جامعة حلوان
HELWAN UNIVERSITY

Faculty of Engineering

Faculty of Engineering Helwan University

Graduation Project 2020

BREAST CANCER DETECTION FROM MAMMOGRAPHY USING DEEP LEARNING

Supervised by

Dr\ Nancy Salem

Prepared by

Ahmed Essam Mohamed

Heba-Tullah Samy Abdullah

Mariam Ahmed Mohamed

Salma Mohsen Salem

Yasmin Mostafa Abdel Azim

Acknowledgement

Surely, there are no words to express our thanks to anyone who helped us with this project. We especially thank Dr. Nancy Salem, who supervised this idea and helped us to implement it by encouraging, guiding and helpful criticism us in this project. We would like to express our great appreciation to Eng. Aya Mohamed Shaaban the teaching assistant in the biomedical engineering at Helwan University for her efforts in helping us.

.

Dedication

We would like to thank our family and friends who supported and encouraged us with all they had so that we could take a step forward with their dearest confidence that we will achieve this.

Table of contents

List of Figures

Tables

CHAPTER1

INTRODUCTION

Breast cancer is the most widespread type of cancer in women worldwide. As the number of cases continue to significantly increase day by day, the dangers of such a disease made it necessary to introduce new methods that help in the early detection and diagnosis of suspicious lesions. For that purpose, Computer-Aided Detection and Diagnosis (CADe & CADx) systems are being developed [1]. The techniques used for developing these systems can be categorized into two types:

1) the first is composed of multiple steps such as pre-processing, segmentation, feature extraction, and classification steps, which entirely based on image processing and traditional machine learning techniques (manually).
2) In contrast, the second category does not employ any feature extraction techniques for detecting the region of interest, but instead, it exploits all information available in the mammogram using the Convolutional Neural Network (CNN) to learn the features.

Numerous CAD systems proposed for detecting and classifying masses in the digital mammograms.so, A deep learning algorithm is developed using “CNN” that can accurately detect breast cancer on screening mammograms using an “end-to-end” training on the CBIS-DDSM digitized film mammograms approach that efficiently leverages training datasets with either complete clinical annotation or only the cancer status (label) of the whole image[2].

1.1 Objective

Making a CAD system that detects and classifies breast lesions that appear in Mammography to be benign or malignant using deep convolutional neural networks (shown in figure 1.1), and limit the need for taking breast biopsies as shown in figure 1.2 that have negative impacts on many aspects such as unnecessary operations, fear, pain, and cost.

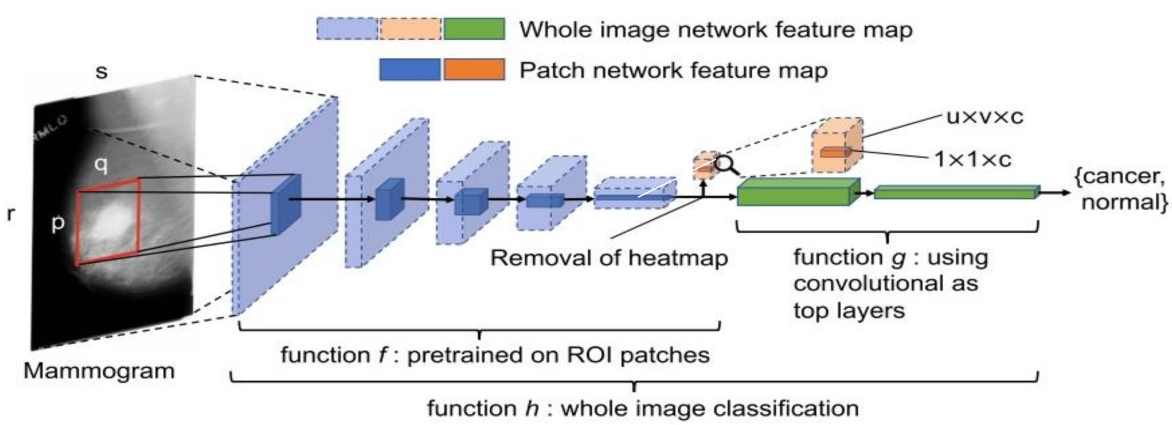

Figure 1.1 Converting a patch classifier to an end-to-end trainable whole image classifier using an all convolutional design[2].

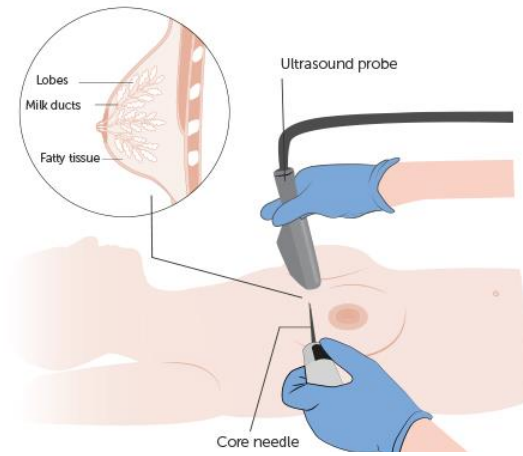

Figure 1.2 Core-needle biopsy [3].

1.2 Medical Background

Breast cancer is a group of diseases in which cells in breast tissue change and divide uncontrollably. These cells usually form a tumor that is visible on an x-ray or felt as a lump. The tumor is cancerous if the cells invade tissues or spread throughout the body. Although it can appear in men, breast cancer occurs almost entirely in women.

It usually begins in one of two places: the ducts that carry milk to the nipples (ductal cancer), or the milk glands (lobular cancer), which is the most abundant type of breast cancer (figure 1.3).

It is important to be aware of how your breasts look and feel normally. The earlier the cancer is detected, the better the chances of successful treatment.

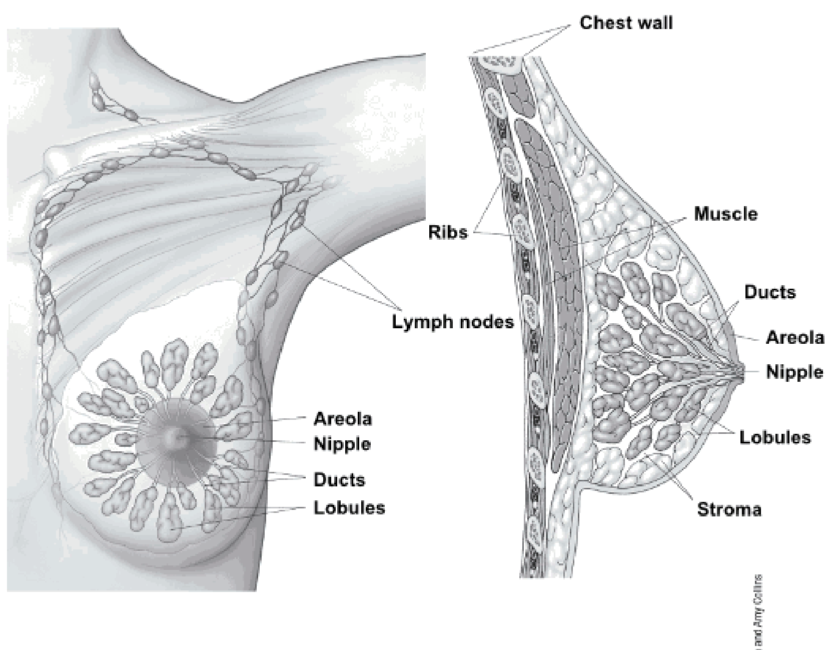

Figure 1.3 Normal Breast tissue[1].

1.2.1 Breast Cancer Symptoms

The best-known symptom is the lumps and nodes inside the breast or underarm area, but warning signs can vary between patients. Here are some other signs:

- Breast size and shape changes. That will commonly involve an increase in size or a change in shape of either one or both breasts.
- General pain in any part of the breast. Be wary of pain in a specific area that does not go away.
- Skin changes on the breast. Look out for skin changes in one or both breasts. These might include swelling, redness, darkening, dimpling or other visible differences. Take note of any prominent veins on the surface of the breast.
- Nipple discharge.
- The appearance of one or both nipples changes.
- Sores or rashes on the nipple, or if the nipple is beginning to turn inward.

An examination can help find breast cancer in its early stages before any symptoms begin to appear [4]. That's why screening mammography is important.

According to the World Health Organization (WHO), breast cancer is the most frequent type of cancer among women, impacting 2.1 million women each year. In 2019, an estimated number of deaths.

Breast cancer typically has no symptoms when the tumor is small and most easily treated. It can spread to under arm lymph nodes causing a lump or swelling, even before the original tumor is large enough to be noticed. This is why screening is important in the early detection. (Figure 1.4)

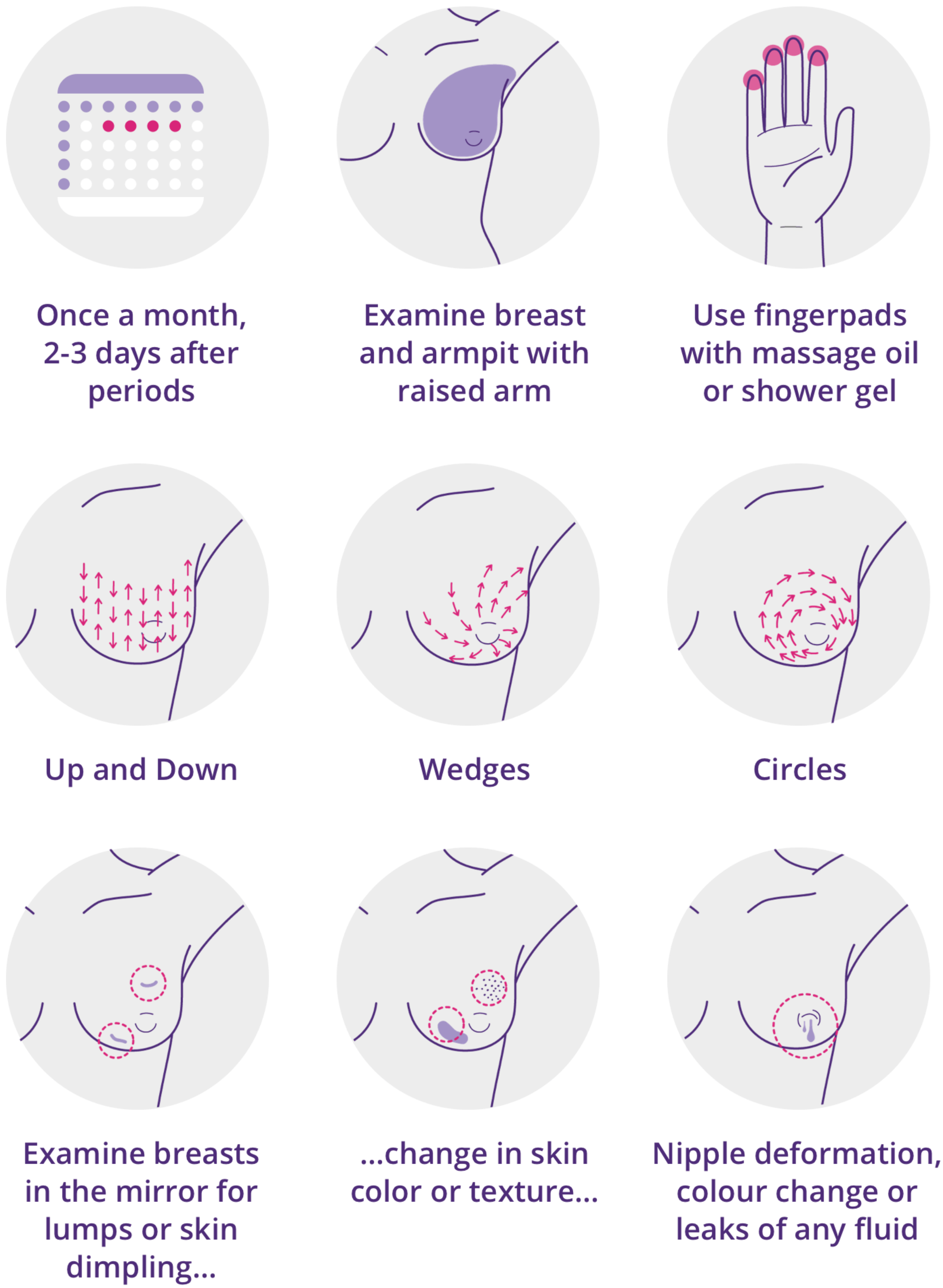

Figure 1.4 Breast Self-Examination[4].

1.2.2 Why Mammograms?

Nearly every woman knows about mammograms, which are x-rays of the breast. And while mammograms are highly recommended as the best method of detecting breast cancer, they don't find every breast cancer.

MRI (magnetic resonance imaging) uses magnetic fields, radio waves, and computers to create images of the inside of the body. The technique is used to check abnormal areas seen on a mammogram or any suspicious areas after breast surgery or radiation therapy. It's also helpful sometimes for finding abnormalities in younger women who have dense breast tissue.

On the downside, MRIs are less specific than mammograms. This means MRIs give a higher number of "false positives." This may cause women to get biopsies and other tests they don't need.

For most women, MRIs are not recommended for breast screenings. However, studies show that MRIs may be useful for women at high risk for breast cancer. For example, if a woman carries the genetic mutations BRCA1 and BRCA2,

➢ **Disadvantages of MRI**

The widespread use of breast MRI for the detection of breast malignancy also has many disadvantages, as follows:

- High equipment and examination costs
- Limited scanner availability
- Need for the injection of a contrast agent
- No standard technique
- Poor throughput compared with that of ultrasonography or mammography
- Large number of images
- Long learning curve for interpretation
- False-positive enhancement in some benign tissues (limited specificity)
- Variable enhancement of in situ carcinoma
- A 5% incidence of slowly or poorly enhancing invasive carcinomas

1.2.3 What Does the Doctor Look for on a Mammogram?

- **Masses**

It is an area of dense breast tissue with a shape and edges that make it look different than the rest of the breast tissue. With or without calcifications, it’s another important change seen on a mammogram. it can be many things, including cysts (non-cancerous, fluid-filled sacs) and non-cancerous solid tumors (such as fibro adenomas), but they may also be a sign of cancer.

Cysts are fluid-filled sacs. Simple cysts (fluid-filled sacs with thin walls) are not cancer and do not need to be checked with a biopsy. If a mass is not a simple cyst, it is of more concern, so a biopsy might be needed to be sure it is not cancer.

Solid masses can be more concerning, but most breast masses are not cancer.

A cyst and a solid mass can feel the same. They can also look the same on a mammogram. The doctor must be sure it is a cyst to know it is not cancer. To be sure, a breast ultrasound is often done because it is a better tool to see fluid-filled sacs. Another option is to use a thin, hollow needle to remove (aspirate) fluid from the area.

If a mass is not a simple cyst (that is, if it’s at least partly solid, or it has other concerning features), more imaging tests might be needed to decide if it could be cancer. Some masses can be watched over time with regular mammograms or ultrasound to see if they change, but others may need to be checked with a biopsy. The size, shape, and margins (edges) of the mass can help the radiologist decide how likely it is to be cancer.

- **Breast Calcifications**

are small calcium deposits that develop in a woman's breast tissue. They are very common and are usually benign (noncancerous). In some instances, certain types of breast calcifications may suggest early breast cancer.

There are two types of breast calcifications: macrocalcifications and microcalcifications.

Macrocalcifications look like large white dots on a mammogram (breast X-ray) and are often dispersed randomly within the breast. Macrocalcifications are common -- they are found in approximately half of women over age 50, and one in 10 women under age 50 -- and are considered noncancerous.

Microcalcifications are small calcium deposits that look like white specks on a mammogram. Microcalcifications are usually not a result of cancer. But if they appear in certain patterns and are clustered together, they may be a sign of precancerous cells or early breast cancer.

- **What is breast density?**

Breast density refers to the tissue composition of a breast. All breasts contain a mixture of fatty and glandular (dense) tissue. The more glandular tissue presents, the “denser” the breast is considered. When a woman has her mammogram, her breast density is assessed to be one of four categories. As shown in figure 1.5.

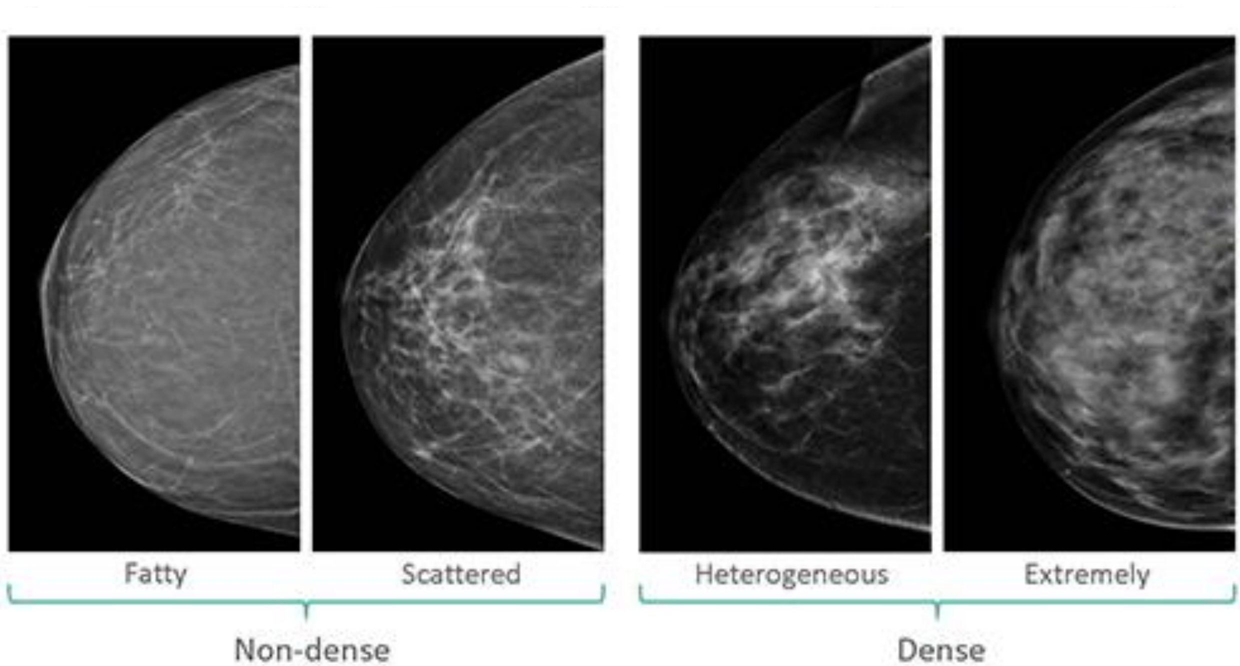

Figure 1.5 Tissue composition of a breast [5].

Heterogeneously and extremely dense, are considered “dense” breasts. As the figures below show, as the dense breast tissue increases, breast tissue appears increasingly white or light grey on a mammogram.

- **Why does density matter?**

Dense breast tissue displays as white or grey on a mammogram and, so does breast cancer. Trying to find a cancer in a dense breast has been compared to trying to find a snowball in a blizzard.

A cancer in a fatty breast is easier to detect as the fatty tissue displays darker on a mammogram providing good contrast to the whiter tumor. In a dense breast, little or no contrast may be present. So, breast density increases a mammogram's ability to show cancer decreases.

Breast density has another important implication: it also increases the risk of developing breast cancer. Women with dense breasts are both more likely to get breast cancer and to have that cancer be missed by mammography.

1.3 Statistics

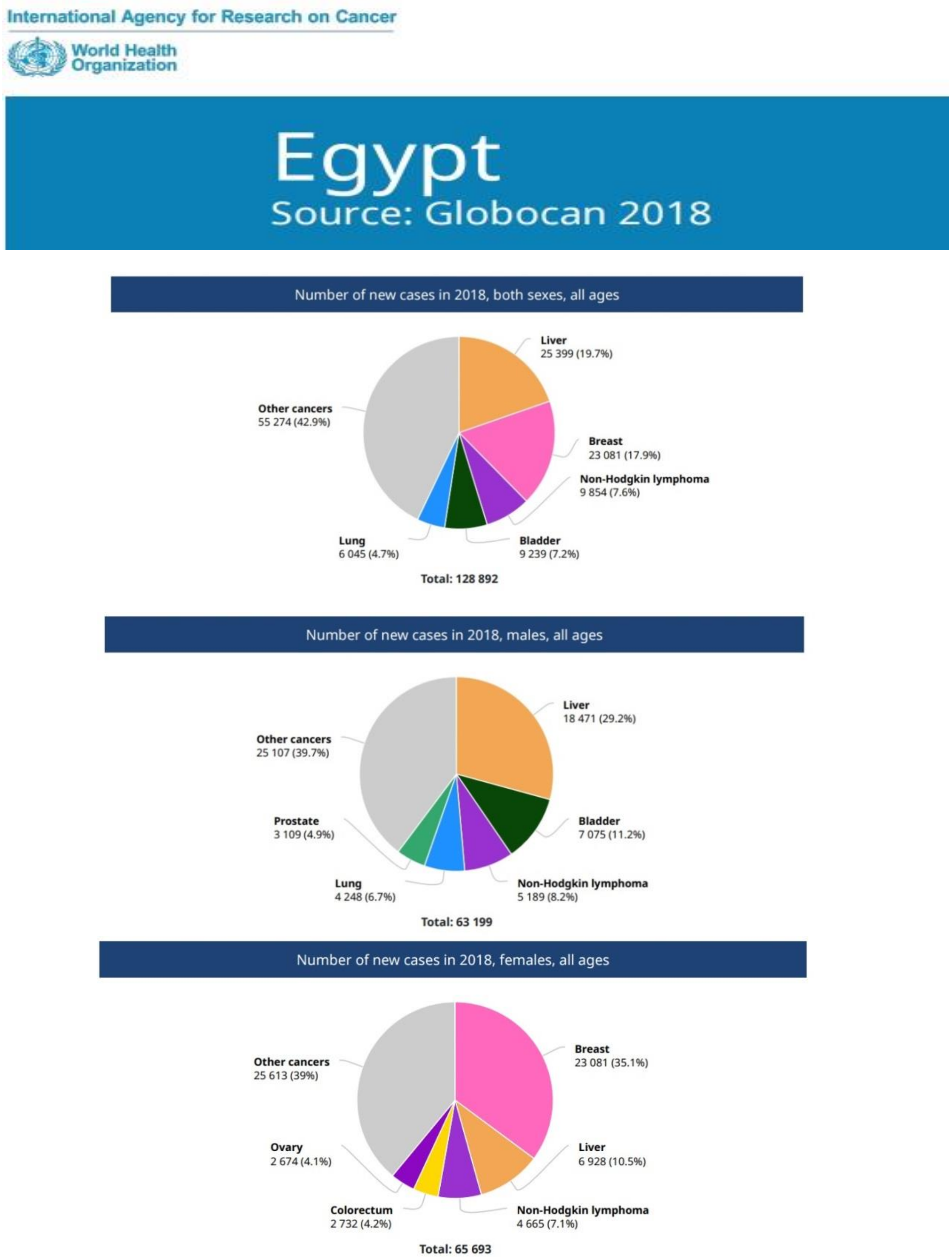

Figure 1.6 Comparing between numbers of new cases[6].

CHAPTER 2 LITERATURE REVIEW

It is difficult to interpret the mammogram to detect of lesions, it depends on level of experience for radiologists and on quality of image. there is Because of misinterpretations or overlooking for breast cancer signs. To overcome the problems associated with mammographic screening, double reading and Computer Aided Detection and Diagnosis (CAD). It combines between many steps: diagnostic imaging with computer science, image processing, pattern recognition, and artificial intelligence technologies. so, we consider that CAD is the second eye for radiologists, and It reduces the workload of radiologists and improve the sensitivity to detect breast cancer early. There are many CAD systems to detect and classify masses in the digital mammograms. The techniques used for developing these CAD systems classified to two: the first is consist of many steps such as pre-processing, segmentation, feature extraction, and classification steps, which entirely based on image processing and traditional machine learning techniques. the second category does not use any technique of feature extraction to detect the region of interest, but instead, it uses all information obtainable in the mammogram using the Convolutional Neural Network (CNN) to explore the features. There are many different methods for detection of breast cancer in DDSM dataset only:

Two support vector machine (SVM) classifiers are suggested by R. Campanini et al. In this method, suggested a new featureless method to detect mass in digital mammograms. It does not extract any feature for the detection of Region of Interest (ROI); but it uses all information available in the image. And for reducing the false positive rate, it used two Support Vector Machine (SVM) classifiers. they applied a multi-resolution over–complete wavelet representation to organize the image with increase information. And applied it to codify the image with increase information. Then the first SVM would identify it as suspect or not because vectors of a massive space obtained and provided. But, the first SVM made a false positive rate and reduce it by second SVM, then classify the input into a mass or non-mass regions. And by voting strategy, detect the suspect regions. In this method, it is used DDSM dataset / 2620 images, achieved in performance: 80% sensitivity with a false positive rate of 1.1 per image on mammograms [12].

CAD system by Twin SVM classifier that are suggested by Si and Jing. In this method, introduced a CAD system for detection and classification for mass of breast cancer based on a Twin SVM classifier. First, a mammogram image is concentrated using a Dyadic Wavelet-based algorithm. then extract ROI after removing the unwanted noise from a mammogram by method of segmentation (combining the Dyadic Wavelet information with mathematical morphology). Then, the suspect regions were segmented based on an optimal threshold value corresponding to the minimum fuzzy entropy. Then, features are extracted from regions of segmented suspect that using Gray Level Differences Statistics (GLDS) and Spatial Gray Level Dependence (SGLD) features. In this method, it is used DDSM dataset / 2620 images, they trained and tested the Twin SVM classifier for classification of masses. The authors reported that the sensitivity of the proposed system is 89.7% with a 0.31 false positive per image [13].

Support vector machine (SVM) and texture analysis are suggested by Eddaoudi et al. In this method, used SVM and texture analysis for a mass detection. It classifies ROI in three stages: in the first, the segmentation of the pectoral muscle is carried out using a method based on contour detection using snakes with automatic initialization. the second stage calculate maxima thresholding and Haralik features from the co-occurrence matrix for segmentation of ROI. In the third, use an SVM classifier to detect if the extracted features are normal or mass. It is used DDSM dataset / 2620 images, where achieved 95% on accuracy, when applied classification on the pre-segmented mammograms [14].

Developed CAD system is suggested by Jen and Yu. In this method, developed a CAD system using a two-stage classifier to detect abnormal mammograms. first, the Abnormal Detection Classifier (ADC) which applies the Principle Component Analysis (PCA). it used techniques of image processing enhancement to overcome the complication of the ROI detection in mammograms, it is used to remove the unwanted noise, non-breast regions for example the background, and the spectral muscle. Enhancement of image leads to detect abnormal areas in mammogram more effectively and exactly. Second, After the pre-processing step, use the gray level quantization to quantize all ROIs in mammograms and then extract a small number of important features. All features are extracted, classified as normal or abnormal using the ADC. It is used DDSM dataset / 2620 images in this method, the sensitivity was 88% and specificity was 84% on MIAS database after testing the ADC for 322 images [15].

Deep learning visual search system is suggested by Ertosun and Rubin. In this method, developed a deep learning visual search system to classify and localize mass in mammograms which comprises two modules: the first to classify the whole mammogram image into two classes (mass and non-mass). While the second to localize mass(es) in mammogram images using a regional probability approach based on a deep learning network. It is used DDSM dataset / 2620 images in this method, achieved in performance: 85% sensitivity in the classification and 85% in the localization of the masses at an average of 0.9 false positives per image [16].

Convolutional neural network (CNN) is implemented by Jadoon et al. In this method, suggested using the CNN for a three-class (normal, malignant, and benign) mammogram classification. This work presented two algorithms: the first based on Discrete Wavelet Transform (CNN-DW); the second bases on Curvelet Transform (CNN-CT). it is more helpful for detection of cancer to extract the features from the mammogram and using them as an input to CNN. It is used DDSM dataset / 2620 images in this method, achieved in performance: CNN-DW and CNN-CT achieved an accuracy rate of 81.83% and 83.74%, respectively [17].

AlexNet and GoogleNet (transfer learning) are suggested by Levy and Jain. In this method, cleared up the benefit of Deep Learning as a classification tool to differentiate between regions of benign and malignant cancerous. We train three different architectures of CNN for breast mass classification and analyze the effect of several model choices. The Three network architectures: a shallow CNN (the baseline model), an AlexNet and a GoogLeNet. We use the same base architecture for AlexNet and GoogleNet, but the last layer of fully connected (FC) is replaced. Also, in GoogleNet, two auxiliary classifiers are removed which we found damaged our training. The early layers of AlexNet are used to inspire the architecture of the baseline model. It is used DDSM dataset / 1820 images in this method, achieved in performance: Accuracy (0.924), precision (0.924) [18].

Multistage fine-tuned CNN are suggested by Samala et al. In this method, presented method of mass classification for digital breast tomosynthesis (DBT) by using multistage fine-tuned CNN, where it is used approach of multistage transfer learning by using different layer variation and selecting the optimal combination. After using cross validation, we found the best transfer network: it is from six transfer networks by changing the level up to which the convolutional layers were frozen. It is used DDSM dataset / 4039 ROIs in this method, achieved in performance: AUC (0.91) [19].

GAN and ResNet50 are suggested by Wu et al. In this method, introduced approach of deep learning (DL) for treatment of category imbalance and to limit of data issues for Detection and classification of benign and malignant calcifications and masses. This way used an infilling approach to generate artificial mammogram spots by using conditional generative adversarial network(cGAN). As shown in Figure 2.1, Initially the multi-domain generator is trained to create artificial patches in the target image using the GAN. The multiscale features are generated by the generator used a cascading improvement to ensure stability at high resolution. The cGAN would infill only lesion in mass or calcifications. It is used DDSM dataset / 10480 images in this method, achieved in performance: AUC (0.896) [20].

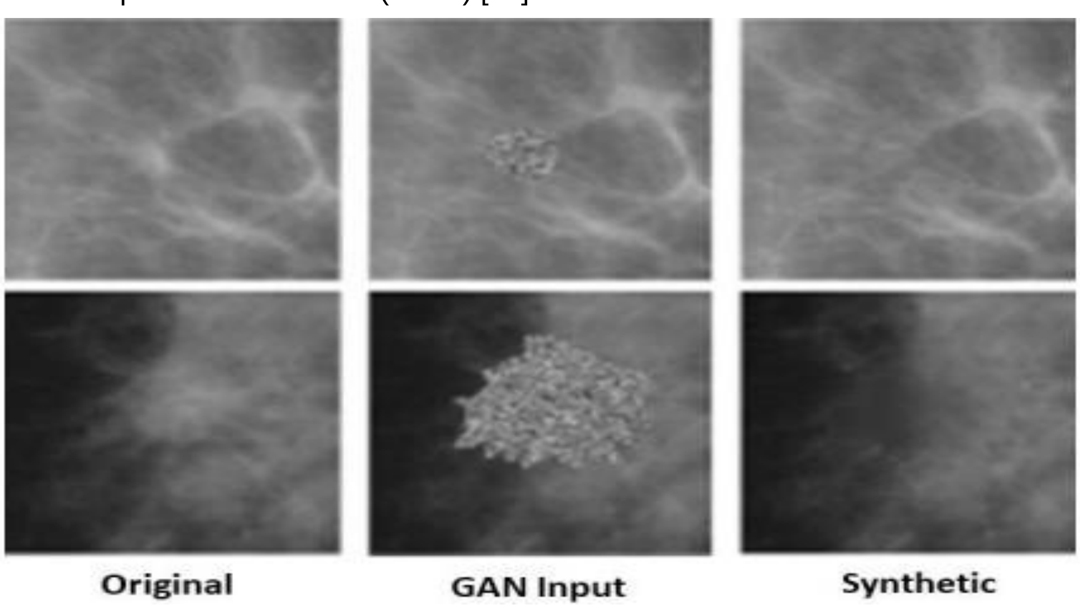

Figure 2.1 The synthetic images generated by cGAN [20].

CHAPTER 3 DEEP LEARNING OVERVIEW

Artificial Intelligence is a branch of computer science that endeavors to replicate or simulate human intelligence in a machine, so machines can perform tasks that typically require human intelligence. Some programmable functions of AI systems include planning, learning, reasoning, problem solving, and decision making.

AI is generally divided into 3 categories

1. Narrow/Weak AI is a type of Artificial Intelligence focused on one single narrow task. It possesses a narrow-range of abilities. This is the only AI in existence today, for now

2. General/strong AI When we talk about Artificial General Intelligence (AGI) we refer to a type of AI that is about as capable as a human It seems unlike that this will happen relatively soon because there is lack of a comprehensive knowledge of the functionality of the human brain.

3. Superintelligence To reach this point and to be called an ASI, an AI will need to surpass humans at absolutely everything. The ASI type is achieved when AI is more capable than a human[21].

Machine-learning technology powers many aspects of modern society: from web searches to content filtering on social networks to recommendations on e-commerce websites. Conventional machine-learning techniques were limited in their ability to process natural data in their raw form. The performance of machine learning methods is heavily dependent on the choice of data representation (or features) on which they are applied.

feature engineering is important but labor intensive and highlights the weakness of current learning algorithms: Their inability to extract and organize the discriminative information from the data.

ML algorithms evolve as they are exposed to more data. Nearly all ML algorithms used to analyze the pixel data of radiology examinations ‘‘learn’’ to give a specific answer by evaluating a large number of exams that have been hand-labeled. For example, a ML algorithm to detect lung nodules on chest Radiographic will be trained by analyzing thousands of chest radiographs that humans have labeled as being normal, or as having nodules in the lungs [21].

Representation learning refers to a subtype of ML in which no ‘‘hand-crafted’’ features are provided. Instead, the computer algorithm learns the features required to classify the provided data. Representation learning is a set of methods that allows a machine to be fed with raw data and to automatically discover the representations needed for detection or classification [25].

Deep learning refers to a subfield of representation learning which relies on multiple processing layers (hence, deep) to learn representations of data with multiple layers of abstraction. The various layers in these algorithms are used to detect features ranging from simple (e.g., lines, edges, textures, intensity) to complex (e.g., shapes, lesions, or whole organs) in a hierarchical structure by composing simple but non-linear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level [21].

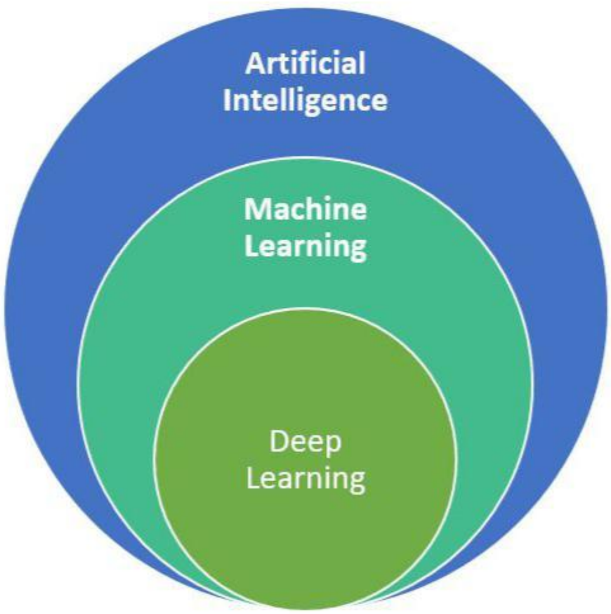

Figure 3.1 AI hierarchy [29].

It turns out that if you use a traditional AI system, then the performance would grow, that as you feed in more data performance gets a bit better. But beyond a certain point it did not get that much better.

But with modern AI, with neural networks and deep learning, if you train a small neural network, then the performance improves a little, and as you feed them more data, performance keeps getting better for much longer. If you train an even slightly larger neural network, say medium-sized neural net, then the performance may improve much as shown in (figure 3.2). If you train a very large neural network, then the performance just keeps on getting better and better [51].

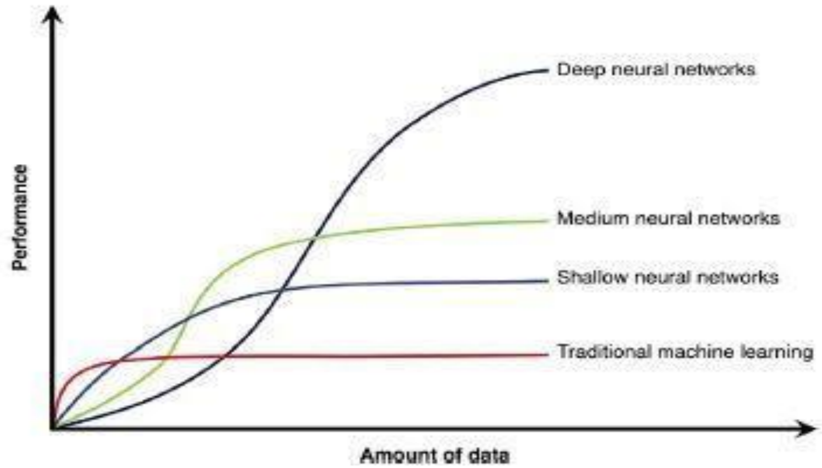

Figure 3.2 The relation between amount of data and performance [27].

3.1 Artificial neural networks

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.

An artificial neural network is a collection of simple artificial neurons connected by directed weighted connections. When the system is set running, the activation levels of the input units is clamped to desired values.

The first artificial neuron was produced in 1943 by the neurophysiologist Warren McCulloch and the logician Walter Pits. But the technology available at that time did not allow them to do too much.

Neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly. Since the network finds out how to solve the problem by itself.

All physical and mental functioning depends on the establishment and maintenance of neuron networks. Connections the brain finds useful become permanent. A neuron consists of soma (cell body), axon (a long fiber) and dendrites. Given a signal, a synapse might increase (excite) or decrease (inhibit) electrical potential. A neuron fires when its electrical potential reaches a threshold. As shown in figure 3.3.

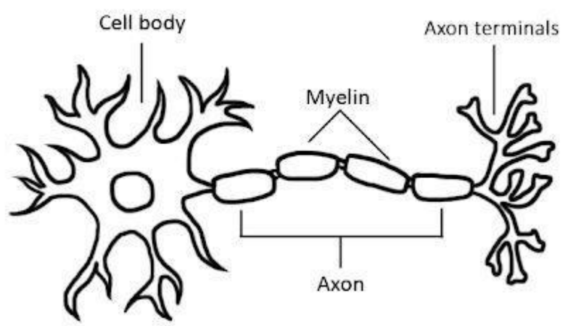

Figure 3.3 Anatomy of Nervous System and endocrine systems [41].

An artificial neural network consists of a number of neurons (units) similar to the biological neurons in the brain (often arranged in layers), a number of connections which are performed by weighted links and whose role is to transmit signals from one neuron to another, and weights. The output signal is transmitted through the neuron's outgoing connection (analogue to the axon in the biological neurons). The outgoing connection splits into a number of branches that transmit the same signal. As shown in figure 3.4.

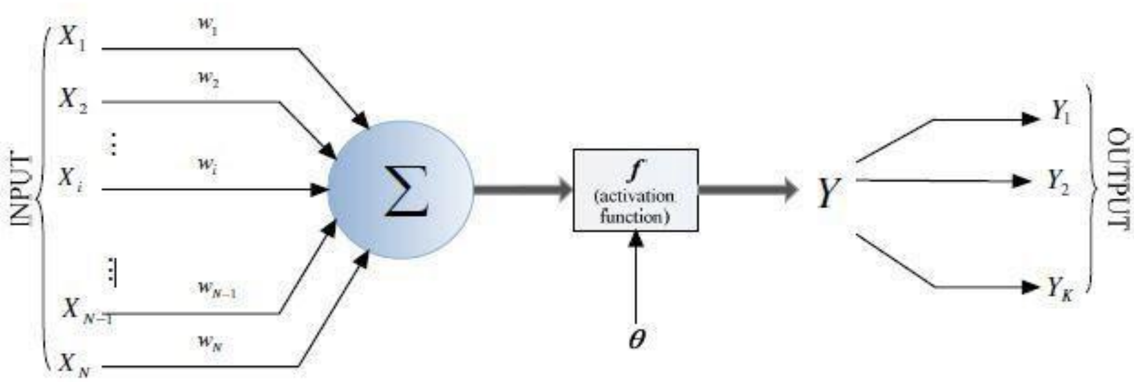

Figure 3.4 Principle of perceptrons [34].

How do perceptron work? A perceptron takes several binary inputs, x1, x2, . . ., and produces a single binary output

- a set of processing units (called neurons or cells);
- a state of activation Yi for every unit, which is equivalent to the output of the unit;
- connections between the units; each connection is defined by a weight wjk which determines the effect which the signal of unit j has on unit k. The contribution for positive wjk is considered as an excitation and for negative wjk as inhibition.
- a propagation rule, which determines the effective input Xi of a unit from its external inputs;
- an activation function f, which determines the new level of activation based on the effective input Xi(t) and the current activation Yi(t);
- an external input (also known as bias, offset) θi for each unit.
- a method for information gathering (the learning rule);
- an environment within which the system must operate, providing input
- signals and / if necessary / error signals.[28]

In analogy, the bias nodes are like the offset in linear regression given as; = +, where "a" is the coefficient of independent "x" and then "b" is called slope. A bias major function is to provide node with a constant value that is trainable, in addition to the normal inputs received by the network node.[41]

There are 3 types of perceptron: (Figure 3.5)

- Input units.
- Hidden units.
- Output units.

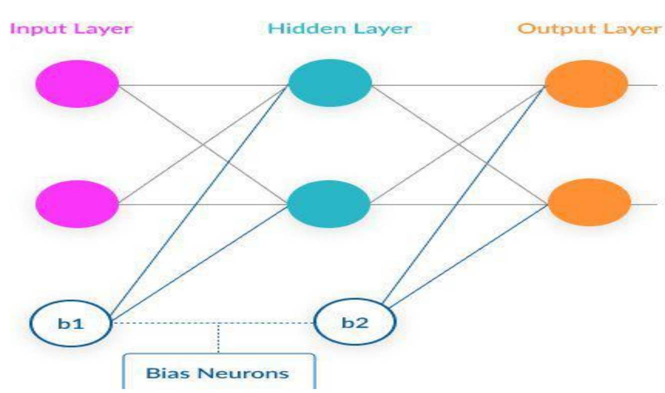

Figure 3.5 Types of perceptron convolutional design [23].

The neurons receive input from their neighbors or external sources and use this to compute an output signal which is propagated to other units. Apart from this processing, a second task is the adjustment of the weights. The system is inherent parallel in the sense that many units can carry out their computations at the same time.

The neural networks can be classified depending on:

- the nature of information processing carried out at individual nodes:
  - single layer network (perceptron);
  - multi-layer network;
- the connection geometries:
  - feedforward network;
  - backpropagation network;
- the algorithm for adaptation of link weights [28].

3.2 Key Concepts of Deep Neural Networks

3.2.1 Activation Functions

The activation function is a mathematical "gate" in between the input feeding the current neuron and its output going to the next layer. It can be as simple as a step function that turns the neuron output on and off, depending on a rule or threshold. Or it can be a transformation that maps the input signals into output signals that are needed for the neural network to function. The Activation Functions can be basically divided into 3 types:

A) Linear Activation Function

A linear activation function takes the form: (Fig. 3.6)

$$A = cx$$

however, a linear activation function has two major problems:

1) Not possible to use backpropagation.
2) All layers of the neural network collapse into one.

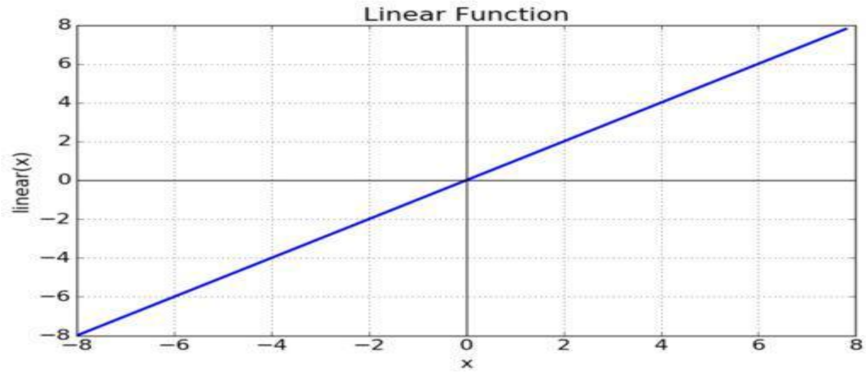

Figure 3.6 Linear activation Function [34].

B) Binary Step Function

If the value of the sum of the weighted input is more than threshold, the neuron is fired.

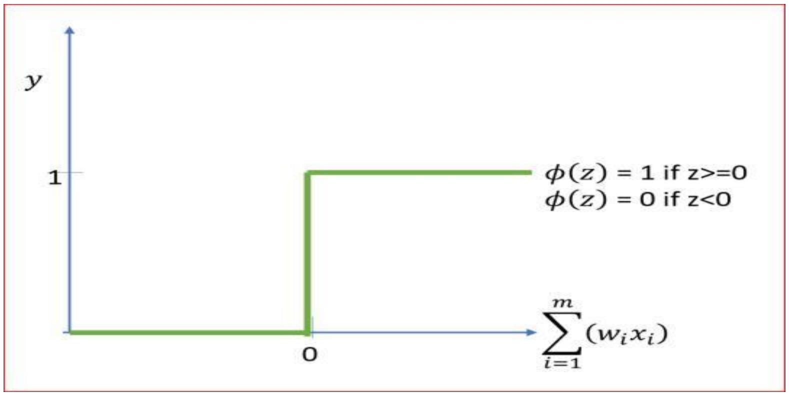

Figure 3.7 Binary Step Function [34].

C) Non-linear Activation Functions

i) Sigmoid / Logistic

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$$

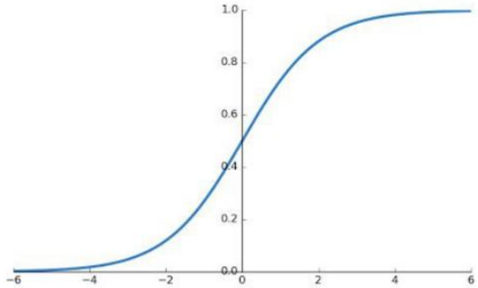

Figure 3.8 Sigmoid / Logistic [34].

Advantages

- Smooth gradient, preventing “jumps” in output values.
- Output values, bound between 0 and 1, normalizing the output of each neuron.
- Clear predictions, For X above 2 or below -2, tends to bring the Y value (the prediction) to the edge of the curve, very close to 1 or 0. This enables clear predictions.

Disadvantages

- Vanishing gradient—for very high or very low values of X, there is almost no change to the prediction, causing a vanishing gradient problem. This can result in the network refusing to learn further, or being too slow to reach an accurate prediction.
- Outputs not zero centered.
- Computationally expensive [28].

ii) ReLU (Rectified Linear Unit)

The rectifier is an activation function defined as positive part of its argument:

$$f(x) = x^{+} = \max(0, x),$$

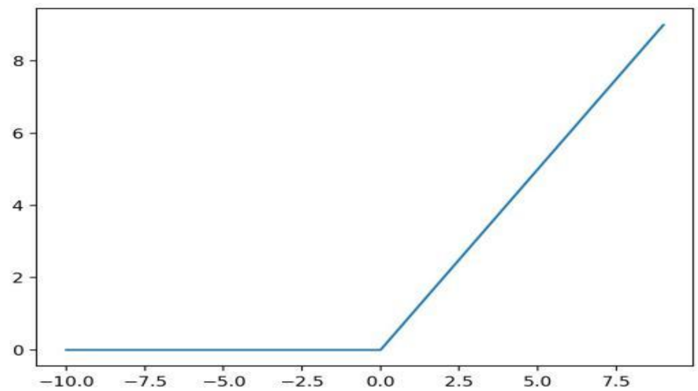

Figure 3.9 ReLU [28].

Advantages

- Computationally efficient—allows the network to converge very quickly.
- Non-linear—although it looks like a linear function, ReLU has a derivative function and allows for backpropagation.

Disadvantages

- The Dying ReLU problem—when inputs approach zero, or are negative, the gradient of the function becomes zero, the network cannot perform backpropagation and cannot learn.

iii) Softmax

- Able to handle multiple classes only one class in other activation functions—normalizes the outputs for each class between 0 and 1, and divides by their sum, giving the probability of the input value being in a specific class.

- Useful for output neurons—typically Softmax is used only for the output layer, for neural networks that need to classify inputs into multiple categories [28]. As shown in figure 3.10.

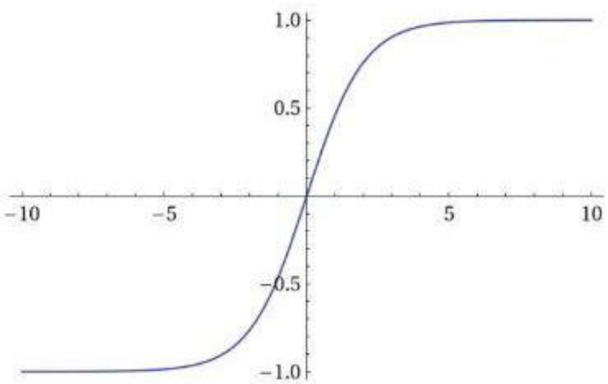

Figure 3.10 Softmax Activation Function [28].

The ReLU and the Softmax activation functions are the dormant AFs used in practical DL applications. Furthermore, the Softmax function is used in the output layer of most common practice DL applications however, the most recent DL architecture used the sigmoid function to achieve it prediction at the output layer, while the ReLU units are used in the hidden layers.

The main difference between the Sigmoid and Softmax AF is that the Sigmoid is used in binary classification while the Softmax is used for multivariate classification tasks.

There are other types of activation functions like Gaussian, TanH, SoftPlus, Exponential linear unit (ELU) so on [29].

3.2.2 Gradient descent

In figure 3.11, Gradient Descent is a process that occurs in the backpropagation phase where the goal is to continuously resample the gradient of the model's parameter in the opposite direction based on the weight w, updating consistently until reaching the global minimum of function J(w).

To put it simply, gradient descent is used to minimize the cost function, J(w).

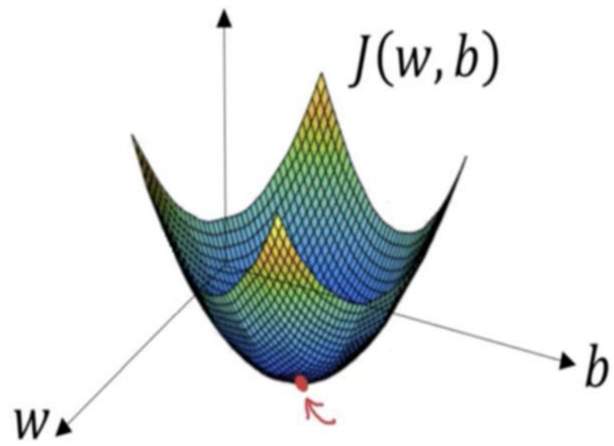

Figure 3.11 Gradient descent [44].

The gradient descent process is exhibited in the form of the backpropagation step where the error vectors δ are computed backward, starting from the final layer. Depending upon the activation function, we identify how much change is required by taking the partial derivative of the function with respect to w. The change value gets multiplied by the learning rate. As part of the output, this value is subtracted from the previous output to get the updated value. We continue this till reaching convergence [30].

Slope of activation function

In Figure 3.12 , When constructing Artificial Neural Network (ANN) models, one of the key considerations is selecting an activation functions for hidden and output layers that are differentiable This is because calculating the backpropagation error is used to determine ANN parameter updates that require the gradient of the activation function for updating the layer.

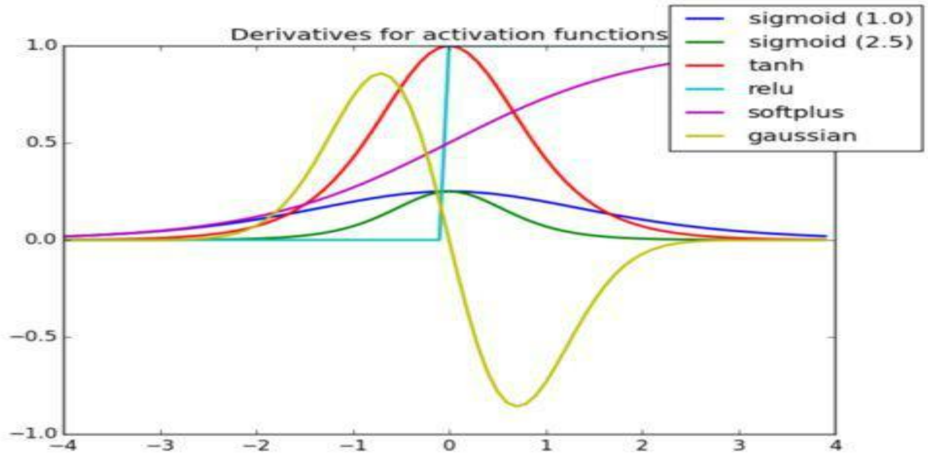

Figure 3.12 Derivatives for activation function.

3.3 Convolutional neural network

For some types of data, especially for images, multilayer perceptron is not well adapted. Indeed, they are defined for vectors as input data, hence, to apply them to images, we should transform the images into vectors, losing by the way the spatial information contained in the images, such as forms. Before the development of deep learning for computer vision, learning was based on the extraction of variables of interest, called features, but these methods need a lot of experience for image processing.
CNN act directly on matrices, or even on tensors for images with three RGB color channels.
CNN are now widely used for image classification, image segmentation, object recognition, face recognition.

Deep convolutional network is more successful over traditional fully-connected neural networks in working with two-dimensional data because of three factors

- Sparse connectivity, in a fully connected neural network every pixel (i.e. node in input layer) of an image is connected to all nodes of the following layer. However, in CNN, the dimension of the kernel is smaller than that of the input, which means that we have sparse interactions.

- Parameter sharing is key to CNNs because it significantly decreases the number of parameters in the network in comparison to traditional neural networks. In CNN, the convolution layer has tied weights, which means that the same weights are used for each node in the layer.

- Equivariant representations, the convolution has also the property called equivariance, which means the order of transformation and convolution operations does not make difference. Transformation is any kind of change applied on input image in practice, this means that even if a car in an image moves from left bottom corner to right upper corner, the convolutional network will still classify it correctly [40].

3.3.1 Layers in a CNN

A) Convolution layer

The principle of 2D convolution is to drag a convolution kernel on the image. At each position, we get the convolution between the kernel and the part of the image that is currently treated. As shown in figure 3.13.

Then, the kernel moves by a number s of pixels s is called the stride. Assume that we apply C0 kernels (also called filters), each of size k on an image. If the size of the input image is Wi Hi Ci (Wi denotes the width, Hi the height, and Ci the number of channels, typically Ci = 3), the volume of the output is W0 H0 C0, where C0 corresponds to the number of kernels that we consider, and

$$W_0 = \frac{W_i - k + 2p}{s} + 1$$

$$H_0 = \frac{H_i - k + 2p}{s} + 1.$$

Convolutional net owes its name to a mathematical operation convolution. Convolution is denoted with a star symbol (*) and can be represented as

s(t) = (x* w) (t)

where x is referred to as the input vector and was weight vector or kernel. The result s(t) is known as the feature map. This process can be seen [40]

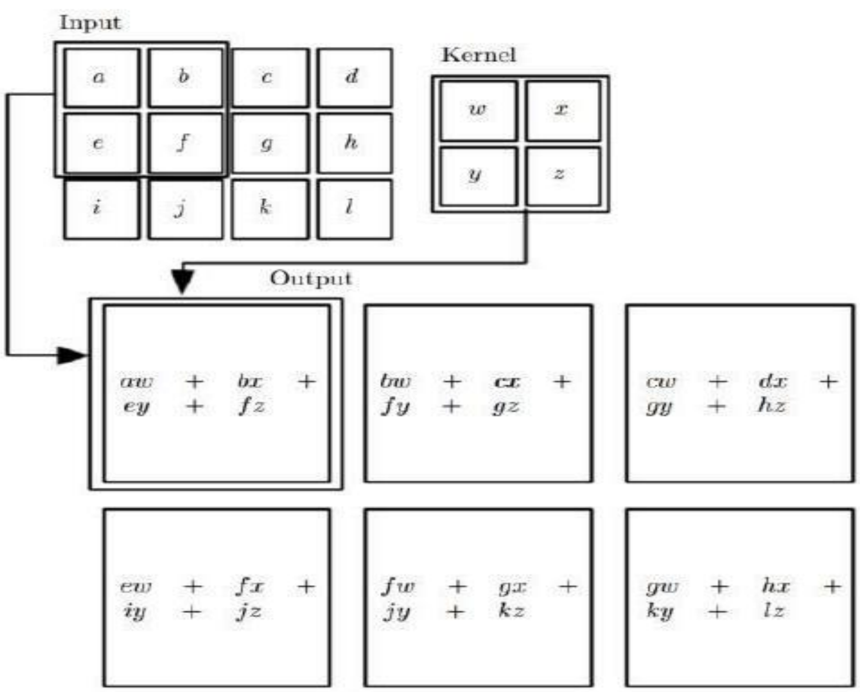

Figure 3.13 The Principle of 2D convolutional [28].

Sometimes, we also add a zero padding, which is a margin of size p containing zero values around the image to control the size of the output. it is a way tocontrol the shrinkage of dimension after applying filters larger than 1x1, and avoid losing information at the boundaries, there are types of padding:

1. Padding same that means when you pad, so the output size is the same as the input size.
2. Padding valid means no padding.

In figure 3.14, Convolution of an image with different filters can perform operations such as edge detection (can be horizontal, vertical, diagonal), blur and sharpen by applying filters. The below example shows various convolution image after applying different types of filters.

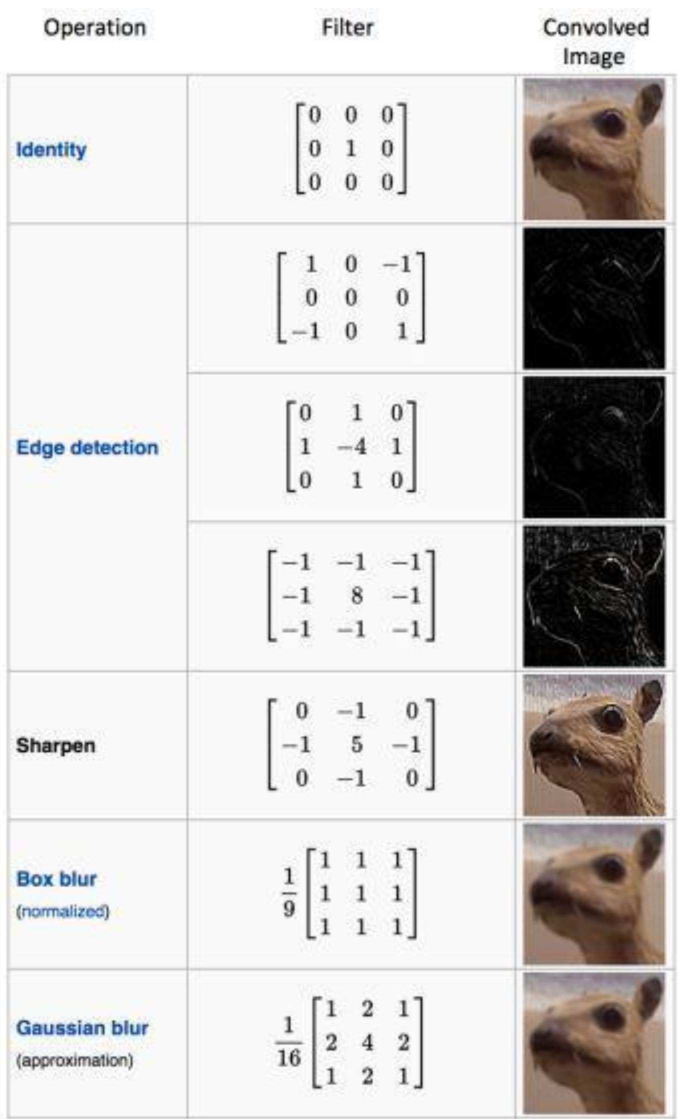

Figure 3.14 Convolution of an image [27].

B) Pooling layer

CNN also have pooling layers, which allow to reduce the dimension, also referred as subsampling, Operation of a pooling allows to reduce the spatial volume of the image in order to decrease the number of parameters and calculations in the network. In addition, altering already unnecessary details helps to reduce overfitting.[39]

Pooling layer's main role is to detect conjunctions of features(patterns) occurring in the previous layer. In other words, its role is to combine similar features into one pattern. If some features have already been detected in the previous folding operation, then a detailed image is no longer needed for further processing. Two major pooling techniques (functions) exist: As shown in figure 3.15.

1. Max-pooling reports the average output of a rectangular neighborhood.
2. Average-pooling it extracts image features smoothly and generalize computation by bringing all pixels into count.

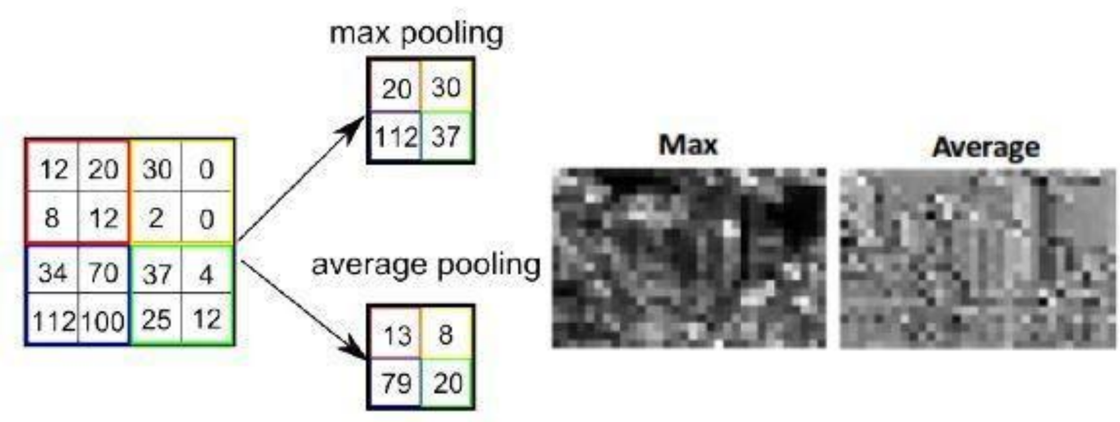

Figure 3.15 Example of pooling layer [27].

Another advantage of the pooling is that it makes the network less sensitive to small translations of the input images.

C) Fully connected layers

After several convolution and pooling layers, the CNN generally ends with several fully connected layers. The tensor that we have at the output of these layers is transformed into a vector (flattened) and then we add several perceptron layers. Over a series of epochs, the model is able to distinguish between dominating and certain low-level features in images and classify them using the Softmax Classification technique.[39]

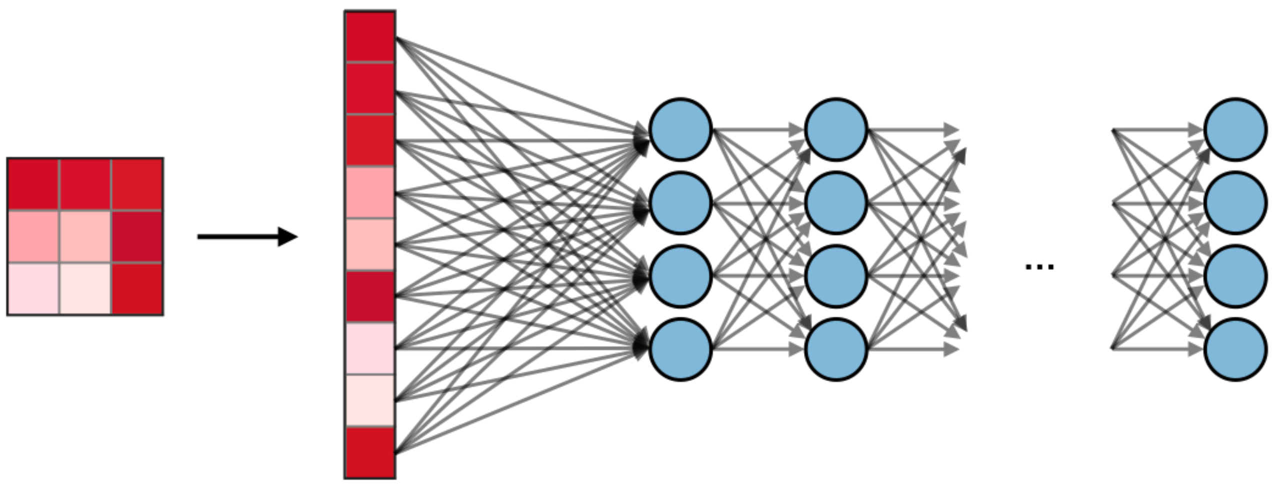

Figure 3.16 Example of fully connected layer [31].

3.3.2 Architectures

We have described the different types of layers composing a CNN. We now present how these layers are combined to form the architecture of the network.

Choosing an architecture is very complex and this is more engineering that an exact science.

1. LeNet-5

   Architecture: LeNet-5 has 2 convolutional and 3 fully connected layers. It has trainable weights and a sub-sampling layer (now known as the pooling layer). LeNet5 has about 60,000 parameters.

2. AlexNet

   Architecture: AlexNet has 8 layers, 3 fully-connected and 5 convolutional.

   AlexNet had 60 million parameters.

   Year of Release: 2012

3. VGG-16

Architecture: VGG-16 has 13 convolutional and 3 fully-connected layers. It used ReLUs as activation functions, just like in AlexNet. VGG-16 had 138 million parameters. A deeper version, VGG-19, was also constructed along with VGG-16.

Year of Release: 2014

4. Inception-v3

Architecture: A successor to Inception-v1, Inception v-3 had 24 million parameters and ran 48 layers deep.

Year of Release: 2015

5. Resnet50

Year of Release: 2015

Residual block

In traditional neural networks, each layer feeds into the next layer. In a network with residual blocks, each layer feeds into the next layer and directly into the layers about 2–3 hops away. As shown in figure 3.17.

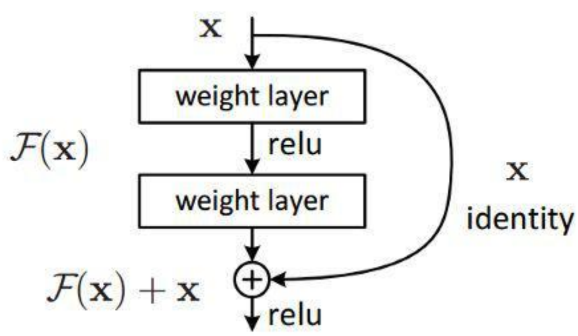

Figure 3.17 Residual Block [26].

The accuracy increases with increasing number of layers. But there is a limit to the number of layers added that result in accuracy improvement This phenomenon is referred to as the degradation problem.

You can skip the training of few layers using skip-connections or residual connections. You can think of this residual function as a refinement step in which we learn how to adjust the input feature map for higher quality features. This compares with a "plain" network in which each layer is expected to learn new and distinct feature maps. If no refinement is needed, the intermediate layers can learn to gradually adjust their weights toward zero such that the residual block represents an identity function [26].

Architectures

Figure 3.18 Deep Residual learning for image recognition [52].

Parameters: 25 million (ResNet 50)

Each colored block of layers represents a series of convolutions of the same dimension. The feature mapping is periodically down sampled by striped convolution accompanied by an increase in channel depth to preserve the time complexity per layer. Dotted lines denote residual connections in which the input via a 1x1 convolution is projected to match the dimensions of the new block.[26]

The diagram above visualizes the ResNet 34 architecture. For the ResNet 50 model, we simply replace each two-layer residual block with a three-layer bottleneck block which uses 1x1 convolutions to reduce and subsequently restore the channel depth, allowing for a reduced computational load when calculating the 3x3 convolution. As shown in Figure 3.18.

Programing language

There are many Programing languages used for AI field including

(Python, R, Java, Prolog, C++)

Python

Python is considered to be in the first place in the list of all AI development languages due to the simplicity. The syntaxes belonging to python are very simple and can be easily learnt. Therefore, many AI algorithms can be easily implemented in it. Python takes short development time in comparison to other languages like Java, C++ or Ruby. Python supports object oriented, functional as well as procedures-oriented styles of programming. There are plenty of libraries in python, which make our tasks easier.

It supports deep learning libraries like MXNet, Keras and TensorFlow.

Tensorflow

Created by the Google Brain team, TensorFlow is an open source library for numerical computation and large-scale machine learning. TensorFlow bundles together a slew of machine learning and deep learning models and algorithms and makes them useful by way of a common metaphor. It uses Python to provide a convenient front-end API for building applications with the framework, while executing those applications in high-performance C++.

Keras

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides.

Keras offers the advantages of broad adoption, support for a wide range of production deployment options, integration with at least five back-end engines (TensorFlow, CNTK, Theano, MXNet, and PlaidML)

Keras proper does not do its own low-level operations, such as tensor products and convolutions; it relies on a back-end engine for that. Even though Keras supports multiple back-end engines, its primary (and default) back end is TensorFlow.

Deployment of mobile net model using flask:

Mobile net model is faster in performance over all the rest of networks like resnet50 and vgg16 that we used in our procedures so, we chose mobile net to launch our web app using flask which is a popular Python web framework, meaning it is a third-party Python library used for developing web applications. It designed to make getting easy, with the ability to scale up complex applications, and has become one of the most popular Python web application frameworks.

The microframework Flask is based on the Pocoo projects Werkzeug and Jinja2.

Werkzeug

Werkzeug is a utility library for the Python programming language, in other words a toolkit for Web Server Gateway Interface (WSGI) applications, and is licensed under a BSD License. Werkzeug can realize software objects for request, response, and utility functions. It can be used to build a custom software framework on top of it and supports Python 2.7 and 3.5 and later.

Jinja

Jinja, also by Ronacher, is a template engine for the Python programming language and is licensed under a BSD License. Similar to the Django web framework, it handles templates in a sandbox.

CHAPTER 4 METHODOLOGY

4.1 Introduction

State-of-the-art models are pre-trained on large datasets like ImageNet to higher the performance of models by transfer learning [1]. To improve upon the state of the art, and create a model capable of diagnosing and classifying breast lesions efficiently, we followed some procedures. In this chapter, we will discuss these procedures one by one. But before talking about these procedures, we shall first talk about the chosen datasets and networks.

The block diagram in Figure 4.1 explains the flow of the project starting from the dataset, and ending with predicting the classes of new images.

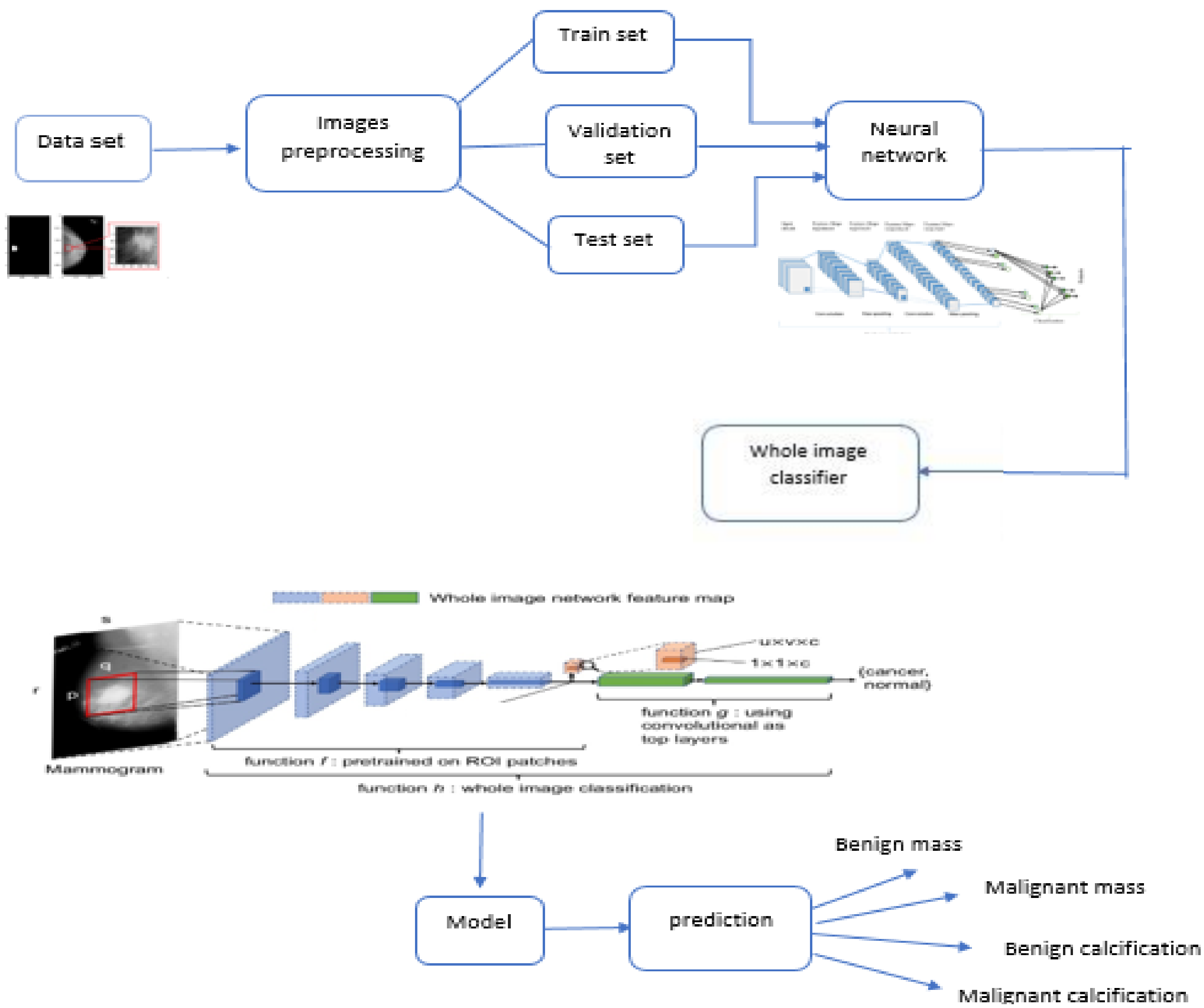

Figure 4.1 Block diagram of procedures for detection of breast cancer.

4.2 Dataset

The images from the CBIS-DDSM dataset are obtained, which is a curated breast imaging subset of the DDSM (digital database for screening mammography). CBIS-DDSM is a standardized and updates version of DDSM providing easily accessible data and improved ROI segmentation, which makes it suitable for evaluation of CAD systems research in mammography.

The CBIS-DDSM includes decompressed images, data selection and curation by trained mammographers, updated mass segmentation and bounding boxes, and pathologic diagnosis for training data.

It contains 753 calcification cases and 891 mass cases, after removing the questionable mass cases, having a total of 10239 benign and malignant images as DICOM files after removing.

Full mammography images provided are taken from two different views: craniocaudal (CC) and mediolateral (MLO) views, as well as two breast sides: left and right. Some cases include images of different times when the patient was examined more than once.

The dataset also includes the region of interest (ROI) after being cropped from the full mammogram lesion, provided with lesion annotations, and binary images as mask of the lesion(abnormalities). A sample image can be seen in Figure 4.2.

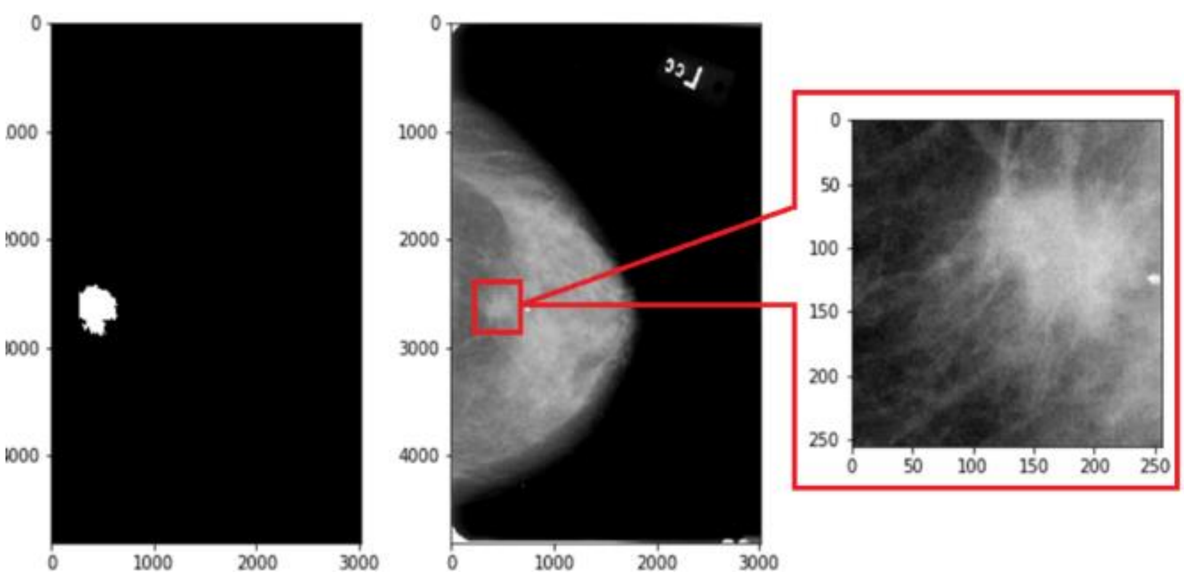

Figure 4.2 The full mammogram in the middle with its cropped ROI on the right and the abnormality as a binary mask on the left [56].

The CBIS-DDSM images are then split into training and testing sets according to difficulty based on the BI-RADS assessment as shown in Figure 4.3 and Table 4.1.

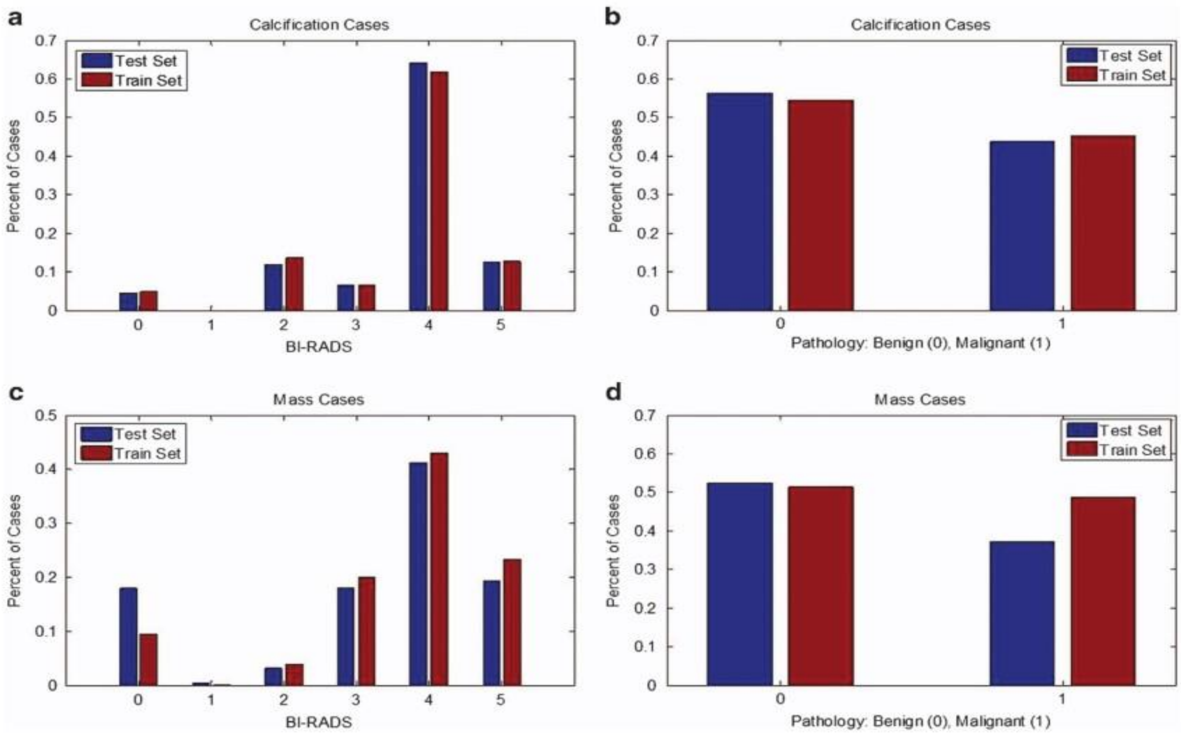

Figure 4.3 Histograms showing the reading difficulties for the training and test sets [56].

(a) Histogram of BI-RADS for each abnormality in training and test sets with calcifications, (b) Histogram of benign and malignant cases for training and test sets with calcifications, (c) Histogram of BIRADS for each abnormality in training and test sets with masses, (d) Histogram of benign and malignant cases for training and test sets with masses.

Table 4.1 Number of cases and abnormalities in the training and test sets [56].

	Benign Cases	**Malignant Cases**
Calcification Training Set	329 cases (552 abnormalities)	273 cases (304 abnormalities)
Calcification Test Set	85 cases (112 abnormalities)	66 cases (77 abnormalities)
Mass Training Set	355 cases (387 abnormalities)	336 cases (361 abnormalities)
Mass Test Set	117 cases (135 abnormalities)	83 cases (87 abnormalities)

These numbers are different as some cases contain more than one abnormality. There is some overlap since some cases contain both calcifications and masses. Metadata for each abnormality is included as an associated CSV file containing the following:

- Patient ID: the first 7 characters of images in the case file.
- Density category.
- Breast: Left or Right.
- View: CC or MLO.
- Number of abnormalities for the image (This is necessary as there are some cases containing multiple abnormalities).
- Mass shape (when applicable).
- Mass margin (when applicable).
- Calcification type (when applicable).
- Calcification distribution (when applicable).
- BI-RADS assessment.
- Pathology: Benign, Benign without call-back, or Malignant.
- Subtlety rating: Radiologists' rating of difficulty in viewing the abnormality in the image.
- Path to image files.

4.3 Networks

In this project, the final prediction models are created based on two state-of-the-art networks: ResNet50, and MobileNet.

- **ResNet50 [44], [45]**

  Residual network is a classical neural network used as a backbone for many computer visions tasks. This model was the winner of ImageNet challenge 2015. The fundamental breakthrough with ResNet was that It allowed us to train extremely deep neural networks with 150+ layers successfully. Prior to ResNet, training neural networks that have such great depths was difficult due to the problem of vanishing gradients. ResNet is frequently used as a starting point for transfer learning. Figure 4.4 shows the effect of the network depth on the percentage of error.

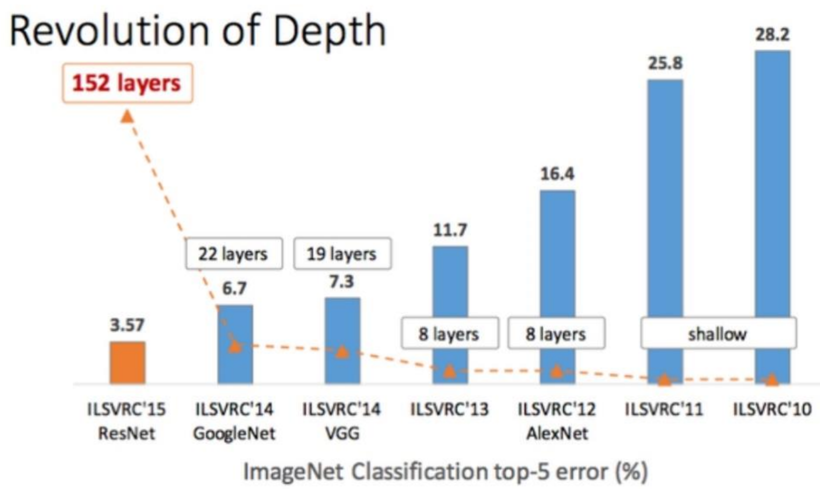

Figure 4.4 Depth to percentage error plot for state-of-the-art networks.

As the gradient is back-propagated to earlier layers, repeated multiplication may make the gradient extremely small. As a result, as the network goes deeper, its performance gets saturated and even starts degrading rapidly.

❖ MobileNet

MobileNets are a class of light weight deep convolutional neural networks that are smaller in size and faster in performance than many other popular models that can be used for classification, detection, and other common tasks convolutional neural networks are good for. these are considered great deep learning models to be used on mobile devices.

4.4 Dataset Preparation and Preprocessing

❖ Move and Rename DICOM Files

The dataset's DICOM files are ordered in the following manner, for example, case #5 shown in figure 4.5. Each case has at least 4 directories with this composition, for: LEFT_CC, LEFT_MLO, RIGHT_CC, RIGHT_MLO.

To start we need to move have all mammograms of the same label in one folder and have them renamed to match the names in the CSV file that contains the metadata.

"New name = patient id + image side + image type +. dcm"

This is done for both mass and calcification cases in the training and test sets without changing the cases in each set.

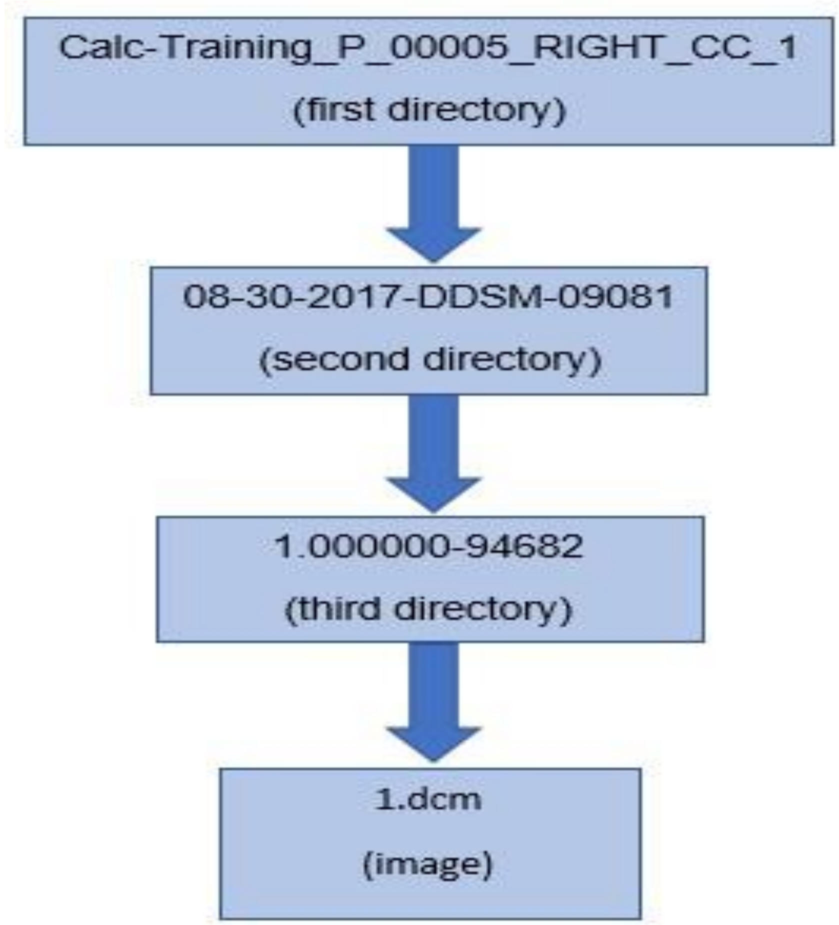

Figure 4.5 Dataset initial directory form.

❖ Convert to JPG Format

Python cannot manipulate DICOM files directly, we found that converting them to a “manip-ulatable” format is a must. That’s why a python code is used here to convert all files in the folders we organized previously, with the aid of the following packages: pydicom, os, and OpenCV.

❖ Resize the Images

The next step is to resize the images to be of the same size as well as be suitable for the networks we are using. The images were resized to be (224*224), where the width of the image = the height of the image = 224. An example is shown in Figures 4.6 and 4.7.

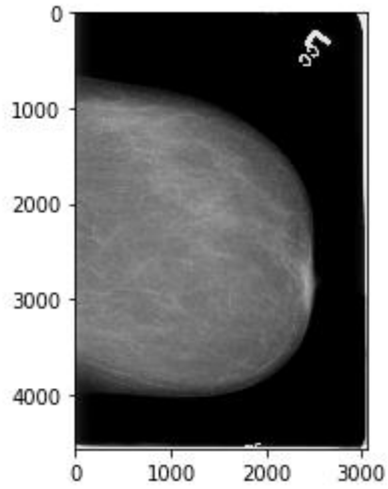

Figure 4.6 Original image.

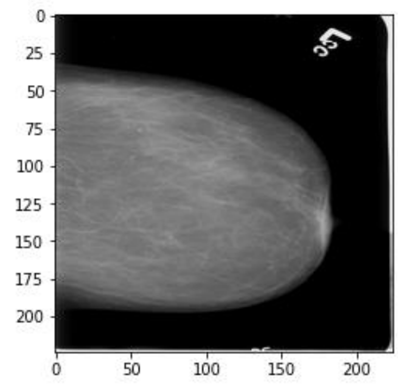

Figure 4.7 Resized image.

❖ Produce 3-Channel Images

Almost all available state-of-the-art CNNs are pretrained on ImageNet, which is composed of colored images (RGB). This means that the input layer of these networks takes an image of shape (w, h,3), where w and h are the width and height of the input image, respectively, and 3 is the number of channels. This being the case, the network can't accept 1-channel images as inputs.
Mammograms are grayscale images (1-channel), which makes feeding them directly to the network a wrong action. There are multiple solutions for this problem, but the most suitable and effective solution found was to "stack" the resized image to produce a 3-layered or 3-channel image of the same height and width.

❖ Labeling

The metadata (.csv) file is used in order to put a label on each image whether it is full mammogram or ROI images. This is done through the connection between the names of the cases found in the metadata and the images' names. In this step, we connect between each image of the same case and its corresponding information in the CSV file. Figure 4.8 shows the directories created for each class in the dataset.

Figure 4.8 Dataset classes after labeling.

❖ Data Splitting

We mentioned earlier that the dataset was already split into training and test sets according to the BI-RADS assessment, where the test set contains 20% of the total number of images in the dataset.

To prepare the data for training the model, the already existing training set is split into two parts; one part for training the neural network, while the other is for validating the performance of that network during the training stage. As shown in Figure 4.9

The validation set was created by randomly taking 10% of the images from the available training set before augmentation.

Figure 4.9 Directories of split data.

❖ Data Augmentation

This is a strategy used to increase the amount of data by using techniques like cropping, padding, flipping, etc.

This technique makes the model more robust to slight variations, and hence prevents the model from overfitting.

Since it's neither practical nor efficient to store the augmented data in the device's memory, we used the ImageDataGenerator class included in tensorflow and keras. ImageDataGenerator class generates batches of tensor image data with real-time data augmentation. This means it can generate the augmented data within itself and pass it directly to the network without the need to store in memory. The output images it generates will have the same dimensions as the original images in the dataset.

We can see an image in figure 4.10 below, this is a sample image that we'll use to demonstrate the augmentation methods' effect.

Figure 4.10 Original image [53].

The methods that we used from this class are:

❖ Rescaling “rescale”

This is done to scale the pixel values in the images between 0 and 1. If None or 0, no rescaling is applied, otherwise the data is multiplied by the value provided (after applying all other transformations).

We used a resale value = 1. /255

❖ Rotation “rotation_range”

Specifying this value generates images rotated randomly by an angle in the range of -(rotation_range) to +(rotation_range). As shown in figure 4.11.

Figure 4.11 Resulting images from rotating the original image [53].

❖ Shifting

- Width shifting → the width_shift_range is a floating-point number between 0.0 and 1.0 which specifies the upper bound of the fraction of the total width by which the image is to be randomly shifted, either towards the left or right. Figure 4.12.

Figure 4.12 Resulting images after shifting the original image horizontally [53].

- Height shifting → Exactly like width shifting, except that the image is shifted vertically instead of horizontally. Figure 4.13.

Figure 4.13 Resulting images after shifting the original image vertically [53].

❖ Shear intensity “shear_range”

Shear transformation slants the shape of the image. This is different from rotation in the sense that in shear transformation, we fix one axis and stretch the image at a certain angle known as the shear angle. This creates a sort of ‘stretch’ in the image, which is not seen in rotation. shear_range specifies the angle of the slant in degrees. Figure 4.14

Shear angle is counter clock-wise in degress. we used shear_range=0.2

Figure 4.14 Applying shear transformation to the image [53].

❖ Zoom “zoom_range”

A random zoom is obtained by the zoom_range argument. A zoom less than 1.0 magnifies the image, while a zoom greater than 1.0 zooms out of the image. Figure 4.15.

We used zoom_range = [0.8, 1.2]

Figure 4.15 Resulting images after zooming in and out of the original image [53].

❖ Flipping

Flipping takes a Boolean value, either: True or False.

- Horizontal_flip → The generator will generate images, which on a random basis, will be horizontally flipped. Shown in figure 4.16.

Figure 4.16 The image after horizontal flipping [53].

- Vertical_flip → vertically flipped images are generated. Shown in figure 4.17.

Figure 4.17 Resulting images after vertical flipping [53].

❖ Fill Mode: Nearest

When augmenting data, there certainly will be some points for which we do not have any values (Figure 4.18). Several options are available to solve this problem, and “Nearest” is one of them.

Nearest is the default option where the closest pixel value is chosen and repeated for all empty values. Figure 4.19.

Figure 4.18 Missing values filled with the nearest point value [53].

Figure 4.19 Resulting images after using fill_mode=Nearest [53].

❖ Channel Shift “channel_shift_range”

Randomly shifts the channel values by a random value chosen from the range specified by channel_shift_range as shown in figure 4.20.

channel_shift_range = 20

Figure 4.20 Resulting images from channel shifting [53].

❖ Zero centering the data

The method used is featurewise_center = True

❖ Preprocessing_function

It’s an argument that can be used to perform custom processing of the image.

4.5 Building the Model

Differentiation between benign and malignant breast cancer cases in mammograms can be difficult due to their similar features. For this, the transfer learning technique has been used to classify benign and malignant breast cancer by fine-tuning various pre-trained networks such as AlexNet, visual geometry group (VGG), MobileNet, and residual network (ResNet) on breast cancer datasets. However, these pre-trained networks have been trained on large benchmark datasets such as ImageNet, which do not contain labeled images related to breast cancers which lead to poor performance [47].

To overcome this obstacle, we followed some procedures that were found to increase the efficiency of the prediction models.

- **Load and Fine-tune the Model**

We followed a certain procedure where we first build a patch classifier model using the lesion annotations an ROI cropped images of the CBIS-DDSM. Then that model is converted into a whole image classifier by adding a few fully connected layers and an output layer of 2 classes [47].

This makes the model familiar with the mammograms first (in addition to the parameters gained from pre-training on ImageNet), which enables the model to extract the necessary features to detect objects on mammograms and classify them. By further training the model on the whole images, a whole image classifier is formed, which can then be further fine-tuned and transferred to other datasets and make predictions on them.

1. The Patch Classifier

The first step now is to load one of the pre-trained networks as a base model. The networks used are ResNet50 and MobileNet.

The network is loaded from the disk (as shown in Figure 4.21), along with its ImageNet weights, without including the top fully-connected layer of the model that produces the output of 1000 classes of ImageNet. This layer is removed for us to be able to fine-tune the model.

```
# Convoluted Base MODEL

conv_base = ResNet50(weights='imagenet',include_top=False, input_shape=(224,224 , 3))

print(conv_base.summary())
```

Figure 4.21 Loading the base model.

The model is fine-tuned by adding a number of layers including fully-connected layers (dense layers) with ReLU activation function. Then, a top layer with a Softmax activation function. The top layer has a number of neurons equal to the output number of classes we desire.

The base model and the added layers are shown in Figure 4.22.

```
model = models.Sequential()
model.add(conv_base)
model.add(layers.Flatten())
model.add(layers.Dropout(0.5))
model.add(layers.Dense(1024, activation='relu',kernel_regularizer=regularizers.l2(0.001)))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(4, activation='softmax'))

print(model.summary())
```

Figure 4.22 Fine-tuned model.

The added layers (Figure 4.23)

- Flatten layer → It collapses the spatial dimensions of the input into the channel dimension.
- Dropout → It randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting. Inputs not set to 0 are scaled up by 1/ (1 - rate) such that the sum over all inputs is unchanged.
- Fully-connected layer → It operates on a flattened input where each input is connected to all neurons. If present, FC layers are usually found towards the end of CNN architectures and can be used to optimize objectives such as class scores.
- Softmax → Classifier.

```
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 7, 7, 2048)        23587712
_________________________________________________________________
flatten_1 (Flatten)          (None, 100352)            0
_________________________________________________________________
dropout_2 (Dropout)          (None, 100352)            0
_________________________________________________________________
dense_2 (Dense)              (None, 1024)              102761472
_________________________________________________________________
dropout_3 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_3 (Dense)              (None, 4)                 4100
=================================================================
Total params: 126,353,284
Trainable params: 126,300,164
Non-trainable params: 53,120
_________________________________________________________________
None
```

Figure 4.23 Fine-tuned model summary.

- **Compile the Model**

The model is compiled with learning rate decay and the Adam optimizer for its advantages:

- It's is too fast and converges rapidly.
- Rectifies vanishing learning rate, high variance.

```
In [24]: from tensorflow.keras.optimizers import Adam

         model.compile(optimizer=Adam(lr=.0001), loss='categorical_crossentropy', metrics=['accuracy'])
```

Figure 4.24 Compiling the model.

- **Model Training [43]**

After comparing the networks with pre-trained weights using the ImageNet database to those with randomly initialized weights, In a pre-trained network, the bottom layers represent primitive feature s that tend to be preserved across different tasks, whereas the top layers represent higher order features that are more related to specific tasks and require further training. Using the same learning rate for all layers may destroy the features that were learned in the bottom layers. To prevent this, a 3-stage training strategy was employed in which the parameter learning is frozen for all but the final layer and progressively unfrozen from the top to the bottom layers, while simultaneously decreasing the learning rate. The 3-stage training strategy on the patches as follows:

a. Set learning rate to 10^{-3} and train the last layer for 3 epochs.
b. Set learning rate to 10^{-4} ,unfreeze the top layers and train the last layer for 10 epochs, where the top layer number is set to 46 for Resnet50.
c. Set learning rate to 10^{-5}, unfreeze all layers and train for 37 epochs for a total of 50 epochs.

the images from the ImageDataGenerator, and pass them to the model through the fit function. The model is now being trained on the images from the training set, then evaluated at the same time by the separate validation set we made previously.

```
In [*]: history = model.fit(train_batches, steps_per_epoch = 1864 // 64,
            validation_data=validation_batches, validation_steps = 661 // 64, epochs=100, callbacks=callbacks_list, verbose=1)
```

Figure 4.25 Fitting the model.

2. The Image Classifier

Training a whole image classifier was achieved in two steps. The first step was to train a patch classifier. The second step was to train a whole image classifier converted from the patch classifier. A 2-stage training strategy was employed to first train the newly added top layers and then train all layers with a reduced learning rate, which was as follows:

1. Set learning rate to 10^{-4}, weight decay to 0.001 and train the newly added top layers for 30 epochs.
2. Set learning rate to 10^{-5}, weight decay to 0.01 and train all layers 20 epochs for a total of 50 epochs.

Web application

- **Deployment of mobile net model using flask:**

Trying to make a simple way to classify the images of mammography by creating a flask framework to launch a web application that able to be interacted with by a user.

- Create app.py , templates and upload folder :

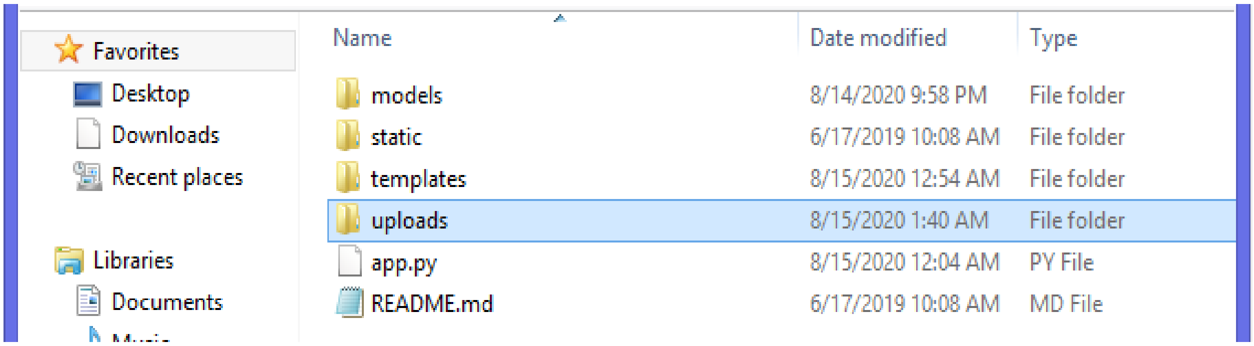

Figure 4.26 Created app.py, templates and upload folder.

- Create html files:

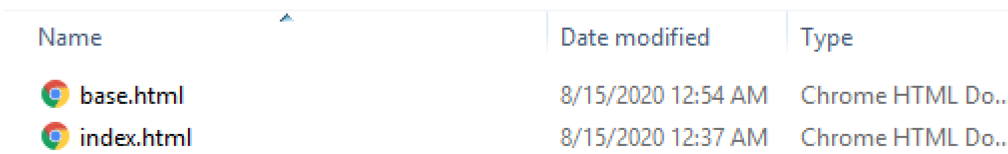

Figure 4.27 Created html files.

- Use flask modules from :

```
from flask import Flask, redirect, url_for, request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer
```

➢ Set the layout of the window of the web app:

```
<html lang="en">

<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <meta http-equiv="X-UA-Compatible" content="ie=edge">
   <title>Breast cancer detection project</title>
   <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
   <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
   <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
   <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
   <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
</head>
<body>
   <nav class="navbar navbar-dark bg-dark">
      <div class="container">
         <a class="navbar-brand" href="#">Biomedical department helwan university</a>
         <button class="btn btn-outline-secondary my-2 my-sm-0" type="submit">Help</button>
      </div>
   </nav>
   <div class="container">
      <div id="content" style="margin-top:2em">{% block content %}{% endblock %}</div>
   </div>
</body>

<footer>
   <script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
</footer>
```

- Image upload:

```html
<h2>Breast cancer detection</h2>

<div>
   <form id="upload-file" method="post" enctype="multipart/form-data">
      <label for="imageUpload" class="upload-label">
         Choose...
      </label>
      <input type="file" name="file" id="imageUpload" accept=".png, .jpg, .jpeg">
   </form>

   <div class="image-section" style="display:none;">
      <div class="img-preview">
         <div id="imagePreview">
         </div>
      </div>
      <div>
         <button type="button" class="btn btn-primary btn-lg " id="btn-predict">Predict!</button>
      </div>
   </div>

   <div class="loader" style="display:none;"></div>

   <h3 id="result">
      <span> </span>
   </h3>

</div>
```

- Launch our application:

```python
@app.route('/', methods=['GET'])
def index():
   # Main page
   return render_template('index.html')
```

- Prediction:

```
@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':
        # Get the file from post request
        f = request.files['file']
```

- Save the file to ./uploads

```
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)

        # Make prediction
        preds = model_predict(file_path, model)

        # Process your result for human
        # pred_class = preds.argmax(axis=-1)            # Simple argmax
        pred_class = decode_predictions(preds, top=1)   # ImageNet Decode
        result = str(pred_class[0][0][1])               # Convert to string
        return result
    return None

if __name__ == '__main__':
    app.run(debug=True
```

CHAPTER 5

RESULTS

In this chapter we will discuss the procedures that had been followed to show results from predictions that configure its performance according to many different parameters.

Procedure1:

The following procedure is to compare between the performance of ResNet50 and MobileNet shown in (Table 5.1).

Hardware environment:

Laptop1 is Core i7 ,16 GB RAM and an NIVIDIA GeForce GTX 960 M GPU.
Pc1 is pc marim

For resnet50 :

Using 1896 training & 661 Validation CBIS-DDSM preprocessed dataset with built-in augmentation, 224*224 "PNG" images, Freeze the weights of all layers of pre-trained resnet50 and trained only the head of networks as shown in figure 5.1, 100 epochs and batch size =64.

```
model = models.Sequential()
model.add(conv_base)
model.add(layers.Flatten())
model.add(layers.Dropout(0.5))
model.add(layers.Dense(1024, activation='relu',kernel_regularizer=regularizers.l2(0.001)))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(4, activation='softmax'))

print(model.summary())
```

Figure 5.1 The head of network.

for MobileNet:

Using 18885 training & 4719 validation CBIS-DDSM preprocessed dataset with built out augmentation, different size of "PNG" images, freeze the weights of all layers except the last 23 layers in the new model. Epochs=30 and batch size =64.

5.1 Evaluation1

Table 5.1 Compare between (ResNet50 and MobileNet).

Parameters	ResNet50	Mobile Net
Nb_epochs	100	50
Batch_size	64	64
Augmentation_parameters	Shear range=0.2 Zoom range= (0.8,1.2) Horizontal flip Vertical flip Rescale=1. /255	Shear range=0.2 Zoom range= (0.8,1.2) Horizontal flip Vertical flip Rescale=1. /255
Augmentation positioning	IN	out
Learning rate (lr.)	0.001	0.001
Validation_set	0.2	0.2
Resize	with	without
Test_acc.	60%	62%

Procedure 2

The following procedure is to use ResNet50 with 2 stages shown in (Table 5.2).

<u>Hardware environment:</u>

Laptop2 is Core i7-5500U CPU ,8 GB RAM
Pc1 is pc marim

<u>Procedure inputs:</u>

Using 1896 training & 661 Validation CBIS-DDSM preprocessed dataset with built-in augmentation, 224*224 “PNG” images.

ResNet50:

ROI:

Training a patch classifier, a 3-stage training strategy was employed in which the parameter learning is frozen for all but the final layer and progressively unfrozen from the top to the bottom layers, while simultaneously decreasing the learning rate.

Whole image:

Training a whole image classifier, A 2-stage training strategy was employed to first train the newly added top layers and then train all layers with a reduced learning rate.

5.2 Evaluation 2

Table 5.2 The result of procedure 2.

Parameters	**ROI**	**Whole**
Nb_epochs	50	50
Batch_size	64	64
Augmentation_parameters	Shear range=0.2 Zoom range=(0.8,1.2) Horizontal flip Vertical flip Rescale=1. /255	Shear range=0.2 Zoom range= (0.8,1.2) Horizontal flip Vertical flip Rescale=1. /255
Augmentation positioning	Built-in	Built-in
Learning rate (lr.)	0.0001	0.0001
Weight decay	0.001	0.001
Validation_set	0.2	0.2
resize	224,224	224,224
Val_ acc.	79	
Test_acc.	-	

Web Application

Procedure 3

In the following procedure, a web application is created by flask framework to classify mammography images.

It returns: 127.0.0.1.5000

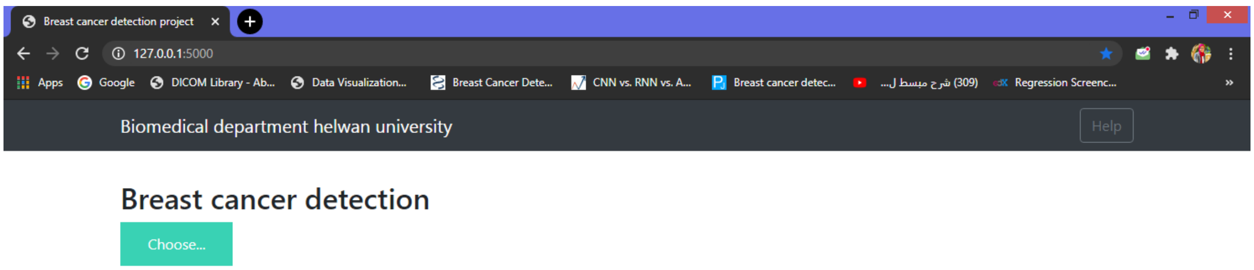

Figure 5.2 Start web app.

➢ Show the image:

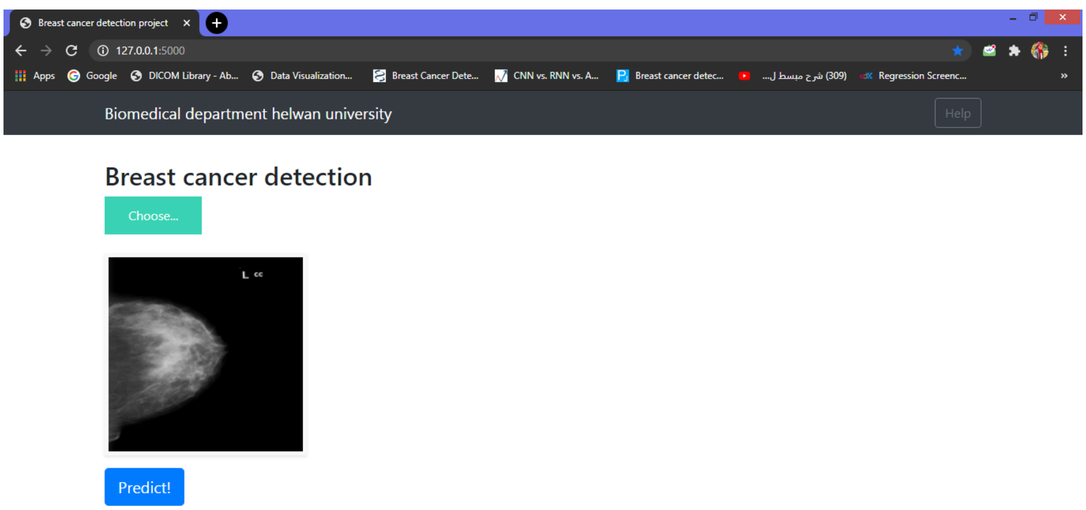

Figure 5.3 Show the image in the web app.

➢ Saving the uploaded image in uploads:

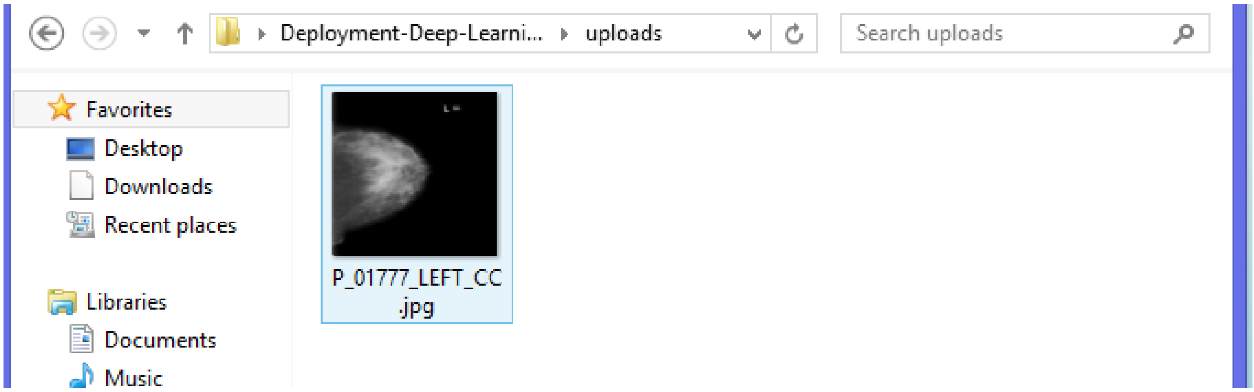

Figure 5.4 Load an image for prediction.

5.3 Limitations:

- The accuracy is noticed in our project is lower than expected because the specs of our laptop doesn't match their laptop that All experiments in their study were carried out on a Linux workstation equipped with an NVIDIA 8 GB Quadro M4000 GPU card and our laptop is Core i7 ,16 GB RAM and an NIVIDIA GeForce GTX 960 M GPU graphic card with 4 GB of RAM.
- Rarity of public Mammogram dataset.
- Data has so many artifacts.
- Unorganized Data folders.

CHAPTER 6

CONCLUSION

Breast cancer is disease that creates from breast tissue. Indications of breast cancer may include many signs ex. a lump in the breast, a change in breast shape, dimpling of the skin, fluid coming from the nipple, a newly-inverted nipple, or a red or scaly patch of skin. There may be bone pain, swollen lymph nodes, shortness of breath, or yellow skin, for people with distant spread of the disease only. There are risk factors for breast cancer include being female, a lack of physical exercise, alcoholism, hormone replacement, obesity, therapy during menopause, ionizing radiation, an early age at first menstruation. Sometimes it can be difficult for doctors to find breast cancer by looking to mammograms.

In this project, digitized film mammograms of CBIS-DDSM dataset are used for 2620 patient, breast cancer is detected by method of (end-to-end) that consist of many steps. First, preprocessing of data set that contained (rename and moving images, file format conversion, splitting and data augmentation before the training). second, frameworks are used to build our model. In addition to the architectures of our model and we trained data set and make fine tuning parameters to improve the accuracy. Finally, the model is evaluated to be able to predict the results of new image.

There is a method to be sure if suspicious area is cancer or not called biopsy. Doctors will remove a small tissue from breast to detect it. There are two main kinds of breast biopsies. One is called surgical biopsy and other is called core-needle biopsy. Doctors recommend the kind of breast biopsy depend on size, shape, and location of suspicious area. Then, doctor take tissue after biopsy under microscope to check changes and know if that cancer or not. Side effects of biopsy are explored that is Bleeding, bruising, and infection. Some of women will suffer from infection after biopsy. And we explored few side effects with method of core-needle.

In our project, we found that the computational environment will influence in the accuracy of classification for whole image. We can improve the accuracy by sampling more or larger patches to include neighboring regions around the ROI and additional background regions, Where the relation between the computational burden and increasing number or size of patches is linear.

Obstacles

Many problems were faced during working in this project. First, was to find dataset with high resolution images to build and train our model. Second, was to have laptop with high specifications to fast the training of dataset and to have model with high accuracy.

Experience gained

From this project we have gained a lot of experience specially in software field. We get knowledge about many courses such as (python, NN, CNN, Keras). We learned how to build a model and train this model to classify new images.

Moreover, building some applications gave us good experience and knowledge in this field and allowed us to deal with some web applications and programs.

Future developments

Developing a project and having clear steps for the future to come ahead is the true meaning of evolution and success. That's why we came up with some future plans to improve this project and make it more beneficial to the society.

These steps are:

- Implement the networks and train them on more advanced computers with higher hardware specifications.
- Try other architectures like VGG and Inception, as well as developing the currently created models in this project until the best possible performance is reached.
- Contact Bahia hospital to get data of real cases in Egypt.
- Develop the web application for better performance and user experience.
- Launch the web application on a server and make it available to the public.

REFERENCES

1. "What Is Breast Cancer?", *Centers for Disease Control and Prevention*.
2. "5 Signs of Breast Cancer (Aside From a Lump) - WomenWorking", *WomenWorking*.
3. O'Harte, F., Parthsarathy, V., Hogg, C., & Flatt, P. (2018). Long-term treatment with acylated analogues of apelin-13 amide ameliorates diabetes and improves lipid profile of high-fat fed mice. *PLOS ONE*, *13*(8), e0202350. doi: 10.1371/journal.pone.0202350.
4. "Breast Cancer". NCI. January 1980. Archived from the original on 25 June 2014. Retrieved 29 June 2014.
5. "Breast Cancer Treatment (PDQ®)". NCI. 23 May 2014. Archived from the original on 5 July 2014. Retrieved 29 June 2014.
6. Saunders C, Jassal S (2009). Breast cancer (1. ed.). Oxford: Oxford University Press. p. Chapter 13. ISBN 978-0-19-955869-8. Archived from the original on 25 October 2015.
7. World Cancer Report 2014. World Health Organization. 2014. pp. Chapter 5.2. ISBN 978-92-832-0429-9.
8. R. Campanini et al., "A novel featureless approach to mass detection in digital mammograms based on support vector machines," Phys. Med. Biol., vol. 49, no. 6, p. 961, 2004.
9. X. Si and L. Jing, "Mass detection in digital mammograms using twin support vector machine-based cad system," in Information Engineering, 2009. ICIE'09. WASE International Conference on, 2009, vol. 1, pp. 240–243.
10. F. Eddaoudi, F. Regragui, A. Mahmoudi, and N. Lamouri, "Masses detection using SVM classifier based on textures analysis," Appl. Math. Sci., vol. 5, no. 8, pp. 367–379, 2011.
11. C.-C. Jen and S.-S. Yu, "Automatic detection of abnormal mammograms in mammographic images," Expert Syst. Appl., vol. 42, no. 6, pp. 3048–3055, 2015.
12. M. G. Ertosun and D. L. Rubin, "Probabilistic visual search for masses within mammography images using deep learning," in 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2015, pp. 1310–1315.
13. M. M. Jadoon, Q. Zhang, I. U. Haq, S. Butt, and A. Jadoon, "Three-class mammogram classification based on descriptive CNN features," Biomed Res. Int., vol. 2017, 2017.
14. Lévy D, Jain A. Cornell University. 2016. Breast Mass Classification from Mammograms Using Deep Convolutional Neural Networks.

15. Samala RK, Chan HP, Hadjiiski L, Helvie MA, Richter CD, Cha KH. Breast cancer diagnosis in digital breast tomosynthesis: effects of training sample size on multi-stage transfer learning using deep neural nets. IEEE Trans Med Imaging. 2019 Mar;38(3):686–96. doi: 10.1109/TMI.2018.2870343.
16. u E, Wu K, Cox D, Lotter W. Conditional Infilling GANs for Data Augmentation in Mammogram Classification. Proceedings of the Image Analysis for Moving Organ, Breast, and Thoracic Images; RAMBO'18; September 16-20, 2018; Granada, Spain. 2018. pp. 98–106.
17. Javier Andreu Perez, Fani Deligianni, Daniele Ravi and Guang-Zhong Yang Artificial Intelligence and Robotics.
18. Intelligent Systems Reference Library, Volume 17.
19. Michael Nielsen Neural Networks and Deep Learning.
20. Yoshua Bengio, Aaron Courville, and Pascal Vincent Representation Learning: A Review and New Perspectives.
21. Yann LeCun1,2, Yoshua Bengio3 & Geoffrey Hinton4,5 Deep learning doi:10.1038/nature14539.
22. Giang Nguyen · Stefan Dlugolinski Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey.
23. Neural Networks and Introduction to Deep Learning Toulouse Mathematics Institute.
24. Ulugbek Abdullaev, Bedilbek Khamidov Convolutional Neural Networks for Image Classification.
25. Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, Humaira Arshad. State-of-theart in artificial neural network applications: A survey. Heliyon 4 (2018) e00938.
26. Deep learning An MIT book deeplearningbook.org.
27. "Understanding of Convolutional Neural Network (CNN) — Deep Learning", *Medium*.
28. "7 Types of Activation Functions in Neural Networks: How to Choose?", *MissingLink.ai*.
29. Chigozie Enyinna Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen MarshallActivation Functions: Comparison of Trends in Practice and Research for Deep Learning.
30. "Various Optimization Algorithms For Training Neural Network", *Medium*.
31. "CS 230 - Convolutional Neural Networks Cheatsheet", *Stanford.edu*.
32. "Flatten layer - MATLAB", *Mathworks.com*.
33. Javier Andreu Perez, Fani Deligianni, Daniele Ravi and Guang-Zhong Yang Artificial Intelligence and Robotics.

34. Intelligent Systems Reference Library, Volume 17.
35. Michael Nielsen Neural Networks and Deep Learning.
36. Yoshua Bengio, Aaron Courville, and Pascal Vincent Representation Learning: A Review and New Perspectives.
37. Yann LeCun1,2, Yoshua Bengio3 & Geoffrey Hinton4,5 Deep learning doi:10.1038/nature14539.
38. Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., López García, Á., & Heredia, I. et al. (2019). Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey. *Artificial Intelligence Review*, *52*(1), 77-124. doi: 10.1007/s10462-018-09679-z.
39. Neural Networks and Introduction to Deep Learning Toulouse Mathematics Institute.
40. Ulugbek Abdullaev, Bedilbek Khamidov Convolutional Neural Networks for Image Classification.
41. Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, Humaira Arshad. State-of-theart in artificial neural network applications: A survey. Heliyon 4 (2018) e00938.
42. Deep learning An MIT book deep learning book.
43. https://medium.com/x8-the-ai-community/cnn-9c5e63703c3f.
44. "Gradient Descent", *Medium*.
45. A. Rosebrock, "Keras ImageDataGenerator and Data Augmentation - PyImageSearch", *PyImageSearch*.
46. J. Brownlee, "Image Augmentation for Deep Learning With Keras", *Machine Learning Mastery*.
47. "Breast cancer: Anatomy and early warning signs", *Medicalnewstoday.com*
48. "5 Signs of Breast Cancer (Aside From a Lump) - WomenWorking", *WomenWorking*.
49. Cancer, B., Cancer, C., Cancer, L., Cancer, P., Types, V., & Program, C. et al. (2020). What are the Symptoms and Signs of Breast Cancer?.
50. Breast Cancer Signs and Symptoms | Most Common Symptoms.
51. deeplearning.ai
52. https://www.towardsdatascience.com/understanding-and-coding-a-resnet-in-keras
53. https://link.medium.com/hUNhM8B3M8

54. Mohammad Alkhaleefah, Shang-Chih Ma, Yang-Lang Chang, Bormin Huang, Praveen Kumar Chittem, and Vishnu Priya Acchannagari “Double-Shot Transfer Learning for Breast Cancer Classification from X-ray Images”.
55. Ridhi Arora, Prateek Kumar Rai, Balasubramanian Raman, “Deep feature–based automatic classification of mammograms”.
56. Rebecca Sawyer Lee, Francisco Gimenez, Assaf Hoogi, Kanae Kawai Miyake, Mia Gorovoy & Daniel L. Rubin, “Data Descriptor: A curated mammography data set for use in computer-aided detection and diagnosis research”.