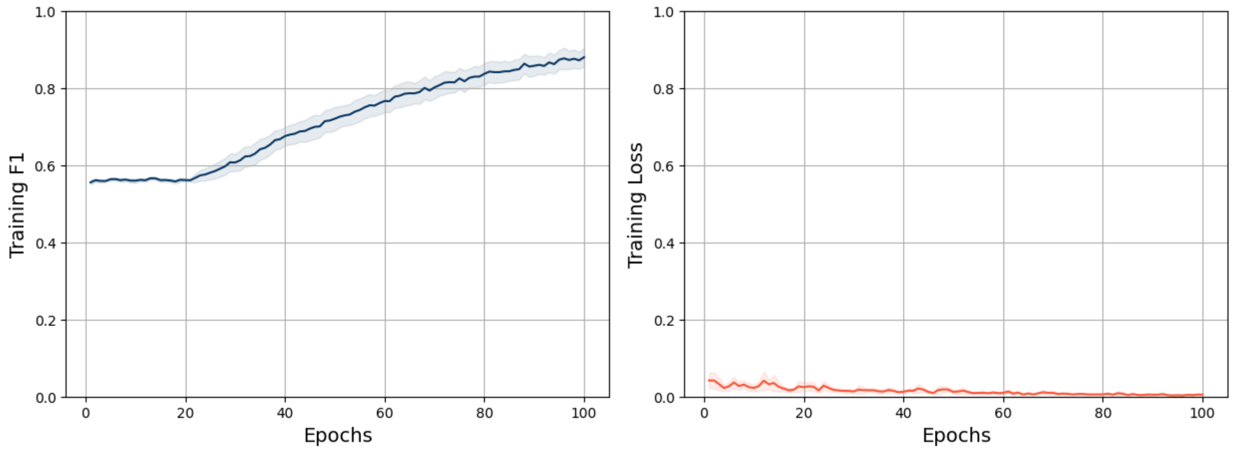
(c) *(d)*

Figure 28. F1 score and Loss of the third experiment (learning rate = 0.001 and 100 epochs). (a) Shows the F1 score of the training set, (b) the Loss of the training set, (c) the F1 score of the validation set and (d) the loss of the validation set.

Through the observation of the figure above, it is easily noticeable that all four charts follow a similar pattern as the ones in Figure 26 with some differences being visible as well. More analytically, for the chart (a), the line of F1 score starts from the area of 0.50 and showed a slight increase during the first two epochs, reaching up to reached 0.56. From that point onwards, until the 34th epoch, it did not demonstrate any further learning and it stayed relatively constant throughout the remainder of the experiment. Although, from the 35th epoch until the last (100th), it showed a huge improvement, reaching up to 0.91. A similar pattern was also followed at the F1 score of the validation set, (c) chart, with the difference being that for the initial 34 epochs the learning process was completely zero, since the F1 score line was completely flat. From that point, the model started "learning", showing repeated improvements and the F1 score line reached approximately 0.82 at the last epoch. Concerning the variance of the F1 score, it is observed that both charts have very small in the first 34 epochs and from that point onwards significantly bigger. When comparing the variances between the two charts, it is observed that (a) has slightly greater.

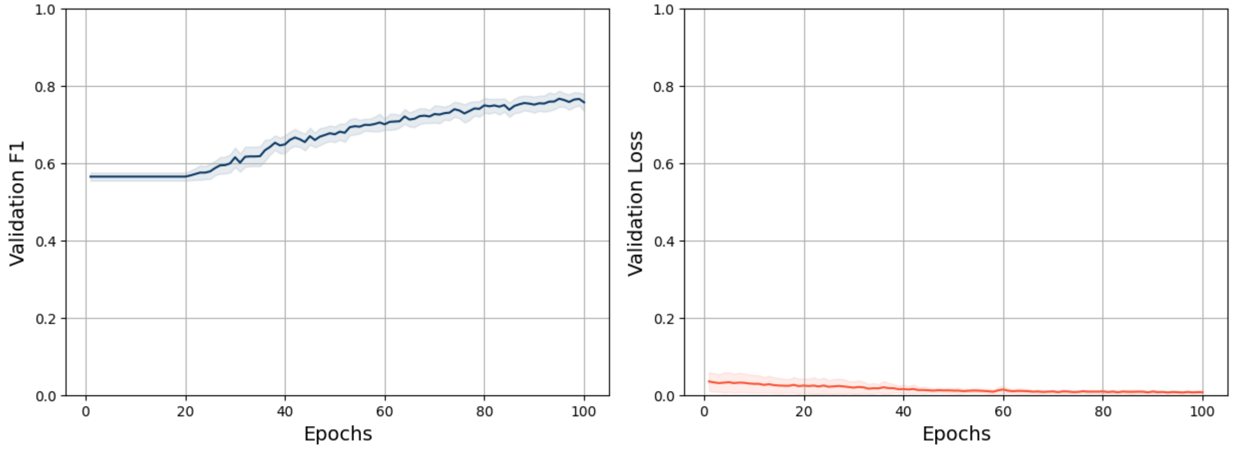
Regarding charts (b) and (d), a notable resemblance between them is also visible. In both cases, they start from a value of approximately 0.2, experience a quick decline below 0.1 within the initial 3 epochs, and ultimately converge at a value of 0.05 by the 5th epoch. From that moment, they show a constant decrease until the end of the experiment, where finally the loss value is approximately 0.01 in both charts. The only difference between those two charts is the slightly smaller variance in the loss chart of the validation set. Furthermore, chart (d) exhibits a smoother line, with less abrupt fluctuations between consecutive epochs compared to the line of chart (a).

Lastly, there are presented in the Figure 28 the results of the experiment that was implemented with a learning rate of 0.01 and 100 epochs, which achieved a testing **F1 score** of **0.7725044441719849**.



(a)

(b)



(c) *(d)*

Figure 29. F1 score and Loss of the fourth experiment (learning rate = 0.01 and 100 epochs). (a) Shows the F1 score of the training set, (b) the Loss of the training set, (c) the F1 score of the validation set and (d) the loss of the validation set.

Considering the figure above, it is easily noticed the high similarity with the Figure 28 charts (a) and (c) start from the area of 0.56 and until the 21st and 20th epoch respectively they failed to exhibit any discernible progress. In more detail, the F1 score at the training set, chart (a), did not show any substantial improvement, being consistent around the area of 0.57, with some small fluctuations, while in the F1 score of the validation set, chart (c), the learning process is completely zero in the initial 20 epochs. From that moment on, the model showed repeated improvements and the F1 score line reached approximately 0.87 for the chart (a) and 0.76 for the chart (c) at the end of the experiment (100 epochs). Concerning the variance of the two lines, it is worth noting that the variance in both charts is very big in the first 30 epochs approximately, and from that point onwards significantly small. When comparing the variances of the two charts, it is observed that chart (a) has slightly greater.

Regarding charts (b) and (d), a notable resemblance between them is also discernible. Both charts initiate at a value of approximately 0.04 and they end after 100 epochs at a value around 0.01. Some of the few differences between the two charts include the slightly smaller variance in the loss chart of the training set in comparison to the one shown in the validation set. Furthermore, chart (d) exhibits a very smooth line, with way less and smaller abrupt fluctuations, while chart (b) has sharp fluctuations between consecutive epochs, especially at the first 40 epochs of the experiment.

Second Approach Results – Analysis

The conclusions drawn from the second approach results, which are based on the analysis of the diagrams above, can be particularly beneficial for the future steps of the study. Similarly to the first analysis, for the evaluation of the suitability of the model's learning rate, the four experiments have to be compared between each other in a way that have the rest conditions and configurations same, such as the epochs and the dataset. As the dataset remains consistent across all first approach experiments, comparisons are made between two pairs: the first-second experiment and the third-fourth experiment, both having the same number of epochs. However, opposite to the first approach, where the learning rates did not have consistent results, it is observed that the small learning rate (lr = 0.001), in both pairs manage to achieve a better performance than the large learning rate (lr = 0.01). Therefore, the model is not sensitive to the number of epochs, like it was in the first approach. Specifically, in the first and third experiments, the ones with lr = 0.001, then model managed to achieve a higher F1 score than the other two, when comparing them in the aforementioned pairs. Concerning the loss values, it is observed that in the first pair, the two experiments have the same loss value in the training set, while for the validation set the lowest value is achieved in the second experiment, viz. when the model was trained with the big learning rate. On the contrary, in the second pair, the two experiments have the same loss value for both training and validation set. Hence, it is proved that between these two learning rates the smaller is more suitable for the model, when the volume of the dataset is big enough to train effectively and generalize the model. However, it is noted that similarly to the first approach the large learning rate was significantly faster compared to the small one during the model's training.

Regarding the most suitable number of epochs, it is easily visible that the two 100-epoch experiments achieve a better overall performance, with both the highest F1 score and smaller loss value, than the two 50-epoch experiments. To be more specific, the F1 score of the testing set in the first and second experiments is around 0.72 and 0.64, respectively, while on the third and fourth is approximately 0.81 and 0.77, respectively, which is a significant difference in performance between the 50-epoch experiments and the 100-epoch experiments. That happens because the model keeps "learning" after the 50th epoch, under these configurations and the augmented dataset, which results in a better segmentation and smaller error. Similarly to the first approach, it is noted that as expected the training of the 100-epoch experiments is slower than the 50-epoch ones. It would be interesting to check the model's behavior in an experiment of more than 100 epochs, in order to see at what point it would reach its plateau in terms of learning, meaning the point that the it cannot improve with additional epochs.

In general, the most successful experiment of the four proved to be the one with 100 epochs and the small learning rate (lr = 0.001), based on both the F1 score and loss value of both training and validation set, as well as the F1 score of the testing set. That is logical, because as previously explained, the model continued improving past the 50-epoch mark, making the 100-epoch experiments more accurate and at the same time the small learning rate is the most suitable of the two. Opposite to that, the least successful experiment was the second one, which includes 50 epochs of training and the large learning rate (lr = 0.01), given the fact that it had the lowest F1 scores in all three sets (training, validation and testing) and the highest error value in the training and validation sets (it was slightly higher than the first's experiment).

Lastly, it is mentioned that the results of the second approach are significantly better than the first's, which means that the data augmentation techniques that were selected are very successful. That is translated into a better generalization of the model, which is gained through the bigger volume of the training set. It would be also interesting to check the model's performance under additional data augmentations, in order to get a more complete understanding on its behavior, its strengths and weaknesses.

Conclusions

In this work, is proposed a convolution neural network for breast tumor segmentation in mammogram images. The proposed framework utilizes the segmentation of the U-net model, while the INbreast dataset, an open dataset that consists of 410 digital mammograms and their corresponding masks, is selected for the training and evaluation of its performance. The metrics used for the assessment of the study's performance are the F1 score and the loss value.

On the contrary of the most studies in the field, which use very complex architectures, this research emphasizes in the implementation of various data preprocessing and augmentation techniques, in order to maximize the performance of the model with the use of a small dataset. Due to the fact that medical datasets are rare to find or have a relatively small size in the real world, this approach could be helpful to the community. The idea of the study relies on the creation of an efficient model for cancer detection and segmentation in mammograms. Its importance can be described in the dataset's nature used for the training of the network. More specifically, in this work are presented two approaches, that differ in the dataset in use. The first entails the utilization of the INbreast dataset as is, while the second involves the augmentation of it, in order to check the performance of the model in a larger dataset. Thus, the second approach uses a dataset six times the size of the initial INbreast dataset. The augmentation techniques selected for the above are the histogram equalization, the gamma correction, and the 180-degree rotation.

In each approach, four experiments were conducted. Two of them are with a small learning rate (lr = 0.001); one with 50 epochs and one with 100 epochs and two with a big learning rate (lr = 0.001); one with 50 epochs and one with 100 epochs. All the experiments of the first approach proved to achieve insufficient results, with the only exception being the fourth experiment, meaning the one with the big learning rate and the 100 epochs. To be more specific, the rest of the experiments did not manage to "learn" from the data since their F1 line in the diagrams of the training and validation sets remained flat as well as did not surpass the F1 score of 0.60 in the testing set. Thus, came the idea of the the second approach, given the fact that the model got overfitted and the only solution to fix this common issue of the small datasets is to feed them with a bigger dataset. Indeed, the second approach provided significantly better results, since all its experiments showed a successful learning. The F1 score of all its four experiments in the testing set was around 0.72, 0.64, 0.81, 0.77, which is a way better performance than the first approach. The best experiment of all proved to be the third of the second approach, meaning the 100-epoch experiment with the small learning rate (lr = 0.001), which achieved an F1 score of 0.81 and a loss value of 0.01.

The conclusion of the above is that the proposed network is able to achieve a decent segmentation result, with the utilization of a small dataset (INbreast), by using data augmentation techniques. In more detail, this study proved that the model needs a dataset with a bigger volume to achieve a good performance and that the augmentation techniques selected were successful. Based on the best experiment of all, it would be safe to assume that the learning rate of 0.001 and the 100 epochs ensure the best results in a large dataset. Given that the second approach was significantly more successful, it is more accurate for the conclusions to be primarily based on it. Thus, similarly to the aforementioned assumption, the correct number of training epochs is 100 and the more efficient learning rate of the two seems to be the smaller one. However, it is highlighted that while on the former parameter the model has consistent results on the two approaches, which shows that indeed the larger number of epochs gives to the model the ability to learn better from the data (or even learn at all), while this consistency is not seen on the latter parameter. The two learning rates do not have consistent results in the two approaches, since in the first approach the best of the four experiments is with the big learning rate, while in the second is with the small. Hence, a more accurate conclusion might be that the correct learning rate depends on the dataset in use but based on the conducted experiments in a sufficiently large enough dataset, the small learning rate is more preferable. Although, it is noted that as expected the small learning rate is way more time consuming during the training compared to the big one. The learning rate is an undeniably vital factor for every ANN, so an interesting idea would be a learning rate that changes during the training, specifically starting as large and decreasing over time, in order to be faster than the small learning rate but theoretically equally accurate. This innovative concept, which involves a variable learning rate, could be investigated in a future work. Moreover, future research could continue to explore other combinations of data augmentations techniques, in order to get a more complete understanding of the model’s behavior, its strengths and weaknesses.

References

[1] Sung, Hyuna, et al. "Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries." CA: a cancer journal for clinicians 71.3 (2021): 209-249.

[2] Cancer Council Australia and N.S.W. Cancer Council, issuing body. Understanding breast cancer : a guide for people with cancer, their families and friends Sydney, NSW: Cancer Council, 2022.

[3] Ruddon, Raymond W. Cancer biology. Oxford University Press, 2007.

[4] Chen, Yuehui, Yan Wang, and Bo Yang. " Evolving hierarchical RBF neural networks for breast cancer detection." International Conference on Neural Information Processing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[5] Ferlay, Jacques, et al. " Cancer statistics for the year 2020: An overview." International journal of cancer 149.4 (2021): 778-789.

[6] Peruchet-Noray, Laia, et al. "Body Shape Phenotypes and Breast Cancer Risk: A Mendelian Randomization Analysis." Cancers 15.4 (2023): 1296.

[7] Zhang, Ben, et al. "Height and breast cancer risk: evidence from prospective studies and Mendelian randomization." Journal of the National Cancer Institute 107.11 (2015): djv219.

[8] Aggarwal, Ravi, et al. "Diagnostic accuracy of deep learning in medical imaging: a systematic review and meta-analysis." NPJ digital medicine 4.1 (2021): 65.

[9] Nassif, Ali Bou, et al. "Breast cancer detection using artificial intelligence techniques: A systematic literature review." Artificial Intelligence in Medicine 127 (2022): 102276.

[10] Yao, Hongdou, et al. "Parallel structure deep neural network using CNN and RNN with an attention mechanism for breast cancer histology image classification." Cancers 11.12 (2019): 1901.

[11] Ha, Richard, et al. "Convolutional neural network using a breast MRI tumor dataset can predict oncotype Dx recurrence score." Journal of Magnetic Resonance Imaging 49.2 (2019): 518-524.

[12] Huang, Mei-Ling, and Ting-Yu Lin. "Dataset of breast mammography images with masses." Data in brief 31 (2020): 105928.

[13] Yassin, Nisreen IR, et al. "Machine learning techniques for breast cancer computer aided diagnosis using different image modalities: A systematic review." Computer methods and programs in biomedicine 156 (2018): 25-45.

[14] Bhatt, Chandradeep, et al. "The state of the art of deep learning models in medical science and their challenges." Multimedia Systems 27.4 (2021): 599-613.

[15] Fujioka, Tomoyuki, et al. "The utility of deep learning in breast ultrasonic imaging: a review." Diagnostics 10.12 (2020): 1055.

[16] Jain, Ravi, and Ajith Abraham. "A comparative study of fuzzy classifiers on breast cancer data." Artificial Neural Nets Problem Solving Methods: 7th International Work-Conference on Artificial and Natural Neural Networks, IWANN2003 Maó, Menorca, Spain, June 3–6, 2003 Proceedings, Part II 7. Springer Berlin Heidelberg, 2003.

[17] Saadatmand, Sepideh, et al. "Influence of tumour stage at breast cancer detection on survival in modern times: population based study in 173 797 patients." Bmj 351 (2015).

[18] Kaplan, Henry G., et al. "Effect of treatment and mammography detection on breast cancer survival over time: 1990-2007." Cancer 121.15 (2015): 2553-2561.

[19] Moreira, Inês C., et al. "Inbreast: toward a full-field digital mammographic database." Academic radiology 19.2 (2012): 236-248.

[20] Sardanelli, Francesco, et al. "Mammography: an update of the EUSOBI recommendations on information for women." Insights into imaging 8 (2017): 11-18.

[21] Sardanelli, Francesco, et al. "Position paper on screening for breast cancer by the European Society of Breast Imaging (EUSOBI) and 30 national breast radiology bodies from Austria, Belgium, Bosnia and Herzegovina, Bulgaria, Croatia, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Israel, Lithuania, Moldova, The Netherlands, Norway, Poland, Portugal, Romania, Serbia, Slovakia, Spain, Sweden, Switzerland and Turkey." European radiology 27 (2017): 2737-2743.

[22] Prummel, Maegan V., et al. "Digital compared with screen-film mammography: measures of diagnostic accuracy among women screened in the Ontario breast screening program." Radiology 278.2 (2016): 365-373.

[23] Suryanarayanan, Sankararaman, Andrew Karellas, and Srinivasan Vedantham. "Physical characteristics of a full-field digital mammography system." Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 533.3 (2004): 560-570.

[24] Törnberg, Sven, et al. "A pooled analysis of interval cancer rates in six European countries." European journal of cancer prevention 19.2 (2010): 87-93.

[25] Carbonaro, Luca A., et al. "Interval breast cancers: absolute and proportional incidence and blinded review in a community mammographic screening program." European Journal of Radiology 83.2 (2014): e84-e91.

[26] Lauby-Secretan, Beatrice, et al. "Breast-cancer screening—viewpoint of the IARC Working Group." New England journal of medicine 372.24 (2015): 2353-2358.

[27] Sampat, Mehul P., et al. "A model-based framework for the detection of spiculated masses on mammography a." *Medical physics* 35.5 (2008): 2110-2123.

[28] Bird, Richard E., Terry W. Wallace, and Bonnie C. Yankaskas. "Analysis of cancers missed at screening mammography." *Radiology* 184.3 (1992): 613-617.

[29] Kerlikowske, Karla, et al. "Performance of screening mammography among women with and without a first-degree relative with breast cancer." *Annals of internal medicine* 133.11 (2000): 855-863.

[30] Zheng, Bin, et al. "Mammography with computer-aided detection: reproducibility assessment—initial experience." Radiology 228.1 (2003): 58-62.

[31] Castellino, Ronald A. "Computer aided detection (CAD): an overview." Cancer Imaging 5.1 (2005): 17.

[32] Hamet, Pavel, and Johanne Tremblay. "Artificial intelligence in medicine." Metabolism 69 (2017): S36-S40.

[33] Rajaraman, Vaidyeswaran. "JohnMcCarthy—Father of artificial intelligence." Resonance 19 (2014): 198-207.

[34] Bowen, Jonathan P. "Alan Turing: founder of computer science." School on Engineering Trustworthy Software Systems. Cham: Springer International Publishing, 2016. 1-15.

[35] Wang, Fei, and Anita Preininger. "AI in health: state of the art, challenges, and future directions." Yearbook of medical informatics 28.01 (2019): 016-026.

[36] Ramesh, A. N., et al. "Artificial intelligence in medicine." Annals of the Royal College of Surgeons of England 86.5 (2004): 334.

[37] Bi, Qifang, et al. "What is machine learning? A primer for the epidemiologist." American journal of epidemiology 188.12 (2019): 2222-2239.

[38] Mahesh, Batta. "Machine learning algorithms-a review." International Journal of Science and Research (IJSR).[Internet] 9.1 (2020): 381-386.

[39] Abadi, Martín, et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." arXiv preprint arXiv:1603.04467 (2016).

[40] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015): 436-444.

[41] Awad, Mariette, and Rahul Khanna. Efficient learning machines: theories, concepts, and applications for engineers and system designers. Springer nature, 2015.

[42] Cunningham, Pádraig, Matthieu Cord, and Sarah Jane Delany. "Supervised learning." Machine learning techniques for multimedia: case studies on organization and retrieval. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. 21-49.

[43] Ragab, Dina A., et al. "Breast cancer detection using deep convolutional neural networks and support vector machines." *PeerJ* 7 (2019): e6201.

[44] Salama, Wessam M., and Moustafa H. Aly. "Deep learning in mammography images segmentation and classification: Automated CNN approach." *Alexandria Engineering Journal* 60.5 (2021): 4701-4709.

[45] Baccouche, Asma, et al. "Early detection and classification of abnormality in prior mammograms using image-to-image translation and YOLO techniques." *Computer Methods and Programs in Biomedicine* 221 (2022): 106884.

[46] Tzortzis, Ioannis N., et al. "Tensor-Based Learning for Detecting Abnormalities on Digital Mammograms." *Diagnostics* 12.10 (2022): 2389.

[47] Vijayakumar, K., Vinod J. Kadam, and Sudhir Kumar Sharma. "Breast cancer diagnosis using multiple activation deep neural network." *Concurrent Engineering* 29.3 (2021): 275-284.

[48] Desai, Meha, and Manan Shah. "An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN)." Clinical eHealth 4 (2021): 1-11.

[49] Antonelli, Laura, Valentina De Simone, and Daniela di Serafino. "A view of computational models for image segmentation." ANN Hongdou DELL'UNIVERSITA'DI FERRARA 68.2 (2022): 277-294.

[50] Kaur, Dilpreet, and Yadwinder Kaur. "Various image segmentation techniques: a review." International Journal of Computer Science and Mobile Computing 3.5 (2014): 809-814.

[51] Sultana, Farhana, Abu Sufian, and Paramartha Dutta. "Evolution of image segmentation using deep convolutional neural network: A survey." Knowledge-Based Systems 201 (2020): 106062.

[52] American College of Radiology. "Breast imaging reporting and data system." BI-RADS (2003).

[53] Malmartel, Alexandre, Arthur Tron, and Ségolène Caulliez. "Accuracy of clinical breast examination's abnormalities for breast cancer screening: cross-sectional study." European Journal of Obstetrics & Gynecology and Reproductive Biology 237 (2019): 1-6.

[54] Abbasi, Fatima, et al. "Frequency of Clinically Palpable Breast Lumps in an Urban Medical Center Importance of a Surgeon Run Breast Clinic." Pakistan Journal of Medical & Health Sciences 17.02 (2023): 414-414.

[55] Huang, Jianglin, Yan-Fu Li, and Min Xie. "An empirical analysis of data preprocessing for machine learning-based software cost estimation." Information and software Technology 67 (2015): 108-127.

[56] Shrivastava, Himanshu, and Srivatsan Sridharan. "Conception of data preprocessing and partitioning procedure for machine learning algorithm." International Journal of Recent Advances in Engineering & Technology (IJRAET) 1.3 (2013): 2347-2812.

[57] Ahmad, Tohari, and Mohammad Nasrul Aziz. "Data preprocessing and feature selection for machine learning intrusion detection systems." ICIC Express Lett 13.2 (2019): 93-101.

[58] Yan, Jianzhou, et al. "Learning the change for automatic image cropping." Proceedings of the IEEE conference on computer vision and pattern recognition. 2013.

[59] Quiring, Erwin, et al. "Adversarial preprocessing: Understanding and preventing {Image-Scaling} attacks in machine learning." 29th USENIX Security Symposium (USENIX Security 20). 2020.

[60] Talebi, Hossein, and Peyman Milanfar. "Learning to resize images for computer vision tasks." Proceedings of the IEEE/CVF international conference on computer vision. 2021.

[61] Liu, Zhenyu. "A method of SVM with normalization in intrusion detection." Procedia Environmental Sciences 11 (2011): 256-262.

[62] Patro, S. G. O. P. A. L., and Kishore Kumar Sahu. "Normalization: A preprocessing stage." arXiv preprint arXiv:1503.06462 (2015).

[63] Betz, Volker. "Random permutations." (2019).

[64] Zheng, Fei, et al. "Towards secure and practical machine learning via secret sharing and random permutation." Knowledge-Based Systems 245 (2022): 108609.

[65] Bacher, Axel, et al. "Mergeshuffle: A very fast, parallel random permutation algorithm." arXiv preprint arXiv:1508.03167 (2015).

[66] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer International Publishing, 2015.

[67] Chen, Pengguang, et al. "Gridmask data augmentation." arXiv preprint arXiv:2001.04086 (2020).

[68] Abeysinghe, Asith, et al. "Data augmentation on convolutional neural networks to classify mechanical noise." Applied Acoustics 203 (2023): 109209.

[69] Li, Bohan, Yutai Hou, and Wanxiang Che. "Data augmentation approaches in natural language processing: A survey." Ai Open 3 (2022): 71-90.

[70] Dosovitskiy, Alexey, et al. "Discriminative unsupervised feature learning with convolutional neural networks." Advances in neural information processing systems 27 (2014).

[71] Shorten, Connor, and Taghi M. Khoshgoftaar. "A survey on image data augmentation for deep learning." Journal of big data 6.1 (2019): 1-48.

[72] DeVries, Terrance, and Graham W. Taylor. "Improved regularization of convolutional neural networks with cutout." arXiv preprint arXiv:1708.04552 (2017).

[73] Gupta, P., et al. "Histogram based image enhancement techniques: a survey." *Int J Comput Sci Eng* 5.6 (2017): 475-484.

[74] Janani, P., J. Premaladha, and K. S. Ravichandran. "Image enhancement techniques: A study." *Indian Journal of Science and Technology* 8.22 (2015): 1-12.

[75] Wadi, Salim M., and Nasharuddin Zainal. "Contrast enhancement methods based on histogram equalization technique: Survey." *International Conference on Engineering and Built Environment (ICEBE)*. 2012.

[76] Kaur, Manpreet, Jasdeep Kaur, and Jappreet Kaur. "Survey of contrast enhancement techniques based on histogram equalization." *International Journal of Advanced Computer Science and Applications* 2.7 (2011).

[77] Teh, V., Kok Swee Sim, and Eng Kiong Wong. "Brain early infarct detection using gamma correction extreme-level eliminating with weighting distribution." *Scanning* 38.6 (2016): 842-856.

[78] Mustafa, Wan Azani, and Mohamed Mydin M. Abdul Kader. "A review of histogram equalization techniques in image enhancement application." Journal of Physics: Conference Series. Vol. 1019. IOP Publishing, 2018.

[79] Dorothy, R., et al. "Image enhancement by histogram equalization." International Journal of Nano Corrosion Science and Engineering 2.4 (2015): 21-30.

[80] Longkumer, Nungsanginla, et al. "Contrast Enhancement Using Various Statistical Operations and Neighborhood Processing." Signal & Image Processing 5.2 (2014): 51.

[81] Zhu, Youlian, and Cheng Huang. "An adaptive histogram equalization algorithm on the image gray level mapping." Physics Procedia 25 (2012): 601-608.

[82] Jaiswal, Rahul, A. G. Rao, and H. P. Shukla. "Image enhancement techniques based on histogram equalization." *International Journal of Electrical and Electronics Engineering* 1 (2010): 69-78.

[83] Cao, Gang, et al. "Contrast enhancement of brightness-distorted images by improved adaptive gamma correction." *Computers & Electrical Engineering* 66 (2018): 569-582.

[84] Gonzalez, Rafael C. *Digital image processing*. Pearson education india, 2009.

[85] Huang, Shih-Chia, Fan-Chieh Cheng, and Yi-Sheng Chiu. "Efficient contrast enhancement using adaptive gamma correction with weighting distribution." *IEEE transactions on image processing* 22.3 (2012): 1032-1041.

[86] Somasundaram, Karuppanagounder, and Palanisamy Kalavathi. "Medical image contrast enhancement based on gamma correction." *Int J Knowl Manag e-learning* 3.1 (2011): 15-18.

[87] MAIYANTI, Sri INDRA, et al. "ROTATION-GAMMA CORRECTION AUGMENTATION ON CNN-DENSE BLOCK FOR SOIL IMAGE CLASSIFICATION." Applied Computer Science 19.3 (2023): 96-115.

[88] Aleem, Sidra, et al. "Random data augmentation based enhancement: a generalized enhancement approach for medical datasets." *arXiv preprint arXiv:2210.00824* (2022).

[89] Khan, Asif, Hyunho Hwang, and Heung Soo Kim. "Synthetic data augmentation and deep learning for the fault diagnosis of rotating machines." Mathematics 9.18 (2021): 2336.

[90] Ghali, Sherif. "Affine Transformations." Introduction to Geometric Computing (2008): 35-50.

[91] Alomar, Khaled, Halil Ibrahim Aysel, and Xiaohao Cai. "Data augmentation in classification and segmentation: A survey and new strategies." Journal of Imaging 9.2 (2023): 46.

[92] Picard, Richard R., and Kenneth N. Berk. "Data splitting." The American Statistician 44.2 (1990): 140-147.

[93] Wu, Wenyan, et al. "A method for comparing data splitting approaches for developing hydrological ANN models." (2012).

[94] Kilic, Arman. "Artificial intelligence and machine learning in cardiovascular health care." The Annals of thoracic surgery 109.5 (2020): 1323-1329.

[95] Muraina, Ismail. "Ideal dataset splitting ratios in machine learning algorithms: general concerns for data scientists and data analysts." 7th International Mardin Artuklu Scientific Research Conference. 2022.

[96] Foody, Giles M., et al. "Training set size requirements for the classification of a specific class." Remote Sensing of Environment 104.1 (2006): 1-14.

[97] Bai, Yu, et al. "How important is the train-validation split in meta-learning?." International Conference on Machine Learning. PMLR, 2021.

[98] Vabalas, Andrius, et al. "Machine learning algorithm validation with a limited sample size." PloS one 14.11 (2019): e0224365.

[99] Kanal, Laveen, and B. Chandrasekaran. "On dimensionality and sample size in statistical pattern classification." *Pattern recognition* 3.3 (1971): 225-234.

[100] Raudys, Sarunas J., and Anil K. Jain. "Small sample size effects in statistical pattern recognition: Recommendations for practitioners." IEEE Transactions on pattern analysis and machine intelligence 13.3 (1991): 252-264.

[101] Joseph, V. Roshan. "Optimal ratio for data splitting." Statistical Analysis and Data Mining: The ASA Data Science Journal 15.4 (2022): 531-538.

[102] Ma, Jun. "Segmentation loss odyssey." arXiv preprint arXiv:2005.13449 (2020).

[103] Jadon, Shruti. "A survey of loss functions for semantic segmentation." 2020 IEEE conference on computational intelligence in bioinformatics and computational biology (CIBCB). IEEE, 2020.

[104] Kervadec, Hoel, et al. "Boundary loss for highly unbalanced segmentation." International conference on medical imaging with deep learning. PMLR, 2019.

[105] Milletari, Fausto, Nassir Navab, and Seyed-Ahmad Ahmadi. "V-net: Fully convolutional neural networks for volumetric medical image segmentation." 2016 fourth international conference on 3D vision (3DV). Ieee, 2016.

[106] Clough, James R., et al. "A topological loss function for deep-learning based image segmentation using persistent homology." IEEE transactions on pattern analysis and machine intelligence 44.12 (2020): 8766-8778.

[107] Sudre, Carole H., et al. "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations." Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3. Springer International Publishing, 2017.

[108] Yi, Dokkyun, Jaehyun Ahn, and Sangmin Ji. "An effective optimization method for machine learning based on ADAM." Applied Sciences 10.3 (2020): 1073.

[109] Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks." 2013 IEEE international conference on acoustics, speech and signal processing. Ieee, 2013.

[110] Deng, Li, et al. "Recent advances in deep learning for speech research at Microsoft." 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013.

[111] Wichrowska, Olga, et al. "Learned optimizers that scale and generalize." International conference on machine learning. PMLR, 2017.

[112] Yang, Li, and Abdallah Shami. "On hyperparameter optimization of machine learning algorithms: Theory and practice." Neurocomputing 415 (2020): 295-316.

[113] Sangwan, Venu, et al. "Estimation of battery parameters of the equivalent circuit models using meta-heuristic techniques." 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES). IEEE, 2016.

[114] Battiti, Roberto. "First-and second-order methods for learning: between steepest descent and Newton's method." Neural computation 4.2 (1992): 141-166.

[115] Beheshti, Zahra, and Siti Mariyam Hj Shamsuddin. "A review of population-based meta-heuristic algorithms." Int. j. adv. soft comput. appl 5.1 (2013): 1-35.

[116] Bottou, Léon, Frank E. Curtis, and Jorge Nocedal. "Optimization methods for large-scale machine learning." SIAM review 60.2 (2018): 223-311.

[117] Anandkumar, Animashree, and Rong Ge. "Efficient approaches for escaping higher order saddle points in non-convex optimization." Conference on learning theory. PMLR, 2016.

[118] Bottou, Léon. "Large-scale machine learning with stochastic gradient descent." Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers. Physica-Verlag HD, 2010.

[119] Dereventsov, Anton, Clayton G. Webster, and Joseph Daws. "An adaptive stochastic gradient-free approach for high-dimensional blackbox optimization." Proceedings of International Conference on Computational Intelligence: ICCI 2020. Springer Singapore, 2022.

[120] Alexandrov, Natalia M., and Robert Michael Lewis. "An overview of first-order model management for engineering optimization." Optimization and Engineering 2 (2001): 413-430.

[121] Attouch, Hedy, et al. "First-order optimization algorithms via inertial systems with Hessian driven damping." Mathematical Programming (2022): 1-43.

[122] Elshamy, Reham, et al. "Improving the efficiency of RMSProp optimizer by utilizing Nestrove in deep learning." Scientific Reports 13.1 (2023): 8814.

[123] Lydia, Agnes, and Sagayaraj Francis. "Adagrad—an optimizer for stochastic gradient descent." Int. J. Inf. Comput. Sci 6.5 (2019): 566-568.

[124] Duchi, John, Elad Hazan, and Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization." Journal of machine learning research 12.7 (2011).

[125] Kandel, Ibrahem, Mauro Castelli, and Aleš Popovič. "Comparative study of first order optimizers for image classification using convolutional neural networks on histopathology images." Journal of imaging 6.9 (2020): 92.

[126] Hinton, Geoffrey, Nitish Srivastava, and Kevin Swersky. "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent." Cited on 14.8 (2012): 2.

[127] Chandra, Kartik, et al. "Gradient descent: The ultimate optimizer." Advances in Neural Information Processing Systems 35 (2022): 8214-8225.

[128] Bae, Kiwook, Heechang Ryu, and Hayong Shin. "Does Adam optimizer keep close to the optimal point?." arXiv preprint arXiv:1911.00289 (2019).

[129] Halgamuge, Malka N., Eshan Daminda, and Ampalavanapillai Nirmalathas. "Best optimizer selection for predicting bushfire occurrences using deep learning." Natural Hazards 103.1 (2020): 845-860.

[130] Zaheer, Raniah, and Humera Shaziya. "A study of the optimization algorithms in deep learning." 2019 third international conference on inventive systems and control (ICISC). IEEE, 2019.

[131] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

[132] Lipton, Zachary C., Charles Elkan, and Balakrishnan Naryanaswamy. "Optimal thresholding of classifiers to maximize F1 measure." Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II 14. Springer Berlin Heidelberg, 2014.

[133] Sokolova, Marina, and Guy Lapalme. "A systematic analysis of performance measures for classification tasks." Information processing & management 45.4 (2009): 427-437.

[134] Sokolova, Marina, Nathalie Japkowicz, and Stan Szpakowicz. "Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation." Australasian joint conference on artificial intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[135] Maen, Adam. "Lesion Segmentation in 3D FDG-PET/CT Scans Using Deep Learning." (2022).

[136] Gu, Qiong, Li Zhu, and Zhihua Cai. "Evaluation measures of the classification performance of imbalanced data sets." Computational Intelligence and Intelligent Systems: 4th International Symposium, ISICA 2009, Huangshi, China, October 23-25, 2009. Proceedings 4. Springer Berlin Heidelberg, 2009.

[137] Bekkar, Mohamed, Hassiba Kheliouane Djemaa, and Taklit Akrouf Alitouche. "Evaluation measures for models assessment over imbalanced data sets." J Inf Eng Appl 3.10 (2013).

[138] Guilford, Joy Paul. "Psychometric methods." (1954).

[139] Brown, J. B. "Classifiers and their metrics quantified." Molecular informatics 37.1-2 (2018): 1700127.

[140] Dice, Lee R. "Measures of the amount of ecologic association between species." Ecology 26.3 (1945): 297-302.

[141] Sørensen, Thorvald Julius. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. I kommission hos E. Munksgaard, 1948.

[142] Van Rijsbergen, Cornelis Joost. "Foundation of evaluation." Journal of documentation 30.4 (1974): 365-373.

[143] Van Rijsbergen, Cornelis Joost. "A theoretical basis for the use of co-occurrence data in information retrieval." Journal of documentation 33.2 (1977): 106-119.

[144] Chinchor, Nancy, and Beth M. Sundheim. "MUC-5 evaluation metrics." Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993. 1993.

[145] Dubey, Aman, and Sandhya Tarar. "Evaluation of approximate rank-order clustering using Matthews correlation coefficient." Int J Eng Adv Technol 8.2 (2018): 106-13.

[146] Hicks, Steven A., et al. "On evaluation metrics for medical applications of artificial intelligence." Scientific reports 12.1 (2022): 5979.

[147] Jiao, Yasen, and Pufeng Du. "Performance measures in evaluating machine learning based bioinformatics predictors for classifications." Quantitative Biology 4 (2016): 320-330.

The source code can be found in the following GitHub link:

https://github.com/IoannisPan11/Breast_Tumor_Segmentation

Key Parts of the software

Code for Data Pre-processing:

```
def crop(img):

    l = 0
    r = img.shape[1] - 1
    t = 0
    b = img.shape[0] - 1

    while np.sum(img[:, l]) == 0: l += 1
    while np.sum(img[:, r]) == 0: r -= 1
    while np.sum(img[t, :]) == 0: t += 1
    while np.sum(img[b, :]) == 0: b -= 1

    return img[t:b, l:r], l, r, t, b


def resize_img(img, mask, dim):

    resized_img = cv2.resize(img, dim, interpolation = cv2.INTER_AREA)
    resized_mask = cv2.resize(mask, dim, interpolation=cv2.INTER_AREA)

    return resized_img, resized_mask
```

Code for Data Augmentation:

```
def augment_img_1(img):

    output = equalize_hist(img)

    return output

```

```

def augment_img_2(img):

 output = adjust_gamma(img,gamma=0.5,gain=1)

 return output


def augment_img_3(img):

 output = rotate(img, angle=180, resize=True)

 return output
```

Code for Data Splitting:

```
training_set_size = int(0.7 * (len(p_dataset)))
validation_set_size = int(0.2 * (len(p_dataset)))
testing_set_size = int(0.1 * (len(p_dataset)))

train_start = 0
train_end = training_set_size
valid_start = train_end
valid_end = valid_start + validation_set_size
test_start = valid_end
test_end = test_start + testing_set_size

training_set = p_dataset[train_start:train_end, :, :, :]
validation_set = p_dataset[valid_start: valid_end, :, :, :]
testing_set = p_dataset[test_start: test_end, :, :, :]

print("Training set: ", training_set.shape)
print("Validation set: ", validation_set.shape)
print("Testing set: ", testing_set.shape)
```

Code for Loaders:

```
import torch

class Dataset(torch.utils.data.Dataset):
 'Characterizes a dataset for PyTorch'
 def __init__(self, dataset):
  self.dataset = dataset

 def __len__(self):
  return len(self.dataset)

 def __getitem__(self, index):
  x = self.dataset[index, :, :, 0]
  y = self.dataset[index, :, :, 1]

  x = torch.from_numpy(x)
  y = torch.from_numpy(y)

  return x, y


train = Dataset(training_set)
params = {'batch_size': 7, 'shuffle': True}
training_generator = torch.utils.data.DataLoader(train, **params)
print("Number of batches (training): ", len(training_generator))

valid = Dataset(validation_set)
params = {'batch_size': 7, 'shuffle': False}
validation_generator = torch.utils.data.DataLoader(valid, **params)
print("Number of batches (validation): ", len(validation_generator))

test = Dataset(testing_set)
params = {'batch_size': 7, 'shuffle': False}
testing_generator = torch.utils.data.DataLoader(test, **params)
```

Code for Training Loop:

```
import numpy as np

max_epochs = 50
in_channels = 1
out_channels = 2

unet = UNET(in_channels, out_channels)
loss_fn = torch.nn.CrossEntropyLoss()
opt = torch.optim.Adam(unet.parameters(), lr=0.001)

training_f1 = np.zeros((max_epochs, 1))
training_loss = np.zeros((max_epochs, 1))
validation_f1 = np.zeros((max_epochs, 1))
validation_loss = np.zeros((max_epochs, 1))

device = 'cuda' if torch.cuda.is_available() else 'cpu'
print(device)
unet = unet.to(device)

print("Number of batches (training): ", len(training_generator))
print("Number of batches (validation): ", len(validation_generator))

'''Training'''
for epoch in tqdm(range(max_epochs)):
 print()
 f1_sum = 0

 for x, y in training_generator:
 x = torch.unsqueeze(x, 1)
 x = x.to(torch.float32)
 y = y.to(torch.int64)

 x = x.to(device)
 y = y.to(device)

 opt.zero_grad()
```

```
predictions = unet(x)
loss = loss_fn(predictions, y)
loss.backward()
opt.step()

f1 = calculate_dice(predictions, y)
f1 = f1.item()
f1_sum += f1

batch_mean_f1 = f1_sum/len(training_generator)
training_f1[epoch] = batch_mean_f1
training_loss[epoch] = loss.item()
print("Batch mean of f1 (training):
{:.6f}".format(float(training_f1[epoch])))


'''Validation'''
f1_sum = 0
for x, y in validation_generator:
x = torch.unsqueeze(x, 1)
x = x.to(torch.float32)
y = y.to(torch.int64)

x = x.to(device)
y = y.to(device)

with torch.no_grad():
 outputs = unet(x)
 loss = loss_fn(outputs, y)

 f1 = calculate_dice(outputs, y)
 f1 = f1.item()
 f1_sum += f1

batch_mean_f1 = f1_sum / len(validation_generator)
validation_f1[epoch] = batch_mean_f1
validation_loss[epoch] = loss.item()
```

```
print()
print("Batch mean of f1 (validation): {:.6f}".format(float(validation_f1[epoch])))
```