Optimizing Convolutional Neural Network Parameters using Genetic Algorithm for Breast Cancer Classification

by

Khatereh Davoudi

A thesis submitted to
The Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements
of the degree of

Master of Science

Department of Computer Science
The University of Manitoba
Winnipeg, Manitoba, Canada
May 2020

Thesis advisor

Dr. Parimala Thulasiraman

Author

Khatereh Davoudi

Optimizing Convolutional Neural Network Parameters using Genetic Algorithm for Breast Cancer Classification

Abstract

Breast cancer, as the most-regularly diagnosed cancer in women, can be controlled effectively by early-stage tumour diagnosis. Clinical specialists use Computer-Aided Diagnosis (CAD) systems to help aid in their diagnosis, as accurate as possible. Deep learning techniques, such as Convolutional Neural Network (CNN), due to their classification capabilities, have been widely adopted in CAD systems. The parameters of the network, including the weights of the convolution filters, and the weights of the fully connected layers play a crucial role in classification accuracy. Back-propagation technique is the most frequently used approach for training CNN. However, this technique has some disadvantages, such as getting stuck in local minima. In this thesis, we propose to optimize the weights of the CNN using Genetic Algorithm (GA). The work consists of: designing a CNN model to facilitate the classification process, training the model using three different optimizer (mini-batch gradient descent, Adam, and GA), and evaluating the model through various experiments on BreakHis dataset. We show that the CNN model trained through GA performs as well as the Adam optimizer with a classification accuracy of 85%.

Contents

List of Figures

List of Tables

Acknowledgments

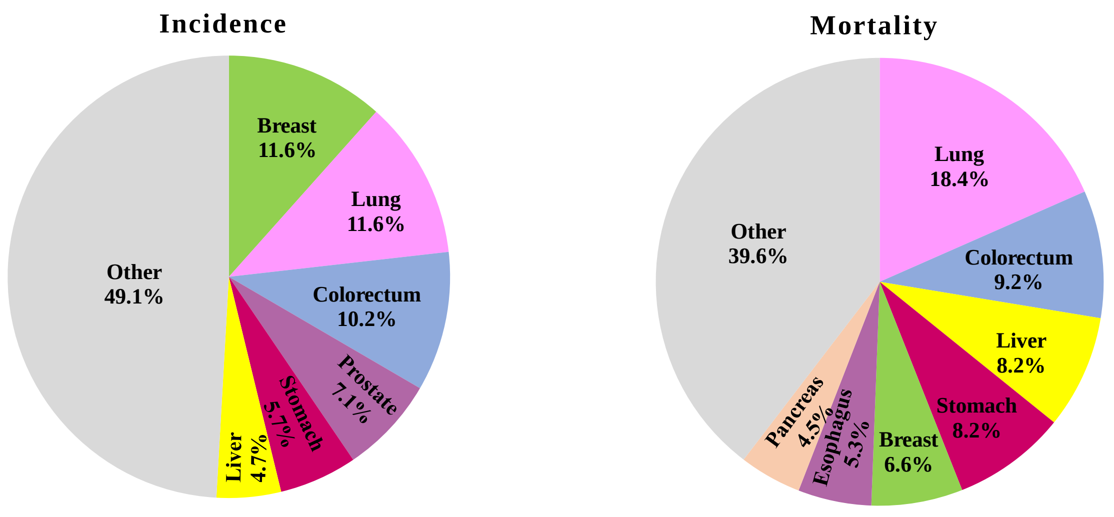
I would like to express my special thanks and appreciation to my thesis advisor, Dr. Parimala Thulasiraman, who guided and encouraged me during my master studies. The goal of this project would not have been realized without her continuous technical, financial and mental support. I wish to extend my sincere gratitude to my committee members Dr. Sherif Sherif and Dr. Shahin Kamali for generously offering their time, support, and insightful feedback. I would also like to thank all my friends and lab mates for their encouragement and support during this wonderful experience. My special thanks goes to my family, especially my loving parents Naser and Behnaz, for their unconditional love and support. Words cannot express how grateful I am to my family for the countless sacrifices that they have made on my behalf. Miles apart, they always have been with me by their heart. Also, my special thanks equally goes to my family-in-law, for their constant love, understanding, and support. Finally, my deepest thanks belongs to the most special person in my life, my best friend and my beloved husband, Esmaeil, who always inspired me to follow my dreams. He kept me going and this work would not have been possible without his endless support, care, understanding, patience, and love.

This thesis is dedicated to Esmaeil, because he always understood!

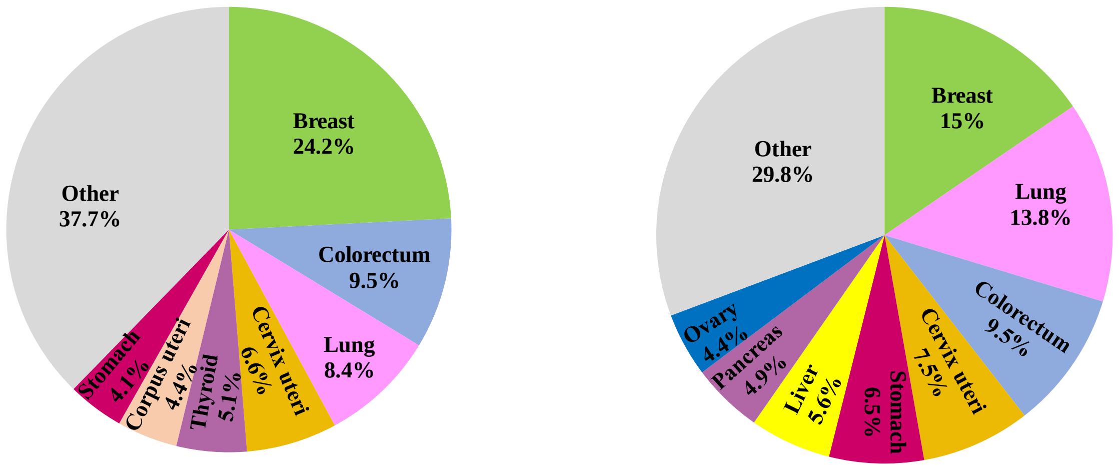
Chapter 1

Introduction

Breast cancer is the most regularly diagnosed cancer, and the main cause of cancer mortality in women around the world [3]. According to Global Cancer Statistics 2018 (GLOBOCAN 2018) , around 24.2% of the total cancer cases among females is breast cancer. Also, these statistics indicate that among all cancer types for both genders, breast cancer is the second most common type, following the lung cancer. Figure 1.1 shows [3] the case and mortality distribution of different cancer types in 2018.



(a) Both genders



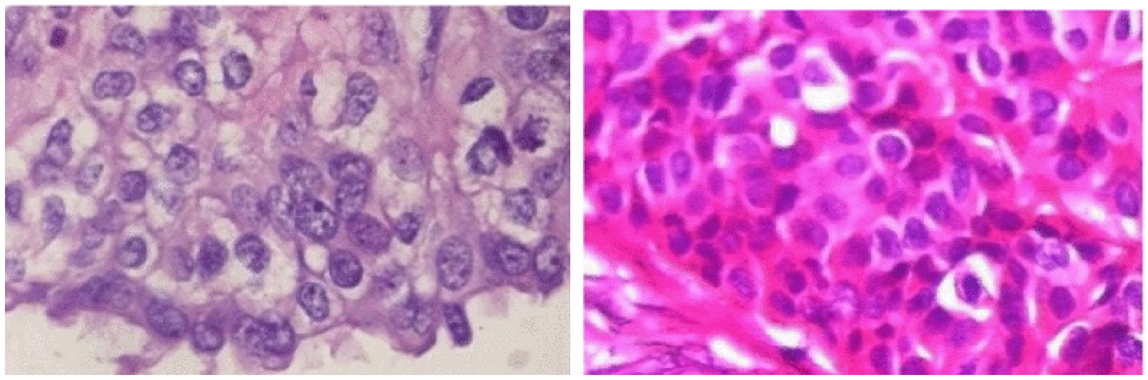
(b) Women

Figure 1.1: Pie charts show the worldwide cancer statistic (a) for both genders and (b) females in 2018.

However, breast cancer is one of the few cancers that can be controlled by early-stage detection, followed by effective clinical treatments. The necessity in early breast cancer diagnosis has led to significant development in breast screening modalities.

Nowadays, there are several non-invasive breast screening techniques such as mammography, magnetic resonance imaging (MRI), ultrasound, and microwave imaging (MI), which improve the breast cancer survival rate considerably by generating high-quality breast images. These images are analyzed in detail by specialists for an accurate diagnosis.

Despite significant development in recent non-invasive imaging technologies, invasive medical photography, which refers to the histologically assessment of breast tissue biopsy by a pathologist, is the gold standard for final breast cancer diagnosis in the clinical scenarios. Analysis of the histopathological images helps clinical experts to distinguish between Normal, Benign, and Malignant tissues [4]. Fine-needle aspiration, core-needle biopsy, vacuum-assisted, and surgical biopsy are the most regularly used breast biopsy methods in laboratories [5]. The cancer diagnosis procedure using histology images includes: collecting samples of breast tissues, processing the materials by a chemical substances called formalin, embedding in a chemical substances called paraffin, cutting the sample precisely and fixing across a glass microscope slide, staining with a stain combination called Hematoxylin and Eosin (HE) for nuclei and cytoplasm visibility, and finally visualizing the inspection of histopathological slides under the microscope by a pathologist [6; 7]. Moreover, by using a whole slide digital scanner, the breast histology slides can be digitized and saved as digital images. Figure 1.2 represents an example of the breast biopsy slides dyed with HE.



(a) Benign tumour (b) Malignant tumour

Figure 1.2: Slides stained with HE, (a) Benign tumour, (b) Malignant tumour

Pathologists search for specific features such as density, variability, architecture, spatial arrangement, and nuclei characteristics of stained cells under the microscope that allow them to distinguish among Benign and Malignant cases. For example, a Malignant sample shows a higher nuclei density and variability as well as distortion of the architecture rather than a Benign case [4].

However, due to the inherent complexity of histopathological images, and the large volume of generated slides in clinics, pathologists are unable to analyze these images easily and classify them using previous image classes existing in their database. Analysis of these images is a complicated, difficult, and highly time-consuming task, affected by different factors such as level of knowledge, experience, attention, and fatigue of the specialists [4; 5; 6; 8]. The current manual examination accuracy achieved by pathologists is approximately 75% [9]. Consequently, to alleviate the shortcomings of human interpretation and decrease the workload of the specialists, pathologists need the help of a computer to make their final decisions as accurately and reliably as possible [6; 10]. To facilitate this diagnosis process and increase the survival chances, Computer-Aided Diagnosis (CAD) systems have become essential and crucial in breast cancer classification problem.

CAD is a technology designed to reduce the false positive or negative diagnosis rate by taking the advantages of different elements such as medical image processing, artificial intelligence, and computer vision [11]. This technology consists of three main steps: image pre-processing, feature extraction, and classification as shown in Figure 1.3.

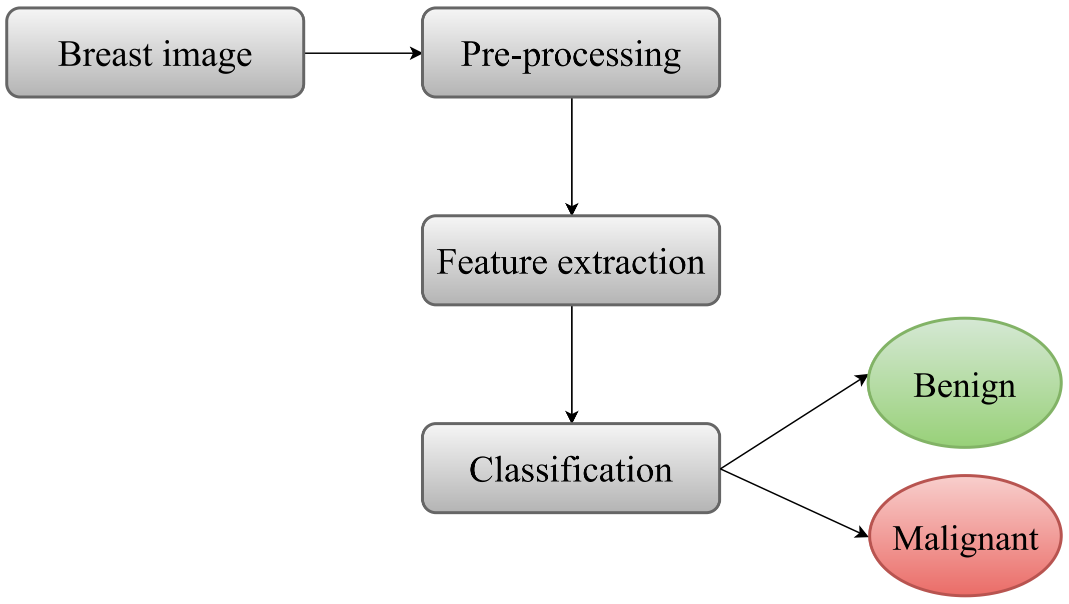


Figure 1.3: An overview of a simple CAD system for breast cancer classification task.

Among these steps, creating an accurate and reliable classification is the most critical component of CAD systems. Classification is a data mining process in which proper labels are assigned to the different existing groups by finding the hidden patterns from datasets. Then, the discovered patterns are used to classify other unknown cases in the future. Machine learning techniques, because of their classification capabilities, have been widely adopted in the classification step of CAD systems to help specialists detect abnormalities [12; 13; 14].

In general, it is essential to choose the correct features to describe an object or image. The image features extraction can be divided into two separate classes: hand-crafted and learned. K-Nearest Neighbors (K-NN) [15], Support Vector Machines (SVM) [16], Artificial Neural Networks (ANN) [17], and Decision Trees [18] that have been developed for breast cancer detection in CAD systems, use hand-crafted feature extraction methods. That is, based on the expert knowledge, features will be extracted from separate images according to a manually designed algorithms. This is one of the disadvantages of hand-crafted approaches. Furthermore, the hand-crafted feature extraction for breast histopathological image classification have other challenges. First, these images are fine-grained and high-resolution, representing the complex textures of the breast. The variability of images in the same class and inhomogeneity of colour distribution makes feature extraction a complex and difficult task producing just some low-level features. Second, the traditional feature extraction methods, such as scale-invariant feature transform [19] do not perform well on these complex images. Third, to extract the most important features, prior knowledge of data is required. These drawbacks cause hand-crafted feature extraction for breast histopathological images, a time-consuming and low-efficiency process, causing poor classification accuracy [8; 20].

To overcome the disadvantages of the feature-based strategies, deep learning, a subset of machine learning, is becoming a valuable alternative. This technique learns the features directly from the image dataset through a training procedure to perform the task at hand, in our case, the classification problem. In fact, deep learning does not require the design of feature extractors by a domain expert to classify images [5].

Convolutional Neural Networks (CNN), as a particular type of deep, feed-forward neural network, have been successfully used in different research areas such as image processing [21], speech recognition [22], signal processing [23], and object recognition [24]. Considering the recent advances in deep learning, CNN is widely used for extracting learned features from different types of breast images in order to solve the classification problem [5; 8; 10; 20; 25; 26]. Despite the recent development of CAD systems, breast cancer diagnosis remains a crucial health issue, and research is still on-going to improve the accuracy of these CAD systems.

This thesis focuses on CNN for the classification of breast cancer histology images. CNN is inspired by the biological neural network of the human brain. It works based on the idea that the system can learn from previous data. CNN includes several convolution layers, pooling layers, and fully-connected layers between the input and output layers, as shown in Figure 1.4. The precision of extracted features in CNN is highly dependent on the weights of the network, including the weights of all convolution filters, as well as the weights of connecting edges in the fully-connected layer. As a result, these weights play a crucial role in classification accuracy. During the training phase of the model, the weights are updated continuously to achieve a minimum classification error rate.

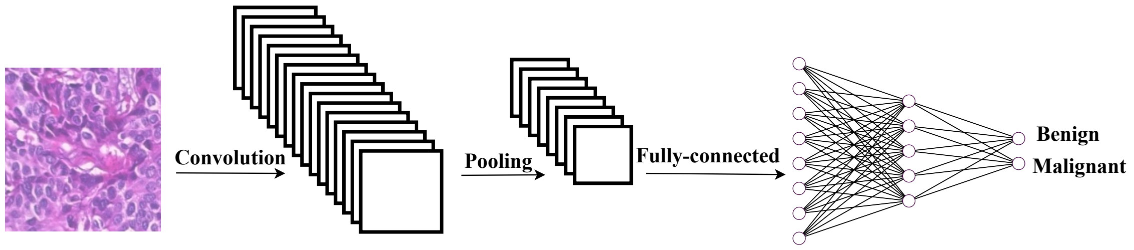


Figure 1.4: An overview of convolution neural network.

Back-propagation is the most frequently used technique for updating the weights in neural network [4; 26]. This algorithm uses an optimization technique called gradient descent to update the weights. The gradient of the model parameters is re-sampled, repetitively, in the backward direction of the network weights, to find a set of weights that minimizes the classification error value [27]. The disadvantage of the back-propagation algorithm, however, is that it requires high convergence time and may get stuck in local optima. Trapping in local optima means that during the process of finding the weights which minimize the classification error value, the back-propagation algorithm may find a weight with smaller error value at nearby points which is not necessarily the smallest one at all other feasible points [27; 28; 29]. To overcome the shortcomings of the back-propagation strategy, Genetic Algorithm (GA) [30], a well-known global optimization technique inspired by the process of natural selection, has been proposed for optimizing the weights of the neural network [31; 32; 33].

To the best of our knowledge, there is no scientific work using GA to optimize the parameters (weights) in CNN for histopathological breast image classification. Therefore, in this thesis, we use GA instead of back-propagation for solving the classification problem. We use the BreaKHis dataset containing breast cancer histopathological images for training and testing our proposed technique [5]. We study the effect of different components of GA and CNN, such as mutation and batch size on the classification accuracy.

We outline here our contributions in designing, optimizing and evaluating the CNN using GA:

- We create our dataset using the BreaKHis database containing breast cancer histopathological images [5], Binary classification implies classifying breast biopsy images into two main groups: Benign (non-cancerous) and Malignant (cancerous). Our dataset contains a total number of 7909 images, of which 70% images are used to train the model, and the rest 30% are used for the testing phase.

- We develop the CNN model, first through some pre-processing steps and then designing a proper network structure that facilitates the classification process. We seek for a network structure that prevents common problems such as over-fitting. Over-fitting refers to the situation in which the classifier does not generalize well between training data and test data. It means that there is a significant gap between training accuracy and test accuracy because the model did not learn the data and memorized it.

- To optimize our network we initially use gradient descent optimizer.

- We then optimize our model using a powerful Adam optimizer [34], a stochastic gradient descent optimization algorithm, that solves the the shortcoming of gradient descent.

- Subsequently, we use a GA optimizer to find the best set of the network weights aimed at achieving higher classification accuracy.
- Finally, we evaluate our designed models based on multiple criterion such as classification accuracy, execution time, recall, precision, and F1-score.

The rest of this thesis is organized as follows. Section 2 provides an overview of relevant literature. Section 3 describes the background for the thesis and methodology. Dataset and evaluation methods are addressed in Section 4. Experimental results are discussed in section 5. Discussion, conclusion and future work are presented in section 6.

Chapter 2

Literature Review

Computer aided diagnosis systems have gained popularity with ever increasing modern high performance computers and newer breast imaging modalities. Together with their own analysis, medical specialists use computerized analysis as a second opinion to make accurate and fast decisions [35]. One of the most challenging steps in CAD systems is classification [11]. Our literature review shows that different works have been done so far to find reliable classification techniques to classify breast tumours into two main groups, called Benign and Malignant cases. In this chapter, the literature review on breast cancer classification with respect to machine learning and evolutionary techniques is discussed.

2.1 Machine Learning

Machine learning techniques offer high classification accuracy and effective diagnostic capabilities. These techniques are heavily used in CAD systems for breast cancer classification problem [36]. In order to effectively use machine learning, we need data. The data has to be analyzed and trained for the specific problem. For breast cancer classification, there are some standard image databases. These databases contain different types of breast images, such as mammography, ultrasound, MRI, and histopathological breast images. Table 2.1 lists a few examples of available breast image databases [37].

Table 2.1: Available breast image database for breast cancer classification

Dtabaset	Number of Images	Imaging Technique	Total patients
MIAS	322	Mammogram	161
DDSM	AX	Mammogram	2620
CBIS-DDSM	4067	MG	237
ISPY1	386528	MR, SEG	237
BREAST-DIAGNOSIS	105050	MRI, PET, CT	88
Mouse-Mammary	23487	MRI	32
Inbreast	419	Mammogram	115
BreakHis	7909	Histopathology	82

2.1.1 Unsupervised Learning

Unsupervised clustering algorithms, such as K-means technique [15], have been used for breast cancer classification. Unlike the supervised classification technique, unsupervised learning algorithm classifies the data without any prior knowledge about the target. K-NN clustering and Fuzzy C-means method are utilized by Cahoon et al. [38] for unsupervised classification of breast mammography images. Singh et al. [39] proposed a method for breast cancer mass identification and calcification in breast mammograms. They used a combination of K-means and Fuzzy C-means clustering to diagnose breast cancer. The experimental results illustrate that this proposed approach can facilitate the early breast cancer diagnosis process.

In [40], Lashkari et al. compared a supervised technique called Adaboost method to unsupervised Fuzzy C-means algorithm for breast thermal images classification. 23 features are extracted by well-known feature selection methods, such as minimum redundancy and maximum relevance. They conclude that Fuzzy C-means method, with 75% of classification accuracy, can be a suitable solution for unsupervised classification. Kowal et al. [41] multiple clustering algorithms such as Fuzzy C-means, K-means, competitive learning neural networks and Gaussian mixture models are tested. for classification of fine-needle biopsy microscopic breast images. In this study, the nuclei regions on the images are used in the classifiers. For segmentation, first adaptive threshold is applied to perform foreground and background segmentation. Then, nuclei regions are segmented from other parts like blood cells. Consequently, the features of nuclei regions are passed as the input of the mentioned classifiers. The achieved classification accuracy of this work was more than 96% by different classification methods. A comprehensive comparison of K-means and Fuzzy C-means algorithms on breast cancer data was conducted by Dubey et al. [42]. K-means algorithm is executed based on distance, threshold, and epoch. On the other hand, the execution of Fuzzy C-means was based on fuzziness value and termination condition. The experimental results showed that Fuzzy C-means method with 97% accuracy outperforms the K-means approach with 92% accuracy. However, Fuzzy C-means requires more computation time compared to K-means algorithm; thus K-means algorithm offers higher performance.

2.1.2 Supervised Learning

SVM is a supervised machine learning technique. It has been effectively applied to both regression and classification problems. Applications of SVM include face detection [43], online hard-writing recognition [44], and bio-informatics [45]. The capability of SVM for binary classifications has allowed it to be broadly used for breast cancer diagnosis. In [16], Akay et al. used SVM technique to provide a breast cancer diagnosis system. As the choice of feature selection enhances the accuracy of the SVM [46], the authors proposed an improved classification system that combines SVM with feature selection to classify breast images. Their proposed model showed a classification accuracy of 98.53% for a subset of five features from WBCD dataset [47]. In other works by Ding et al. [48], a combination of multiple instance learning algorithms and SVM is used for breast ultrasound image classification. Their technique achieved classification accuracy was 91.07%.

Shirazi et al. [49] proposed a model based on SVM and a mixed gravitational search algorithm for tumour detection in breast mammography images. The main objective of this work was to improve the SVM classification accuracy by reducing the number of features. The experimental results of this work show that SVM with mixed gravitational search method obtained 93.10% Accuracy. In [50], Taheri et al. classified breast mammograms into normal and abnormal images accurately by using SVM technique and removing non-necessary parts of the images to reduce the computational complexity. Zheng et al. [51] proposed a hybrid algorithm with K-means and SVM algorithm (called K-SVM) to extract features and diagnose breast tumours. The K-means algorithm distinguishes the hidden patterns for both Malignant and Benign tumours. Using this extracted information, SVM is able to provide an accurate classification. K-SVM improves the accuracy to 97.38% on WBDC dataset. In another research, the linear-kernel SVM algorithm, proposed by Verma et al. [52], gained 90.3% accuracy for breast cancer diagnosis.

Wang et al. [53] worked on a SVM-based ensemble learning technique for breast cancer diagnosis on multiple datasets like WBDC. In this study, instead of using common ensemble mechanisms such as adaptive boosting and bagging [54] classification tree, a novel ensemble learning model called Receiver Operating Characteristic Curve Ensemble (WAUCE) is combined with SVM. This decreased the variance by 97.89% and improved accuracy by 33.34% compared to other single SVM models. In another study, Mavroforakis et al. [55] provided a comparison of different linear and non-linear classifiers, including K-nearest neighbours, Linear Discriminant Analysis, Least Square Minimum Distance, ANN, and SVM for the breast tumour classification. Their results showed that SVM outperformed all other methods by achieving a classification rate of 83.9%.

Bayesian classifier, a statistical method based on Bayes theorem, is another machine learning technique used for breast cancer classification problem. This method is not based on explicit decision rules and instead works by estimating probabilities. Naive Bayes method is the most common type of Bayesian learning algorithms. This model is used for breast image classification by a few research groups. In 2004, Butler et al. [56] used Naive Bayes method to classify X-ray breast images by extracting features from the low-level pixels. Their results show an accuracy of 90.0% for all combinations of features. In [57], Soria et al. made a comparison of three different methods, including decision tree C4.5, multi-layer perceptron and Naive Bayes classifier for classification of breast cancer data. They conclude that the Naive Bayes method outperforms the other two models. Bayesian network model is utilized by Rodriguez-Lopez et al. [58] for breast mass diagnosis, as the main symptom of malignancy. The experimental results of this study illustrate that by using a subset of three clinical features, 82.00% accuracy is obtained.

Artificial Neural Networks, (ANN), referring to a computational technique inspired by structure and functions of human brain, allows computers to learn and make decisions based on previous experiences in a human-like manner. In the ANN model, the output Y will be calculated using the mathematical form of $Y = g\ (\Sigma)$, in which $\Sigma = W.X$, $X = \{x_0, x_1, ..., x_n\}$, and $W = \{w_0, w_1, ..., w_n\}$. Note that, g is a function called activation function [37].

ANN is widely and efficiently used for breast cancer classification problem. Earlier work in ANN with back-propagation training approach, which utilizes the error information in ANN has been for mammography [17]. In [59], Rajakeerthana et al. worked on integration of rough set theory and back-propagation to classify breast mammogram images. In this study, rough set theory is used for image segmentation and handling more uncertain data. This classifier obtained 99.20% accuracy.

Nahato et al. [60] proposed RS-BPNN, that combined rough set indiscernibility relation method with gradient descent back-propagation neural network (BPNN) for breast tumour classification. The indiscernibility relation method was used to handle missing values to obtain a reliable dataset, and select attributes from clinical data while the BPNN was used as a classifier on the dataset. RS-BPNN provided 98.6% classification accuracy.

In another study, Bhattacherjee et al. [61], trained a BPNN to obtain an accuracy of 99.27%. A performance comparison of SVM and ANN is done by Ali et al. [62] for the binary classification of WDBC dataset. The results of this study represent that ANN approach outperforms SVM in terms of accuracy, precision, and efficiency for the classification of breast images as either Benign or Malignant cases. Kaymak et al. [63] showed that the accuracy of their proposed method using back-propagation neural network, in which radial basis neural networks were used to improve the performance of the automatic breast cancer image classification, was 70.4%.

A combination of multi-fractal dimensions and back-propagation ANN was proposed by Mohammed et al. [64] for automate identification of ultrasound breast images . The experimental results illustrated an acceptable range of 82.04% for precision. In a recent study, Saritas et al. [65], compared the performance of ANN and Naive Bayes classifiers and reported an accuracy of 86.95% and 83.54% with ANN and Naive Bayes algorithms, respectively. Abdelsamea et al. [66] proposed a classifier that modified weights of the network based on the updated class reliability of the neuron together with a supervised learning method. This technique achieved a classification accuracy of 95% for binary classification of breast images.

2.1.3 Deep Learning

Although most of the discussed conventional classification methodologies provide an acceptable classification accuracy, their performance depends on proper data representation and feature engineering using previous expert domain knowledge of the data to create useful features (also called hand-crafted features). This is a complex, challenging, and time-consuming task. The other alternatives that has gained momentum are learned feature techniques such as *deep learning*. *CNN*, as a well-known class of deep neural networks used for the visual imagery analysis, is widely employed by different researchers to classify complex breast images.

Fakoor et al. [67] used a deep learning model to increase the performance of cancer detection and diagnosis by forming the features automatically based on gene expression data. The experimental results showed that deep learning produced better performance than previous classification methods. In [68], Cirecsan et al. used a max-pooling CNN model to detect mitosis in breast biopsies, which outperformed other existing techniques significantly.

Wang et al. [26] studied metastatic breast cancer diagnosis using a combination of deep learning and human pathologist's diagnoses. The authors showed the power of deep learning to increase the accuracy of diagnosis. In [69], an unsupervised deep belief network path followed by back-propagation supervised path was used to develop a CAD system for breast cancer detection. The deep belief network path initialized the network weights. Tested on WBCD dataset, this classifier provided an accuracy of 99.68%.

Spanhol et al. [5] used CNN to classify breast histopathological images, achieved by microscopic examination of breast biopsies. Their proposed methodology works based on image patches extraction for training the CNN system, followed by combining the image patches to classify the histopathological images. The results of this study on the BreaKHis breast cancer [7] dataset indicated that CNN performed better compared to hand-crafted classifiers.

Bayramoglu et al. [70] proposed a CNN-based approach to automate diagnosis of breast cancer in histopathology images, independent of their magnifications. In their proposed model, a single task CNN architecture is employed to find the tumour type, and a multi-task architecture is used to diagnose either tumour type or magnification level simultaneously. A comparison of classification performance of this method and previous models, in which hand-crafted feature extraction techniques are used, represents that the performance of magnification specific model improved by using CNN. Moreover, the advantage of this model is the potential for utilizing additional training data with different magnification levels.

In another study, a CNN model for multi-classification of the breast biopsy images was proposed by Araujo et al. [4] to cover the shortcomings of conventional feature extraction classification techniques. In this model, the features were extracted by recovering the information from nuclei and overall tissue organization. The experimental results of this study showed a classification accuracy of 77.8% for multi-classification of breast biopsies into Normal, Benign, In-situ Carcinoma, and Invasive Carcinoma categories. Also, 83.3% of accuracy is obtained for binary classification.

Bardou et al. [71] compared a hand-crafted SVM approach and a CNN model for automatic classification of breast biopsies. Besides, dataset augmentation methods are utilized to improve the accuracy of CNN, SVM, as well as a combination of CNN and SVM. They concluded that CNN outperformed SVM by gaining 96.15%-98.33% of accuracy for the binary classification of histopathological breast images.

Nawaz et al. [10] created a CNN-based multi-class breast cancer classification. First, they classified breast tumours in Benign or Malignant cases. Then, their model was used to classify the tumours in different categories, such as Fibroadenoma, or Lobular carcinoma. Using the DenseNet and BreaKHis training dataset, their proposed approach provided a high classification accuracy of 95.4%.

An ensemble deep learning-based method is employed by Kasani et a. [72] for binary classification of histopathological biopsy images into Malignant and Benign cases. This approach is implemented on famous pre-trained CNN networks such as MobileNet and DenseNet. In this classifier, the required features for classification are extracted using ensemble technique. Applying on multiple datasets, including the BreakHis, ICIAR, PatchCamelyon and Bioimaging, ensemble deep learning-based method achieved 83.10-98:13% of classification accuracy.

In addition to the aforementioned studies for histopathological breast image classification, there are some other researches in which deep learning approach is investigated for the binary or multi-classification of breast biopsies, aim at improving the cancer diagnosis accuracy [73; 74; 75; 76; 77].

Despite the significant success of the neural network-based classification approaches, still, there are some obstacles regarding using this approach to classify breast images. As mentioned in Chapter 1, the back-propagation techniques, like mini-batch gradient descent and Adam optimizers, are the most regularly used approach for updating the weights of the network during the training process of neural network. However, because of the disadvantages of back-propagation method such as the inability to escape local optima, high convergence time, as well as sensitivity to noisy data, research on alternative training method are pursued to overcome these challenges. Evolutionary techniques, such as GA, is one of the most investigated alternative techniques to overcome the weaknesses of the back-propagation technique in neural network [78; 79]. In the section, we elaborate on the literature in this area.

2.2 Evolutionary Genetic Algorithm

Theoretically, it is expected that CNN should outperform other machine learning techniques for breast biopsy image classification. An important reason for any probable failure to obtain high classification accuracy could be due to the training algorithm used for updating the network weights during the learning process. In fact, many of the existing approaches find the weights of the model by performing a gradient descent algorithm, such as back-propagation. Thus, the limitations of gradient-based techniques for finding the best set of the network weights, including finding local optima instead of global optima and high convergence time, decrease the classification performance [80].

To overcome the limitations of the back-propagation technique, evolutionary algorithms are one of the most investigated alternatives. Literature indicates that multiple evolutionary algorithms such as GA [80; 78], Particle Swarm Optimization (PSO) [81] and Whale optimization [82] are widely used instead of gradient approach in the learning procedure for different applications. GA is a bio-inspired optimization technique that follows the process of natural selection of genes in nature. This technique is well-suited for generating accurate solutions for global optimization and search problems through its operations like evaluation, selection, crossover, and mutation [30]. Table 2.2, provided by [2], represents a performance comparison of gradient descent and GA for updating the weights of neural network, achieved by some previous studies.

In the literature, GA has been used either for updating the weights of the neural network or learning the network structures. In one of the earlier works, Montana et al. [28] used GA to train a feed-forward neural network. The authors showed that in comparison to back-propagation neural network, GA improved the accuracy by optimizing the weights during the neural network learning process. Ahmad et al. [2], compared the performance of gradient descent and GA-based ANN applied on cancer and diabetes benchmark datasets. Moreover, the effect of the crossover on GA performance was tested. Their results illustrated better classification accuracy of GA on cancer dataset, versus a higher accuracy of gradient descent on diabetes images.

Belciug and Gorunescu [29] proposed a combination of neural network and GA to classify the patient dataset into Malignant or Benign and recurrent or non-recurrent breast cancer cases. They designed a multi-layer perceptron using GA to update the network weights during the training phase. The results of this study indicated that

Table 2.2: A performance comparison of gradient descent and GA-based ANN by previous studies [2].

Author	Problem	Conclusion
Cao et al. [83]	Forecast earning per share	GA is significantly better than back-propagation.
Guptaet et al. [84]	Chaotic time series prediction	GA outperforms gradient descend in terms of effectiveness and efficiency.
Ghaffari et al. [85]	Modeling of biomodal drug delivery	predictive ability of GA was more than back-propagation.
Alba et al. [86]	Classification of PROPEN1 medical images datasets	GA achieved a higher classification error rate than back-propagation.
Sexton et al. [80]	Classification of PROPEN1 medical images dataset	GA performs better than back-propagation.
Liu et al. [87]	Genetic-evolved neural network	GA performs better than gradient descend for image classification.

the classification accuracy of the hybrid approach outperformed the traditional back-propagation neural network. Bhardwaj and Tiwari [79] used genetic programming (GP) to optimize a neural network for breast cancer diagnosis. GP was used to optimize the weights and network architecture. The crossover and mutation functions were modified to expand the search area and improve the performance. The results showed 99.26% classification accuracy using 10-fold cross-validation.

GA is also widely used to improve the performance of deep learning approach. Young et al. [88], proposed to use GA for automating model selection in deep learning. Authors concluded that this approach can be more powerful than a random search for finding the best network topology. Ijjina and Chalavadi [89], proposed a model for human action recognition. They minimized the classification error rate by initializing the network weights using GA. They used a gradient descent algorithm for CNN classifiers during fitness evaluations of GA chromosomes. The experimental results of this study demonstrated that the combination of gradient descent and GA provided a recognition accuracy of 99.9%. Xie, and Yuille, in [33], used GA to learn new deep neural network structures automatically, instead of using a manually designed network structures. The proposed method encoded the large and complex networks into a fixed-length binary string. GA, with its defined operators, was used to explore the large search space to find high-quality solutions. The authors mentioned that for future work, GA could be used to learn the network structure and train the network and update weights simultaneously. In another work, Such et al. [78] used a gradient-free GA to optimize the weights of a deep neural network. This technique performed well on deep reinforcement learning problems such as Atari. The proposed method easily evolved networks with more than 4 million parameters and trained the system faster.

Martin et al. [90], proposed an evolutionary approach for automatic deep neural networks parametrization, achieving a classification accuracy of 98.93%. Sun et al. [32] proposed using GA to optimize both architectures and initial weight values of a deep CNN for image classification problems. To do so, they designed an efficient variable-length gene encoding for various building blocks and a new representation scheme to initialize the connection weights of deep CNN to prevent getting stuck in local optima. The results of this study indicated a significant superiority over state-of-the-art algorithms in terms of classification accuracy (error rate) and the number of weights.

In [31], optimizing the weights of ANN using GA for image classification problem was discussed by the author. To the best of our knowledge, there is no such study in combining GA and CNN to improve the breast cancer classification accuracy. Thus, in this work, the proposed approach in [31] is extended by using a combination of CNN and GA for histopathological breast image classification problem.

Chapter 3

Evolving CNN through GA

Histopathological breast image diagnosis plays a crucial rule in improving the survival rate of patients. Integrating the knowledge of pathologists and computerized techniques, the accuracy of image classification can be increased. Therefore, CAD systems, that aim to find and classify the abnormalities in breast biopsies, have received significant attention to help in early breast cancer detection [11]. In CAD systems, classification is one of the most important components. In this thesis, we investigate how to build an accurate classifier for binary classification of breast biopsy images. To do this, we rely on machine learning. Machine learning techniques are inherently capable of handling different classification tasks [37]. We attempt to build a CNN model for breast image classification by using GA to optimize the CNN parameters. We compare our proposed algorithm to the gradient descent algorithm. We use the BreaKHis dataset, containing breast cancer histopathological images for training and testing our proposed model [5].

In this chapter, we will start by discussing the background needed for our proposed approach. In particular, discuss the basics of CNN and GA, two techniques used in this thesis. This is followed by a detailed discussion on our proposed approach, evolving CNN through GA for breast image classification.

3.1 Convolutional Neural Network

Neural network is inspired by the operation of biological neurons in the human brain. Figure 3.1 provides an overview of biological neuron, in which the signal flows from Dendrites (input layer) to Axon (output layer) [1]. CNN is a class of deep learning which almost follows the same principle. This method is highly suitable for working with two-dimensional image classification tasks. CNN accepts the images as input and extracts the required features automatically without any need for hand-crafted feature extraction methods. In general, CNN applies different linear and non-linear mathematical operations on an image to predict the probability class or label of an image.

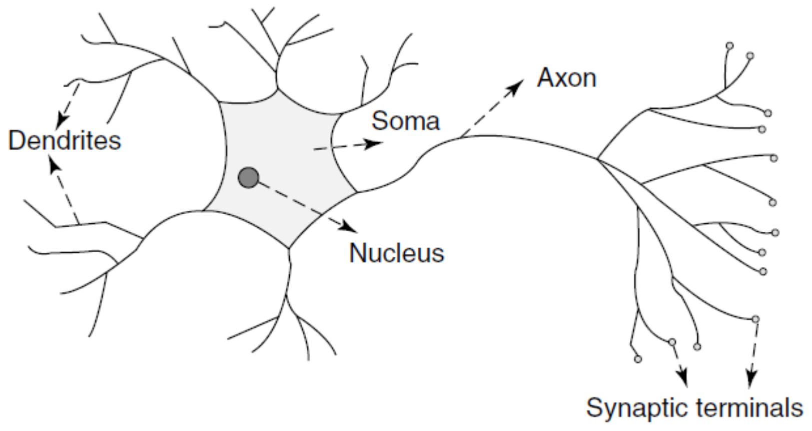


Figure 3.1: Biological neuron [1].

3.1.1 Design

A neural network consists of following layers: input, hidden and output layer. In CNN architecture, hidden layers contain convolution, pooling, flattening and fully-connected layers that transform the input data to the output layer accurately [31]. The convolution layer includes multiple filters of the network weights. Also, the fully-connected layer has a semantic group of nodes that are connected via weighted edges to the nodes in the following and previous layers. Finding the best set of the weights for these convolution filters and connection edges plays a crucial role in the training process of the neural network. Figure 3.2 represents an overview of CNN used for histopathological breast image classification [91].

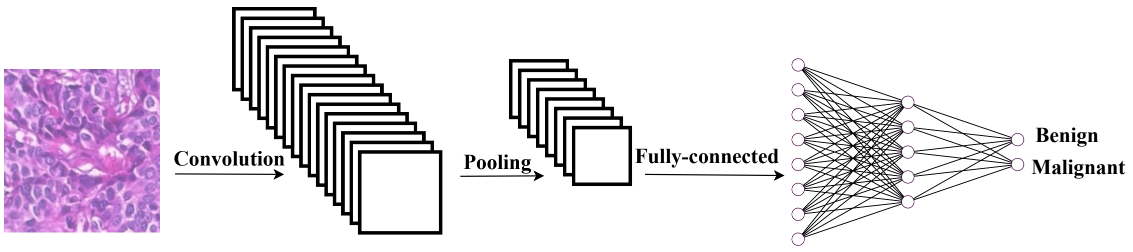


Figure 3.2: An overview of CNN to classify breast biopsy images.

In neural networks, a input vector is transformed with a set of weights similar to a linear function as shown in Equation 3.1. In this equation, y, x, and w refer to the output, input, and weight, respectively. The bias term, b, is a constant added to a linear equation (here product of input features and weights) to increase the flexibility of the model. Bias term is especially helpful when all the feature values are zero. Suppose that in Equation 3.1, we have no bias term; in this case, if $x = 0$, then the output y always would be 0, therefore, we consider a bias parameter to better fit the data.

$$y = w.x + b \tag{3.1}$$

An activation function is a function that determines the outputs of the network based on inputs. This function will be applied to the above linear equation. Activation function decides whether a neuron should be activated (or not activated) based on the weighted sum and bias of the neuron. Without activation function, the model just would be a linear regression model. It means that no matter how many hidden layer is used in the network, the classifier will be equivalent to a linear network without any hidden layer. Therefore, activation function is utilized to introduce the non-linear transformation to the network. Non-linearity means that the output is not generated from a linear combination of the inputs. Adding non-linearity allows neural network learn and perform complex tasks.

There are different types of activation functions, such as linear regression model, non-linear models, Sigmoid functions, and Rectified Linear Unit (ReLU) activation functions. In what follows, two of the most commonly used activation functions are listed.

- **Sigmoid activation function**

  The Sigmoid non-linear function has the following mathematical form:

$$y = \sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.2}$$

  It takes a real value and projects it between 0 and 1. Note that, the gradient of the function for the tail values is almost close to zero (Figure 3.3). The training process, therefore, fails for the features close to this region.

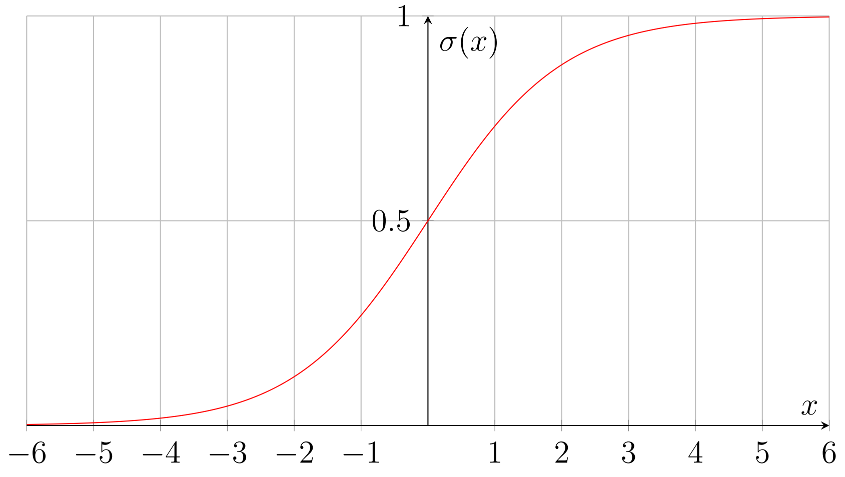


Figure 3.3: Sigmoid activation function.

- **Rectified Linear Unit (ReLU) activation function**

Based on the schematic of the ReLU activation function, represented in Figure 3.4, this function is defined as $y = max(0, x)$. In comparison to the Sigmoid function, the ReLU considerably improves the convergence rate of the training process due to its linear and non-saturating form. Also, ReLU is not computationally intensive. However, the disadvantage is that since ReLU eliminates all the negative information, it is not suited for all dataset and architectures.

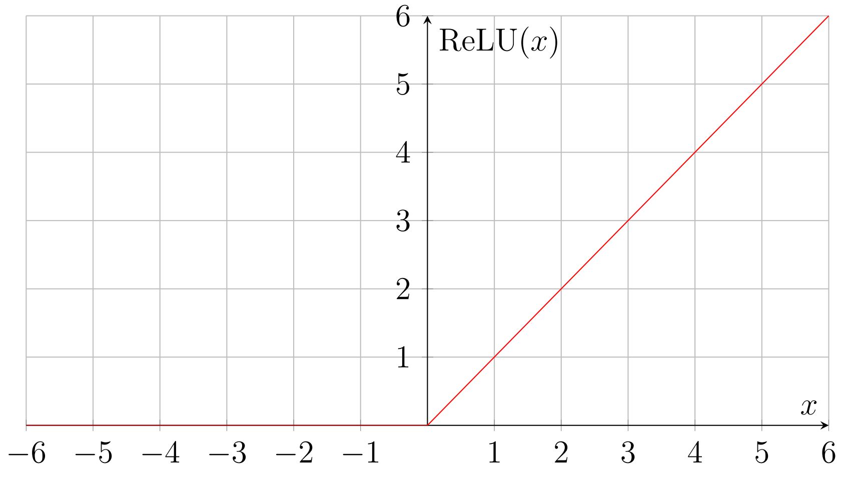


Figure 3.4: ReLU Activation function.

There are three main layers in CNN. These are discussed below.

3.1.2 Convolutional Layers

The convolutional layer is the main component of CNN. This layer converts the input image to a map of features, by performing a linear operation called *convolution*. An image, an array of pixel values, is a matrix with three different layers. The size of the matrix depends on the resolution and size of the image, like $480 \times 480 \times 3$ array of numbers, where 3 refers to the RGB color. A value from 0 to 255 is assigned to each component of the matrix to describe the intensity of each point. To extract the differences between various images, a group of filters are applied to these images. These filters convolve the input and pass the result to the next layer by applying a dot product, *convolution*, over the whole size of each image. The parameters of these filters (i.e. the weights) are initialized randomly and need to be learned by the network subsequently.

3.1.3 Pooling and Flattening

After the convolutional layer, a pooling layer is added to the network, specifically, when a non-linear activation function has been applied to the feature maps. In this step, a pooling operation will be selected to be applied on feature maps aimed at reducing the number of features. The size of this filter is less than the size of the feature map; specifically, it is usually 2×2 pixels. Following functions are commonly used as the pooling operation for different applications:

- Average Pooling: Calculating the average of each patch of the feature matrix.
- Maximum Pooling: computing the maximum value of each patch of the feature matrix.

After convolution and pooling steps, the entire obtained feature map matrix will be transformed into a single vector using a procedure called *flattening*. This vector is then fed to the fully-connected layer for further processing.

3.1.4 Fully-Connected Layers

After flattening, the obtained feature vector is sent to the fully-connected layer as its input. This input vector, then, will be connected to the output layer via a fully-connected hidden layer. Consequently, classification error rate will be calculated using the predicted and actual classes. The recursive training process will continue until finding the highest classification accuracy.

3.1.5 Training Methods

Based on the definition of the CNN, filters and neurons carry a set of weights. The process of adjusting these weights using available dataset is named as the training process. These weights are initialized randomly at the beginning and are updated using different optimization techniques in a recursive process. Back-propagation is one of the most commonly used techniques in the CNN training process. In what follows, two instances of this approach are explained.

Gradient descent

A vector indicating the direction with the largest changing value at one point refers to the gradient or the rate of change of a function. Gradient descent is frequently used in neural networks to define the training process of the networks. This deterministic approach works based on the following process:

1. Initialize the weights, with a random or an all-zero vector.
2. Adjust the weights so that the loss function decreases in the direction in which the gradient decreases.

To quantify the capacity of the architecture in approximation of the ground truth labels for all training inputs, we define a *loss function*. The number of incorrectly classified images could be used as a simple loss function. Defining a precise loss function leads to an accurate trained model. Mean Square Error is one of the most used loss functions. The gradient descent algorithm is used to minimize the loss function $J(w)$ as shown in Algorithm 1 [92]. The value of $J(w)$ has to be made as small as possible to achieve the highest classification accuracy.

Algorithm 1 Gradient descent

```
Initializing the network weights randomly;
Adjusting the weights to decrease the loss function:
for $k = 0, 1, \ldots, n$ do
    $g_k \leftarrow \nabla J(w_k)$    {$n$: number of iteration, $w$: weight, $J()$: loss function}
    $w_{k+1} \leftarrow w_k - \alpha g_k$    {$\alpha$: learning rate}
end for
```

Figure 3.5 shows how the gradient descent algorithm proceeds. At each point, the gradient (dashed-line) is computed and a step is taken in the opposite direction until the algorithm stops at some point near the minimum. For each training data, the prediction and it's associated loss values are computed. All the loss values are added to compute the final error value. The back-propagation is used in the next step to propagate the error in order to compute the partial derivatives of $J(w)$ for all the weights. α is the learning rate and determines the step size of each iteration. A very small learning rate slows the convergence and makes the computation intensive, while a large value for α may decrease the chance of reaching to the global minimum.

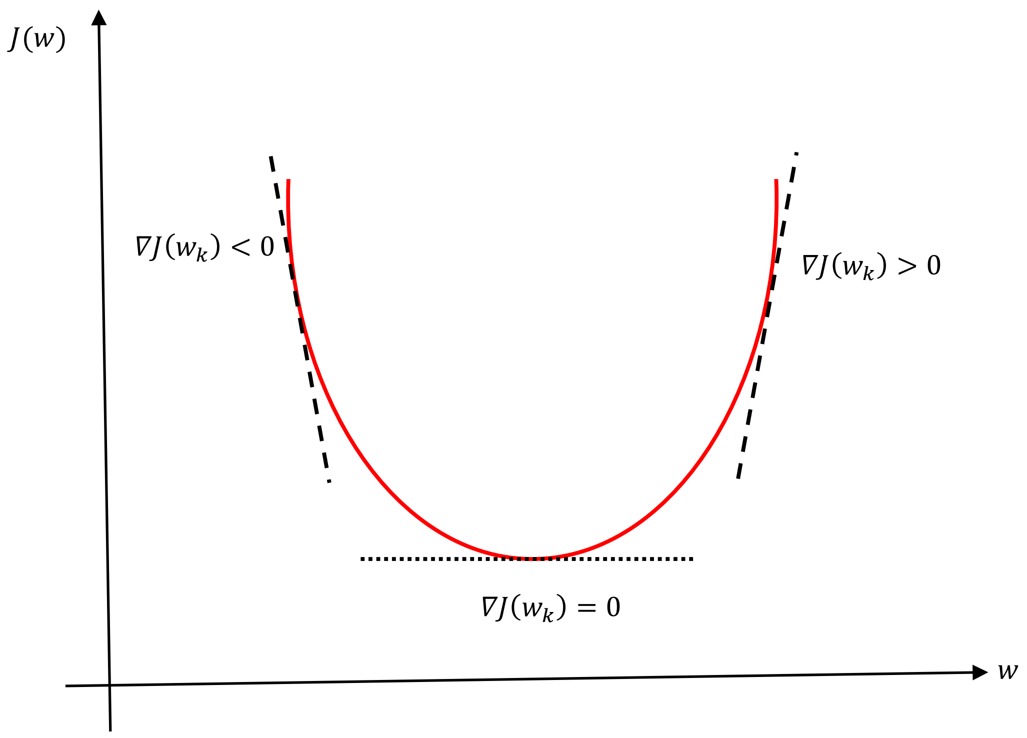


Figure 3.5: Demonstration of gradient descent.

For large datasets, gradient descent is computationally intensive, since the whole dataset has to be trained at once. To overcome this problem, we can take the advantages of stochastic gradient descent, which selects a subset (or batch) of training data randomly to train the model. Stochastic gradient descent often converges much faster compared to gradient descent since it does not need to train the whole dataset at the same time. Mini-batch gradient descent in which the batch size is set to greater than one and smaller than whole dataset, is the most frequently used variant of stochastic gradient descent. Batch size of 32 is a good default batch value for mini-batch gradient descent [93]. However, all gradient based approaches may trap at local minima rather than finding global minima.

Adam

A constant learning rate is considered during the training process for all the weights and biases in both gradient descent and stochastic gradient descent techniques. Choosing the proper learning rate is fairly difficult because this rate changes the learning process significantly. Choosing a small constant learning rate that gives stable convergence is a viable solution, although this increases the computation time. To overcome this problem, the learning rate could be set as a small value for some initial epochs (each epoch refers to a full pass through the training set) and then choose a smaller value as convergence slows down for the rest of the training process. This approach uses the first and second derivative of the weights to define an adaptive learning rate for all the parameters changing over time.

The *Adaptive Moment Estimation* (*Adam*) algorithm, is an extension of the stochastic gradient descent optimization approach, proposed by Kingma and Ba [34] in 2014. This algorithm follows the idea of computing specific learning rates for each parameter. *Adam* uses the first-order gradient-based optimization. This is a straight-forward and efficient method with low memory usage. For the cases with large dataset, *Adam* fairly speeds up the training process. Although *Adam* can train the models faster than other techniques, it may still get stuck in local optima [94].

3.2 Genetic Algorithm

GA is a class of evolutionary algorithms that follows Darwin's natural theory of evolution. It is a powerful technique for solving different search problems by finding globally optimal solutions [95]. In this approach, the fittest individuals will be selected to produce offsprings for the next generation. GA consists of four main phases:

- Creating a random population of individuals.
- Evaluating and selecting the individuals based on the fitness function.
- Crossover.
- Mutation.

Typically, GA begins by producing a random set of individuals, called *population*. Then, these individuals are evaluated based on a fitness function. The fitness function determines the strength or weakness of the individual. Using this information, the individual may or may not be used in future solutions. The individuals with better fitness values have more chances of being selected as parents for reproducing the offsprings for the next generations than individuals with lower (i.e. poor) fitness values. After evaluating the individuals, evolutionary operations including selection, crossover, and mutation are applied to select the fitter parents, produce the new offsprings for the next generation, and increase the diversity of the population to improve and find the optimal or near-optimal solutions [95]. In what follows, the steps of GA will be discussed briefly.

3.2.1 Initial Population

The first step in GA is generating an initial population of individuals randomly. This helps in incorporating a diverse range of possible solutions for the problem. The variables that define the characteristic of an individual are called *genes*. A collection of genes form a *chromosome*, i.e. solution to the search problem. Finally, a set of chromosomes form the initial population. There are different types of gene representations in GA, such as binary, real-coded, or integer representation.

3.2.2 Evaluation and Selection

The evaluation of an individual is done using an evaluation operator, the fitness function. The fitness function measures the ability of an individual to generate an optimal solution. In this phase, the fitness value of each individuals are calculated and sorted. As a result, individuals with better fitness values will receive more opportunities to be selected for reproducing the offsprings in the next generations. Fitness functions differ depending of the application and the search problem. For example, the classification error rate is the most commonly used fitness operation for different types of classification problems.

After calculating the fitness values, selection phase is executed. In the selection phase, the mating process is determined to select fitter solutions for producing the offsprings of the next generation. Two pairs of individuals with better fitness scores are selected for passing their genes to the next generation. One of the most common selection technique is tournament selection that uses random selection to select the parent individuals for breeding. Other techniques include roulette wheel selection or truncate selection .

3.2.3 Crossover

Crossover operation is considered as one of the most critical component in GA. The action of this genetic operation is generating two new children chromosomes using the genes of two parent chromosomes. Among the different types of crossover operations, single-point crossover is the most frequently used approach. In a single-point crossover, a random crossover point is selected in each of the individuals. Then, the genes of each side are swapped between parents to generate new offsprings [96]. Figure 3.6 illustrates an example of this technique.

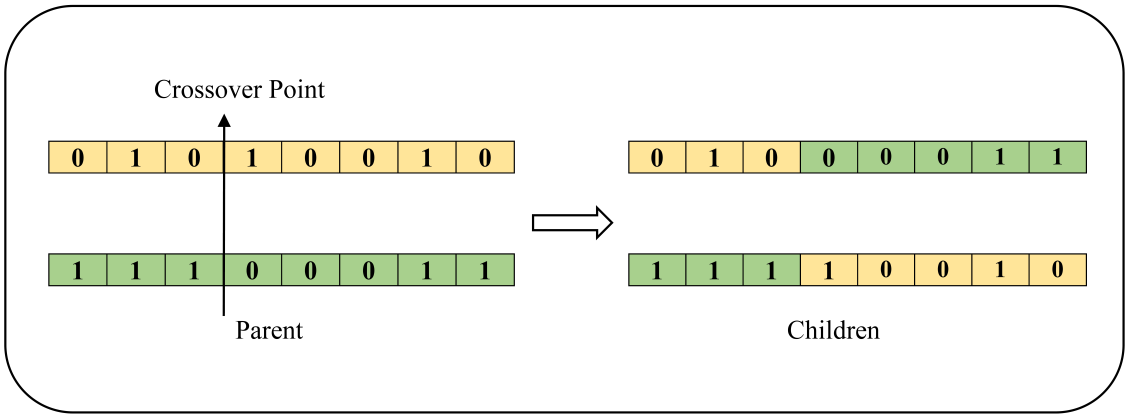


Figure 3.6: An example of single-point crossover in GA.

3.2.4 Mutation

After producing the new offsprings, some of the genes in the new individuals are mutated to increase the diversity in the population from one generation to another. Increasing the diversity [97] of the population can produce stronger individuals or solutions. Flip bit, inversion and random mutation are some of the most common types of mutation operations used for real-coded or binary representation in GA. In these techniques, a parameter within an individual is changed by flipping or replacing with a randomly generated value [98].

3.3 Combining CNN and GA

In this section, we describe how we use the global search capability of GA to evolve the CNN weights for histopathological breast image classification problem. A CNN architecture suitable for binary classification of histopathological breast images is designed.

Figure 3.7 shows the block diagram of our model. The layers used by the model are described below.

- Input layer: This layer loads the input images and passes them to the convolutional layer. In our design, inputs are breast biopsy images from the BreakHis dataset with the dimension of 210×210×3.

- Convolutional layers: Convolutional layers apply convolution operation on input images using kernels (filters) to generate the feature map. The network consists of 8 convolutional layers with multiple filters of size 3×3. The number of filters in each layer is as follows: 8 filters for the first two layers, 16 filters for the third and fourth layers, 24 filters for the next two layers, and finally 32 filters for the last two layers. These filters of weights are initialized randomly using either uniform or normal distribution.

  Since using matrix form makes the operation of CNN easier, all the weights

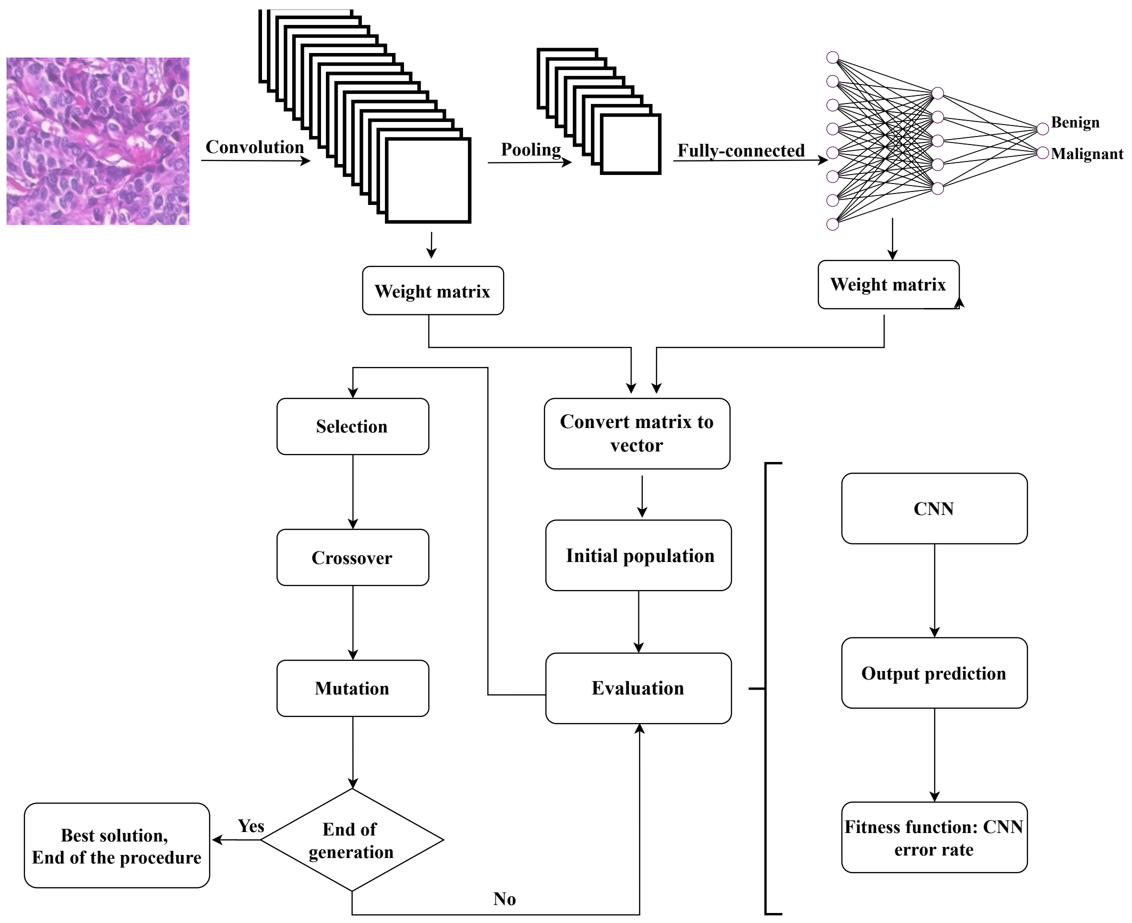


Figure 3.7: Block diagram of optimizing the parameters of the CNN using GA.

of the network, including the weights of the convolutional filters and fully-connected layers store in a matrix for further computation. However, the initial population of GA is stored in 1-dimensional vectors. Later in this section, we will explain how each weight matrix needs to be converted into a 1-dimensional vector to be used as the initial solution of GA [31].

- Pooling layer: This step is performed after the convolution layers step. As explained in Section 3.1.3, an *Average pooling* function is applied to reduce the features map dimension. This function calculates the average of each filter of convolutional layer.

- Activation layer: All convolutional and pooling layers are then followed by a *ReLU* activation function (Section 3.1.1) to improve the converge rate of the learning process.

- Fully-connected layer: After convolution and pooling steps, the entire matrix of features is transformed into a single vector by applying a procedure called *flattening* (Section 3.1.3). This vector is then fed to the fully-connected layer of the network as its input neurons. In other words, the first layer is a vector of features from convolutional and pooling layers and the last layer is a vector of size two which is the number of the classes in our binary classification, i.e. Benign and Malignant classes.

- Training process: After creating the CNN blocks, the whole system is trained. Training refers to the procedure of updating the weights of the network until finding the most accurate output. In our network three different optimization techniques are utilized and compared in terms of classification accuracy: mini-batch gradient descent, Adam optimizer and GA.

GA evolves the population during run time and improves the solutions through several generations. The classification accuracy in CNN highly depends on the weights in all layers. We improve this model using GA. All initial weights of the neural network, including the weights of filters as well as fully-connected layer are stored in a weight matrix [31]. A single solution of our model includes 26,125 weights. We have at least 20 solution per population which means a total minimum of 522,500 parameters.

The weight matrix is converted to a vector to be used as the initial population of GA as indicated in the flowchart in Figure 3.7. This conversion is done using a Python function from Numpy library, called *numpy.reshape*(●). The inputs of this function are the weight matrix and the size of output vector. The output is a 1-dimensional vector with size equal to the total number of components of the weight matrix. In our case, each 1-dimensional vector contains 26,125 elements. Thus, this vector will be used as the starting solution of the GA. A real-number representation of the genes is used to represent this vector of network weights. That is, each gene (or weight) in chromosome (or solution) is a real number generated by a random uniform distribution.

Then to find the best set of the network weights, each solution is evaluated by a fitness function (”Evaluation” block in Figure 3.7). That is, each set of weights is used to construct the corresponding neural network structure. The fitness of this solution is then calculated based on the classification error rate as shown in Equation 3.3. The best error rate is 0.0, and the worst is 1.0 in which all predictions are wrong. After the evaluation step, the solutions are sorted based on their fitness values. Consequently, fitter individuals, are selected to generate new offsprings.

$$Classification\ error\ rate = \frac{False\ positives + False\ negatives}{Total\ number\ of\ samples} \tag{3.3}$$

The selected solutions generated by the evaluation process are improved through GA. The crossover and mutation operations of the algorithm, are applied on selected solutions. Single-point crossover operator is performed on randomly selected parent chromosomes to generate new offsprings for the next generations. Mutation operation is used to change a single gene in each offspring aimed at increasing the diversity of new generation. This operation adds a random value, generated by a uniform distribution, to a randomly selected gene. The iteration of evaluation, selection, mutation, and crossover is repeated (as shown in the flowchart) until the best set of the weights are found, producing the lowest classification error rate.

Algorithm 2 Optimizing CNN through GA

```
Create the CNN architecture;
Generate the initial weights of the model randomly;
Store the weights of the model in a matrix;
Set the parameters of GA;
Convert the weight matrix to the vector of initial population;
while Termination condition is not true do
    Start the CNN process;
    Predict the output using CNN;
    Calculate the fitness function using equation 3.3;
    Sort individuals based on fitness value to select fitter individuals for producing
  the offspring of the next generation;
    Applying a single-point crossover operation;
    Applying mutation operation to add a random value to a randomly selected
  gene with probability of 0.1.
end while
```

Algorithm 2 describes the algorithm for optimizing CNN through GA. We use the following parameters in the proposed GA optimizer implementation: (i) number of solution per population varies between 20 to 60; (ii) number of generation is changed from 100 to 1000; (iv) mutation rate rate varies from 0.1 to 0.01; (v) network weights are initialized using both normal and uniform distributions.

The next chapters discuss the implementation, results and evaluation based on the above user-defined parameters.

Chapter 4

Evaluation and Dataset

We evaluated the performance of our proposed classifier on the BreakHis [7] dataset. The performance analysis is conducted according to the evaluation metrics on the test dataset. We used different evaluation metrics, including classification accuracy, recall, precision, F1-score and execution time. The main focus of this study will be on the classification accuracy metric. In what follows, we will describe the evaluation factors we used here. Moreover, we will introduce the BreakHis dataset thoroughly.

4.1 Evaluation Metrics

For evaluating the proposed model, we consider the basic performance measures obtained from confusion matrix [99]. The confusion matrix is a table containing the outcomes of a binary classifier on test data. This matrix summarizes the number of correct and incorrect predictions achieved by a machine learning-based classifier.

The confusion matrix indicates the performance measurement for our application. Using this matrix, we can derive multiple performance metrics, including classification accuracy, recall, and precision.

The confusion matrix contains four components that are the outcomes of the binary classification:

- TP: True-Positive prediction
- FP: False-Positive prediction
- TN: True-Negative prediction
- FN: False-Negative prediction

Table 4.1 represents an overview of the confusion matrix for breast cancer binary (Benign and Malignant) classification problem. The outcomes are different combinations of predicted and actual values.

Table 4.1: Confusion matrix for breast cancer classification problem.

Predicted / Actual	Benign	Malignant
Benign	TN	FP
Malignant	FN	TP

Accuracy

Classification accuracy is one of the important metrics to evaluate the performance of a classifier. This metric illustrates how accurate data will be classified using a particular classification model. Classification accuracy refers to the ratio between the number of correct predictions and the total number of predictions generated by a classifier as shown in Equation 4.1. The best classification accuracy is 1, while the worst is 0.0 [99].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Recall and Precision

The breast cancer classification problem is an imbalanced classification problem. That is, the two classes we need to identify-Benign and Malignant-are different and one is more important than the other; since we can probably tolerate the false-positive predictions, but not false-negative ones. Because false-negative prediction means that the tumour is cancerous, while our classifier predicted it as a non-cancerous case. This wrong prediction may increase the treatment cost and cancer mortality rate.

Metrics such as recall and precision measure the relevance of the classification. Recall, also known as sensitivity in binary classification, illustrates the proportion of the total number of correct positive predictions and the total number of positive cases (both true-positive and false-negative). A low recall rate means a large number of false-negative predictions, which is intolerable and leads to irreparable consequences for the treatment process. On the other hand, precision refers to the fraction of the total number of correct positive predictions and the total number of predicted positive cases (i.e. both true and false-positive predictions). A low precision rate indicates a large number of incorrect positive predictions, which means that the tumour is

Benign but predicted as Malignant. Although low precision is not our preference, but would be tolerable to some extent, since it does not have the risk of death. Recall and precision are given by Equation 4.2 and Equation 4.3, respectively [99; 100]:

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

$$Precision = \frac{TP}{TP + FP} \tag{4.3}$$

Furthermore, the balance between recall and precision can be calculated using a metric called F1-score. F1-score is a realistic measure to test the accuracy of the model by considering both recall and precision rates. F1-score is given by Equation 4.4.

$$F1 - score = 2 \times \frac{Recall \times precision}{Recall + Precision} \tag{4.4}$$

The best value of F1-score is 1 which means a perfect recall and precision, while the worst value is 0.0.

Other Metrics

Execution time is another performance metric to evaluate the proposed model. Execution time is defined as the total required time to complete a particular task by the central processing unit (CPU). This metric will be calculated based on the difference of the start time and the end time spent by CPU in executing a program, without considering the I/O waiting time or time required to complete other jobs as shown in Equation 4.5. The execution time measures the computation of the proposed algorithm on a given dataset.

$$ExecutionTime = start\ time - end\ time \tag{4.5}$$

Besides considering the above-mentioned machine learning metrics, extensive experiments are performed for evaluating the impact of different GA parameters on the performance of the proposed classifier, such as number of generation, number of solution per population, mutation rate, and random number generator methods.

4.2 Dataset

The BreakHis dataset, a public dataset available at `http://web.inf.ufpr.br/vri/databases/breast-cancer-histopathological-database-breakhis`, is used to evaluate the performance of our proposed model for binary classification of histopathological breast images. Researchers need a unique dataset to evaluate the performance and prove the effectiveness of their proposed classifiers. Therefore, the BreakHis dataset, containing breast cancer histopathology images, is introduced by Spanhol et al. [7] as a well-known database for breast cancer classification problem. Figure 4.1 indicates some samples of Benign and Malignant cases provided by the BreakHis dataset.

As presented by Table 4.2, the BreakHis dataset includes 7909 breast cancer histopathological images of size 700×460, obtained from 82 patients. This dataset has both Benign and Malignant classes which makes it a suitable choice for binary

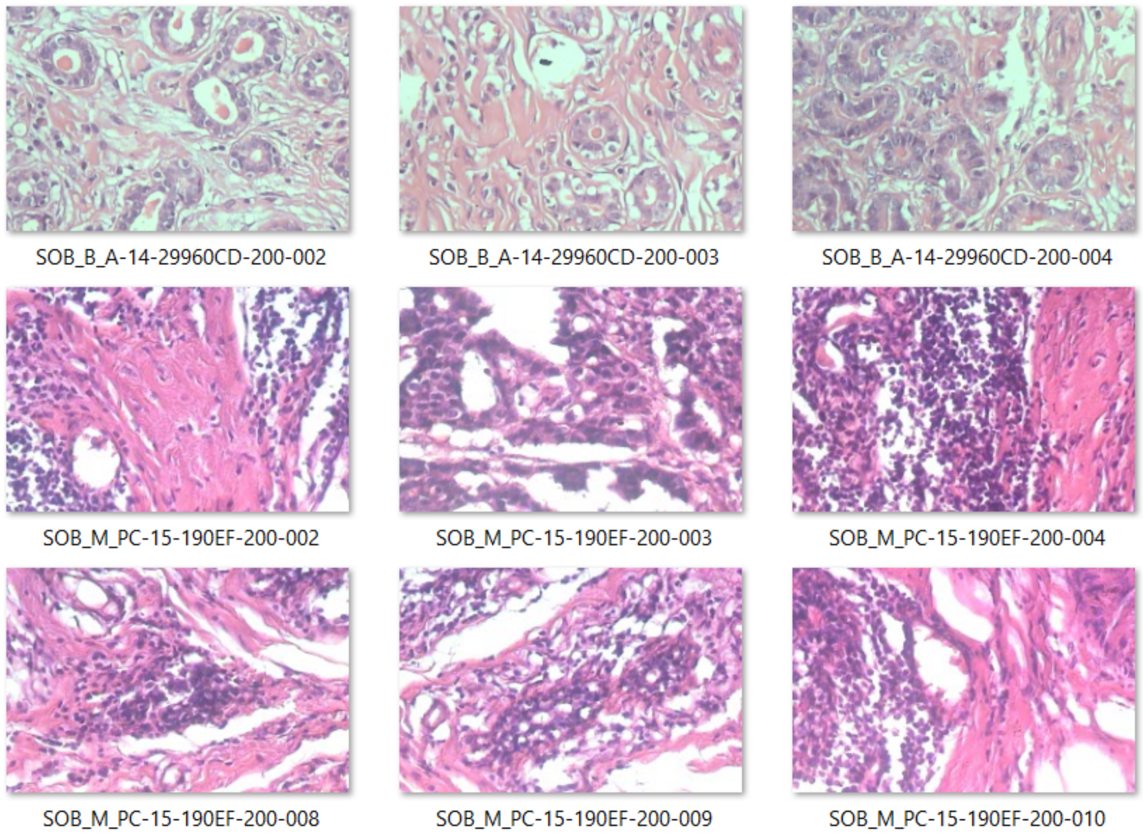

SOB_B_A-14-29960CD-200-002
SOB_B_A-14-29960CD-200-003
SOB_B_A-14-29960CD-200-004
SOB_M_PC-15-190EF-200-002
SOB_M_PC-15-190EF-200-003
SOB_M_PC-15-190EF-200-004
SOB_M_PC-15-190EF-200-008
SOB_M_PC-15-190EF-200-009
SOB_M_PC-15-190EF-200-010

Figure 4.1: Slides of breast Benign and Malignant tumours with magnification factor of 200X in BreahHis dataset.

classification. Moreover, it contains multiple sub-classes of cancer types, such as Fibroadenoma, Adenosis, Phyllodes tumour, Tubular Adenona, Carcinoma, Lobular Carcinoma, and Papillary Carcinoma which can be used for multi-classification of breast biopsies. Also, the images are categorized based on their biopsy procedures as well as magnification factors (i.e. 40X, 100X, 200X, and 400X). The BreakHis dataset is widely used to design a valuable CAD system for automated classification of breast biopsies [5; 70; 20].

Table 4.2: Structure of BreakHis dataset.

Benign	Malignant	Total Number of images
2,480	5,429	7,909

In this study, we randomly divided the BearkHis dataset into two sets, including training set and testing set, so that 70% of the existing images are used to train our classifier and the rest 30% are used to test the proposed model. These two sets are separated patient-wise, which means that patients used to create the training set and the test set are not the same. Moreover, as our focus is on the binary classification of breast histopathological images into Benign and Malignant cases, we are categorizing the breast biopsy images independent of their sub-classes and magnification factors.

Chapter 5

Experimental Results

In this chapter, we present the results of histopathological breast image classification, provided by the BreakHis dataset [7], based on different metrics such as classification accuracy, recall, precision, F1-score and execution time discussed in the previous chapter.

5.1 Accuracy

We train our proposed CNN model using the BreakHis dataset images as input and three different optimization approaches: mini-batch gradient descent optimizer and Adam optimizer, discussed in Section 3.1.5, and GA. The output is the binary classification of the input images. Each image is passed to the network and labelled as a Benign or Malignant case. The predicted labels are then compared to the actual labels, provided by the dataset, to determine the classification accuracy. Then, the three discussed optimization techniques are evaluated by comparing the provided classification accuracy, recall, precision, F1-score and execution time on an identical test set from the BreakHis dataset.

As mentioned in Chapter 4, classification accuracy is the metric to evaluate the performance of different classifiers. The results shown in Table 5.1 presents the highest classification accuracy achieved by each optimization approach, i.e. mini-batch gradient descent, Adam, and GA. As the table demonstrates, Adam optimizer, with 0.8583 of the classification accuracy, outperforms mini-batch gradient descent and GA methods, and mini-batch gradient descent provides lower classification accuracy than the other two optimizers. On the other hand, our proposed GA-based classifier performs almost as powerful as Adam optimizer with negligible difference. The best accuracy for mini-batch gradient descent and Adam is obtained when the batch size is equal to 32, whereas GA provides the best results with a batch of size 128.

Table 5.1: Classification accuracy of mini-batch gradient descent, Adam, and GA on the BreakHis dataset.

Optimizer	Batch size	Iteration	Accuracy
Gradient descent	32	300	0.6988
Adam	32	400	0.8583
GA	128	300	0.8549

To do this experiment, first, we ran the network using a mini-batch gradient descent technique to update the weights of the model. These weights are initialized randomly following an uniform distribution. The classification error rate is considered as the loss function of the network. The learning rate is set to 0.001. Learning rate is a hyper-parameter referring to the step size in each iteration for updating the parameters of the model in order to minimize the loss function. Finding the proper learning rate is a challenging task, since the large value may prevent the convergence and very small value makes the training process significantly longer, without improving the accuracy of the model. The common learning rate used to train the CNN models varies from 0.1 to 0.0001. As 0.001 is introduced as a reasonable base learning value by some researches [101; 102], in this study, we set the learning rate equal to 0.001 as well.

As explained in Chapter 3, the weights of the model are updated after passing a batch of the training images through the network instead of a single image at a time. While a batch of size 32 is a good default batch value [93], we also tried higher batch sizes, i.e. 64 and 128. Note that we experimented with these batch sizes following the work by Radiuk et al.[103] to study the impact of batch sizes on the CNN model.

The number of iterations is set between 100 and 1000 for this optimizer. The approach used to select this range was by starting from a small number of iterations and then increasing it gradually (in our case from 100 to 1000) to find the proper number of iterations providing the best performance and accuracy. For some batch sizes, we also tried 2000, 5000, and 10000 number of iterations. However, these iterations needed a very long computation time and did not provide higher classification accuracy than what we obtained from iteration numbers between 100 and 1000. Thus, the long execution time limited us to try with higher number of iterations. We believe that the long time computation could be resolved through parallel processing, but since these iterations do not provide the desired accuracy, we only concentrated on the ones that were useful.

As can be seen from Table 5.1, the best classification accuracy obtained by mini-batch gradient descent is 0.6988. Using the same configuration, Adaptive Moment Estimation optimizer (Adam) is then used to train the model. As explained in Section 3.1.5, Adam is an extension of the stochastic gradient descent optimization approach. The only difference between gradient descent and Adam is that the Adam optimizer uses an adaptive learning rate during the training process for each parameter of the network instead of a fixed learning rate.

Finally, we trained our CNN model using GA. The weights are initialized randomly by uniform distribution. Solutions per population and number of parents mating are set to 40 and 8, respectively. To select these values, we followed what is used in [31] to optimize a ANN network using GA. The difference is that since our CNN model has more parameters to update, we used a larger initial population. The number of parents mating is defined based on our training results. Since our experimental results indicated that in each iteration, almost 20% of solutions produce more acceptable classification accuracy, so we decided to use them for producing offsprings of the next generation.

Single point crossover is used for offspring reproduction followed by a mutation operation, for adding a random value to a randomly selected gene, with a probability of 0.1. This value is recommended as a typical base value for mutation probability by literature [104]. The classification error rate is considered as the fitness function for the solution evaluation. These experiments are done in batches of size 32 to 128, for several generations between 100 to 1000. Since the results of Adam optimizer and GA are better than the gradient approach, their results are presented in detail at Table 5.2.

Table 5.2 illustrates that Adam optimizer provides the best accuracy (0.8583) for batch size of 32 and 400 iterations. For the GA, this is not the case. The batch size of 32 does not produce a good accuracy unless the number of iterations increased to 1000. The best accuracy (0.8549) is obtained for 128 batches with 300 iterations. As can be seen, the GA performs equally well in comparison to Adam optimizer for batch sizes 128. However, for other smaller batch sizes, the accuracy of GA is better only for larger number of iterations. For example, for 32 batches, compared to 0.7044 for 300 iterations, 1000 iterations is 0.8214.

We can observe that for both optimizer the overall classification improves for larger batch sizes. For a particular batch size, for smaller iterations, the accuracy is low. For example, for 100 iterations, batch size of 32 or 64, the accuracy is too low compared to higher number of iterations for the same batch sizes. However, for the same number of iterations, 100, for batch size 128, the accuracy for both optimizer is close to those in higher iterations. From these observations, we can conclude that larger batch sizes provide better accuracy than smaller batch sizes.

The pattern we see in the table with larger batch sizes matches the results obtained by Radiuk et al.[103], in which the impact of batch size on the performance of the CNN is studied. They conclude that by increasing the batch size, accuracy will increase, as well. We hypothesize that the reason for this improvement in accuracy and with increasing batch sizes, could be due to the way in which the gradient of the

Table 5.2: Classification accuracy achieved by Adam and GA optimization approaches on the BreakHis dataset.

Batch size	Iteration	Adam Accuracy	GA Accuracy
32	100	0.6873	0.7160
32	200	0.6320	0.7414
32	300	0.8453	0.7044
32	400	0.8583	0.7893
32	500	0.8301	0.8274
32	1000	0.8230	0.8214
64	100	0.7985	0.7490
64	200	0.8510	0.7517
64	300	0.8410	0.8030
64	400	0.8092	0.8501
64	500	0.8480	0.8415
64	1000	0.8274	0.8358
128	100	0.8356	0.8150
128	200	0.8563	0.8528
128	300	0.8446	0.8549
128	400	0.8569	0.8198
128	500	0.8411	0.8487
128	1000	0.8294	0.8297

loss function is calculated. When the batch size is a large number, a more accurate gradient can be obtained since we will update the weights after passing a large number of images through the model. Moreover, increasing the batch size decreases the chance of trapping in a local minimum, because the updates are done after training more images from the dataset. Consequently, these updates are globally (for the whole dataset) acceptable (or good). On the other hand, when batch size is a small number, the model is updated more frequently. Note that more updates are not necessarily better than fewer updates for large batch sizes. Actually, there is a trade-off between the more bad updates versus the less good updates.

Furthermore, the results demonstrate that in general increasing the number of iterations, improves the classification accuracy as well, since more images are used to train the model and the learning process is applied to the entire dataset many more times. As a result, the system learns better.

However, if we train the network with either a very high number of iterations or a too large number of batch sizes, the model will probably get over-fitted. Over-fitting refers to the situation in which the classifier does not generalize well between training data and test data. It means that there is a significant gap between training accuracy and test accuracy because the model did not learn the data and memorized it. As Table 5.2 represents, in the most cases, by increasing the number of iterations, for example, from 500 to 1000 for batch size 128, the accuracy drops. We hypothesize that by increasing the number of batch sizes with increasing number of iterations, over-fitting would lead to poor accuracy.

Our proposed algorithm, performs as well as the Adam optimizer and follows similar pattern as that of the Adam optimizer. Unlike the Adam optimizer, the GA requires higher batch sizes for training the model. This is because, in general, in GA, higher diversity is needed to produce accurate results. It also requires larger number of iterations to maintain stability. Therefore, this is not a surprise for this application. Thus, GA requires higher batch size (128) for training the model to gain the highest accuracy, 0.8549. We can also conclude that the accuracy classification for both optimizers is low for smaller batch sizes with smaller number of iterations. Figure 5.1 and Figure 5.2 shows the graphical representation of the results in Table 5.2 for both Adam optimizer and GA, respectively .

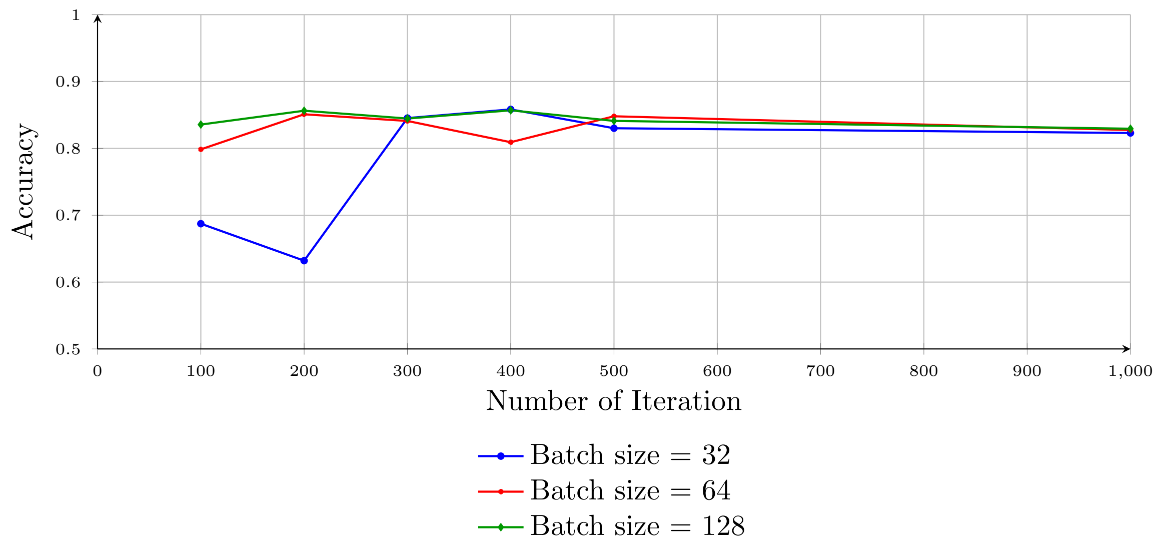


Figure 5.1: The testing accuracy of the trained CNN with Adam and batch size values of 32, 64, and 128 on the BreakHis dataset.

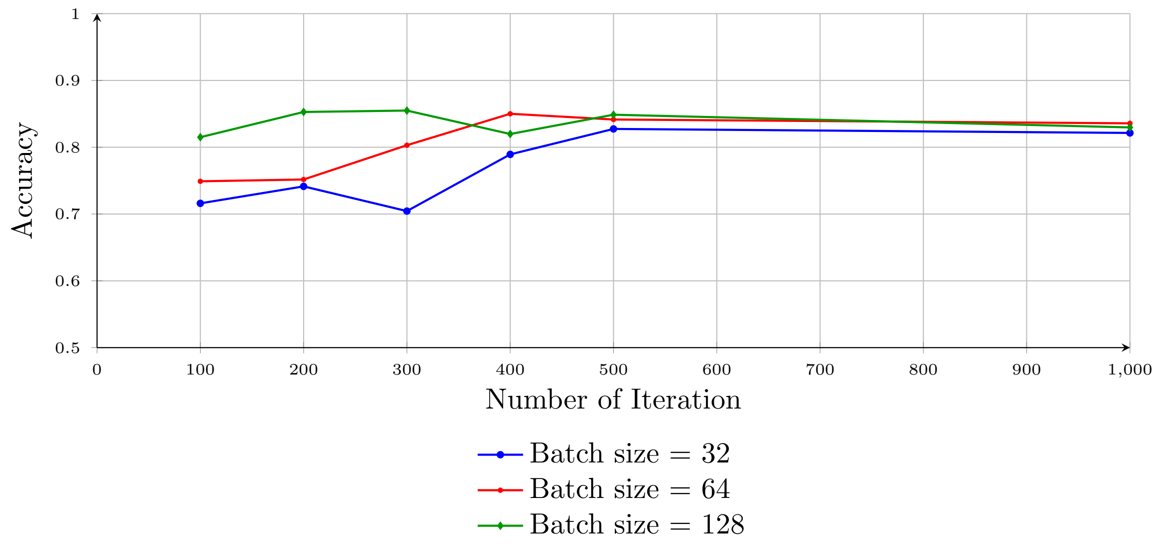


Figure 5.2: The testing accuracy of the trained CNN with GA and batch size values of 32, 64, and 128 on the BreakHis dataset.

5.2 Recall, Precision, and F1-score

As discussed in Chapter 4, the importance of the existing classes in the BreakHis dataset is not equal, since the correct prediction of Malignant tumours is more crucial than Benign ones. Thus, in addition to measuring the classification accuracy, we need to evaluate our classifiers using recall and precision metrics, as well. Moreover, we should calculate the F1-score of the model using the weighted average of recall and precision. In this way, we can consider both false-positive and false-negative cases to examine our model. To do so, the cases providing the highest classification accuracy for Adam and GA optimization techniques are considered.

Table 5.3: Comparison of Adam and GA techniques in terms of classification accuracy, recall, precision, and F1-score on the BreakHis dataset.

Optimizer	Accuracy	Recall	Precision	F1-score
Gradient descent	0.6988	0.5561	0.8437	0.6702
Adam	0.8583	0.7231	0.9623	0.8256
GA	0.8549	0.6943	0.9471	0.8011

As Table 5.3 represents, the model using Adam and GA on the BreakHis dataset obtained the highest accuracy compared to gradient descent as we already discussed in the last section. Both optimizers also produce close values for recall, precision, and F1-score.

Both Adam and GA achieved a high precision rate: the precision for Adam and GA are 0.9623 and 0.9471, respectively. This implies that the number of false-positive predictions made by our model is significantly low (less than 0.1). Whereas, the recall rate produced by either Adam or GA is not as good as precision. This difference indicates that among wrong predictions, most of them are false-negative ones, which is not our preference. In other words, the model classifies the images with a good accuracy and F1-score. However, among wrong predictions, there are more false-negative than false-positive cases, while we expected a better balance between them as discussed in Chapter 4. To overcome this shortcoming, there are different ways; one possible solution can be decreasing the probability threshold in loss function aimed at predicting the positive cancer cases more often. This will be considered in future work.

Moreover, as shown in Table 5.3, the F1-score rates of all three classifiers are slightly less than their classification accuracy. However, for Adam and GA, this rate is in an acceptable range (more than 0.80). Based on what was discussed in Chapter 4, the best F1-score will be provided by a perfect recall and precision at a value of 1. In our case, since we have a lower recall and a higher precision rate, the achieved F1-score by Adam and GA is 0.8256 and 0.8011, respectively. Overall, the results reveal that both Adam and GA achieved acceptable accuracy and F1-score rates which make them capable for classification tasks.

5.3 Execution Time

In clinical scenario, it is important to produce real time fast results. In this section, we therefore, discuss the execution time of the proposed classifier. We implement the models in a sequential setting on a single processor. We experiment and discuss the total required time to complete a single classification task on a single processor. Table 5.4 and its corresponding bar chart (Figure 5.3) summarize the execution time for the three different classifiers: gradient descent, Adam, and GA learning approaches, with a batch size of 128.

Table 5.4: Total execution time (in second) spent by gradient descent, Adam, and GA to complete a classification task by batch size of 128.

	Execution Time (second)		
Iteration	Gradient descent	Adam	GA
100	443.91	360.72	787.28
200	797.00	645.24	1543.12
300	1157.11	936.78	2253.84
400	1560.95	1231.32	3091.63
500	1902.80	1569.79	3979.22
1000	3670.12	3032.84	7543.49

As presented in Figure 5.3, in all three classification models, increasing the number of iterations, increases the execution time. More iteration implies more computation time to train the system using the images. Therefore, it is reasonable to expect increase in execution time for more number of iterations. Thus, for all models, there is an increase in execution time for 500 to 1000 iterations which is consistent with the fact that more iterations implies more computations.

Figure 5.3 shows the bar graph for the three optimizers. The gradient descent, although the execution time is lower compared to GA, it does not produce good accuracy. So, the gradient descent approach can be ignored. The comparison is basically between Adam and GA.

We showed that the accuracy of the GA is similar to the Adam optimizer. We

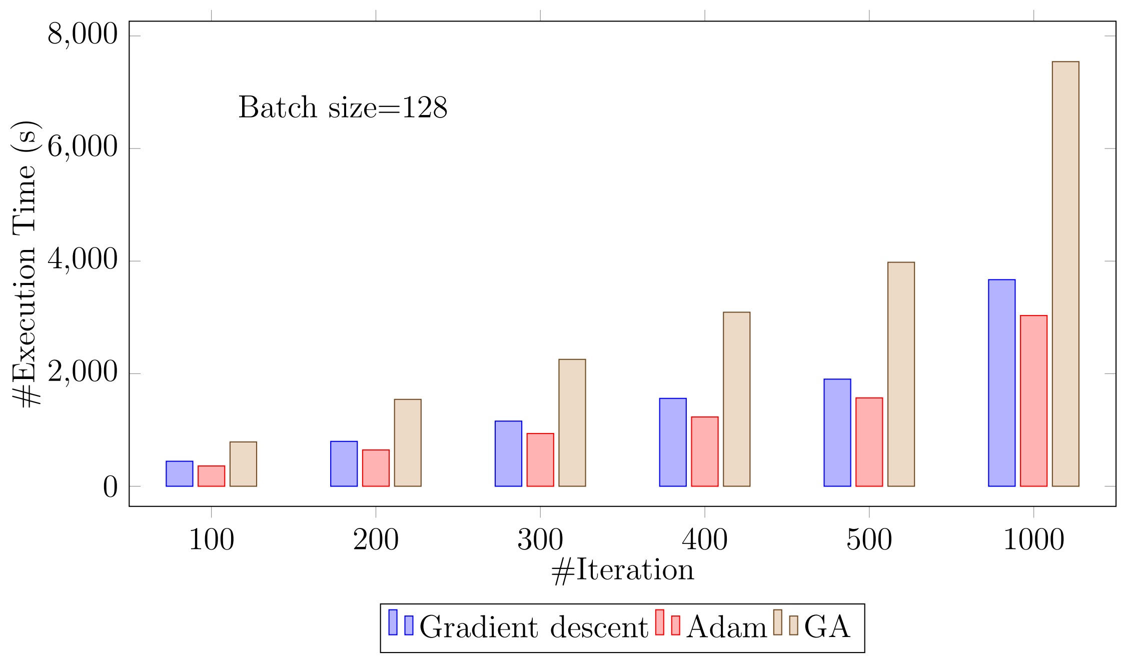


Figure 5.3: Bar chart compares the execution time of gradients descent, Adam, and GA for different number of iteration and a constant batch size of 128.

also indicated that to get the desired accuracy we need to increase the batch size (128 in our case) or increase the number of iterations with respect to the batch size. For example, for batch size 32 (Table 5.2), we get better accuracy with 500 and 1000 iterations. For Adam, with the same batch size, we get better accuracy with 300 iterations and stays almost constant for larger iterations. Therefore, we run the models till we reach the desired accuracy, 300 iterations for Adam and 500-1000 iterations for GA.

In the GA literature, increasing the number of iterations produces better results. This is because, this is an exploration algorithm. It explores the solution space with an initial population and randomly generates other fitter solutions over a period of time. In the model, we first randomly generate the initial weights. Since for each population we consider multiple solutions, the total number of parameters increases significantly. As a result, more time is required to complete the breast cancer classification task using GA. This is consistent with the literature on GA [105]. To expedite the GA process, it is important to execute the algorithm on parallel machines [106; 107]. Since this thesis was to demonstrate the feasibility of using GA for training CNN, parallelization was not considered. This is for future work.

GA Parameters

We also conducted other experiments to understand GA, a bit better. In particular the impact of the different GA parameters on the performance of the proposed classifier. We considered parameters such as the initial population, random number generator methods, and mutation rate. Classification accuracy is used for performance comparison.

Finding the proper size of the initial population is the primary step in running a GA. Small or large population sizes may lead to a poor final solution, while using the proper size increases the chance of finding a more accurate solution [108; 109; 110]. The proper number of starting solutions is a number that warranties enough diversity in the whole population. Therefore, we considered the impact of varying the initial population size on the classification accuracy of histopathological breast images. In this experiment, we considered the results for 300 and 400 iterations.

Table 5.5: The effect of initial population size on classification accuracy of GA. Red color shows classification accuracy more than 0.80.

		Classification Accuracy		
Batch	Iteration	Population size=20	Population size=40	Population size=60
32	300	0.6956	0.7044	0.8219
32	400	0.7632	0.7893	0.8097
64	300	0.8174	0.8030	0.7861
64	400	0.7021	0.8501	0.8480
128	300	0.7649	0.8549	0.7221
128	400	0.8073	0.8198	0.8524

Table 5.5 indicates that the final results achieved by the initial population of size 20 are not as good as either starting population size of 40 or 60. As mentioned above, the reason for this weakness could be due to the less diversity and search space of the initial solutions. In this work, we considered an initial population of size 40 to start searching for the best CNN weights, since it provides good accuracy and needs lower computational resources and execution time rather than population size of 60 or higher.

Table 5.6: Testing the the impact of the random number generator methods on the classification accuracy of GA.

Batch size	Iteration	Accuracy of Normal distribution	Accuracy of Uniform distribution
32	100	0.7908	0.7160
32	200	0.7153	0.7414
32	300	0.6389	0.7044
32	400	0.8364	0.7893
32	500	0.7297	0.8274
32	1000	0.7421	0.8214
64	100	0.8198	0.7490
64	200	0.8370	0.7517
64	300	0.7875	0.8030
64	400	0.8047	0.8501
64	500	0.7213	0.8415
64	1000	0.7956	0.8358
128	100	0.8377	0.8150
128	200	0.8157	0.8528
128	300	0.7972	0.8549
128	400	0.8238	0.8198
128	500	0.7810	0.8487
128	1000	0.8156	0.8297

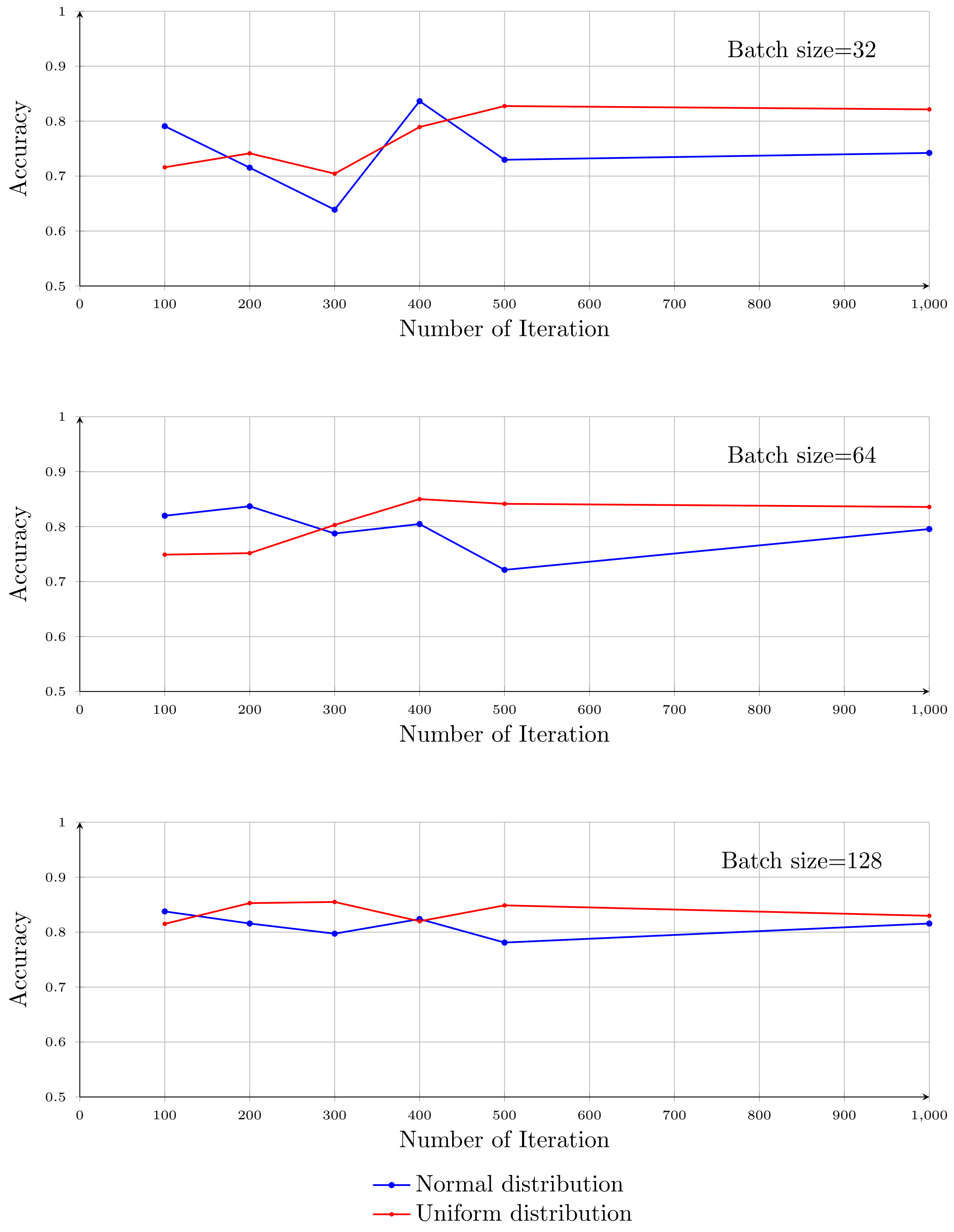


Figure 5.4: Testing the classification accuracy of GA optimizer using normal and uniform distribution on batch size values of 32, 64, and 128.

GA takes an initial population of possible solutions and evolves them slowly until finding the best solution. Through this process, random numbers are used broadly in GA to generate the initial population. Thus, the random number generator methods may influence the quality of the final results [111]. In this work, we investigate the impact of normal and uniform distribution to find the best set of the initial network weights by GA. Table 5.6 and its corresponding graph (Figure 5.4) represent how classification accuracy changes by using normal and uniform distribution.

In general, the classification accuracy achieved by uniform distribution, to generate the initial weights of the CNN model, is slightly better than normal distribution. This result coincides with what is dictated by Maaranen et al. [112] about the random generation of the initial population in GA. As discussed in this study, uniform distribution contains diverse points which makes it a useful technique to generate the initial population of GA, when there is no prior knowledge about the final result.

Finally, we investigated the impact of the mutation operation on the classification accuracy of the proposed GA-based CNN model. Like other GA parameters, mutation probability [113] has an influence on the quality of the final results. Lynch et al. [113] in their article discuss the influence of mutation on population diversity, on improving replication and also the adverse effect of mutations in the population. This mutation rate determines the number of solutions that need to be mutated in each generation to generate new offsprings. Mutation are applied in applications to help GA to avoid trapping in a local minimum. If we ignore the mutation operation after applying crossover, the chances of getting stuck in the local minimum increases since diversity is not considered. The mutation function solves this problem by producing offsprings different from good parents and encouraging the diversity of the solutions [97].

While a very high mutation probability may prevent the convergence of the population to a good solution, too small probability can lead to premature convergence, since just good offsprings are produced by the good part of the parents after applying crossover. Our model is tested using a mutation rate of 0.1 and 0.01 as shown in Table 5.7. As we have the highest accuracy (around 0.85) when the mutation rate is 0.1, we considered this mutation probability for all other experiments.

The next chapter concludes with a discussion and conclusion of our work.

Table 5.7: Testing the impact of mutation probability on the classification accuracy of GA.

		Classification Accuracy	
Batch	Iteration	Mutation rate=0.01	Mutation rate=0.1
32	100	0.6984	0.7160
32	200	0.6320	0.7414
32	300	0.7695	0.7044
32	400	0.7422	0.7893
32	500	0.8138	0.8274
32	1000	0.8091	0.8214
64	100	0.7188	0.7490
64	200	0.7654	0.7517
64	300	0.8352	0.8030
64	400	0.8206	0.8501
64	500	0.7893	0.8415
64	1000	0.8236	0.8358
128	100	0.8474	0.8150
128	200	0.8129	0.8528
128	300	0.8246	0.8549
128	400	0.8095	0.8198
128	500	0.8307	0.8487
128	1000	0.7823	0.8297

Chapter 6

Discussion, Conclusion, and Future Work

6.1 Discussion and Conclusion

Breast cancer is the most regularly diagnosed cancer, and the main cause of cancer mortality in women around the world [1]. Computer-Aided Diagnosis (CAD) systems have become essential and crucial in breast cancer classification problem. Machine learning techniques, such as Convolution Neural Networks (CNN), because of their classification capabilities, have been widely adopted in CAD systems. The precision of extracted features in CNN is highly dependent on the weights of the network, including the weights of all convolution filters, as well as the weights of connecting edges in the fully-connected layer. As a result, these weights play a crucial role in classification accuracy. In this thesis, we proposed to optimize the weights of the CNN using Genetic Algorithm (GA), for histopathological breast image classification problem. GA, a well-known global optimization technique, has been widely used for many real-world applications. However, there is very little work in combining nature-inspired computing with machine learning. This thesis attempted to do this. We provide some insight into the feasibility of hybridizing CNN with GA for breast cancer classification problem.

The proposed algorithm consisted of five layers (Figure 3.7): input layer, convolution layer, pooling layer, activation layer, fully-connected layer and training process. The classification accuracy highly depends on the weights of these layers. The weights are evolved using selection, crossover and mutation operators. We compared our proposed method with mini-batch gradient descent approach and Adam optimizer using the BreakHis dataset. We performed various experiments on different batch sizes and number of iterations to study metrics such as accuracy, recall, precision, execution time and the effect of genetic parameters on the proposed algorithm.

Our experimental results indicated that among the three classifiers, the mini-batch descent classifier provided lower accuracy. We showed that the proposed classifier is as good as the Adam optimizer but with higher batch size and iterations. In general, we observed that the classification accuracy improves for Adam and our proposed classifier for larger batch sizes. This observation coincides the observations by Radiuk et al. [103]. Larger batch sizes improve the training process by using more images to update the weights and also eliminate the model from getting trapped in local optima.

Our results show that the highest test classification accuracy provided by our model is less than what is reported by some other of the state-of-the-art studies.

Although we could improve the test accuracy by making our model deeper with increasing the number of parameters, but to avoid over-fitting we did not. Over-fitting refers to a condition in which there is a significant gap between train and test accuracy. In the current model, the maximum achieved training accuracy is around 90% while the highest testing accuracy is around 85% , indicating a good agreement between test and training accuracy.

We also noted that both Adam and GA achieved higher precision rate. This implies that the number of false-positive predictions made by our model is significantly low (less than 0.1). However, the recall metric result is not satisfactory for both optimizer. The F1-score is an acceptable rate.

The total execution time for GA is higher than the other two optimizer. This is because, the accuracy of GA depends on larger batch size and higher number of iterations. Since GA is an evolutionary process, the stability and accuracy is dictated by the selection, crossover and mutation process over a period of time.

In line with the literature [113], the population size affected the accuracy of the GA. Larger population size produced better results. The classification accuracy achieved by uniform distribution, to generate the initial weights was slightly better than normal distribution.

In conclusion, the proposed GA classifier is a viable method in evolving the weights of the network. Combining the strengths of machine learning and evolutionary algorithms (evolutionary machine learning) is a promising research to pursue for real-world applications.

6.2 Future Work

This thesis studied the feasibility of hybridizing evolutionary algorithm with machine learning models. Here are some ideas on future work:

- Incorporating Diversity: Currently, we have one single population on a single "island". The crossover and mutation are performed on this island. Diversity is important in GA for better accuracy. However, in the GA literature, there is a model called the island GA model. Island GA is a multi-population technique in which chromosomes migrate between existing sub-populations (or islands) to increase the diversity. This technique may prevent early convergence by increasing the population diversity to achieve higher classification accuracy. We can decrease the size of the population on each island. Various island topologies provide different cross fertilization. We believe this approach can accommodate smaller batch sizes as the Adam optimizer.

- Improving computation time: Although, we could parallelize the proposed GA (and there is lots of work on parallel GA), the island model would be better suited for parallelization. Each island could be implemented on a different processor providing lots of concurrency.

- GA parameters: The parameters impact the application. There are other strategies for selection and crossover that we could try on the proposed classifier.

- Collaborative approach: Rather training the weights, we can use Adam and GA in a collaborative setting. That is, first use the Adam optimizer to train the model. Then, the best set of the network weights found by Adam can be passed to GA as its initial population. GA will then evolve the solutions.

Bibliography

[1] A. Abraham, "Artificial neural networks," *Handbook of measuring system design*, 2005.

[2] F. Ahmad, N. A. M. Isa, M. K. Osman, and Z. Hussain, "Performance comparison of gradient descent and genetic algorithm based artificial neural networks training," in *2010 10th International Conference on Intelligent Systems Design and Applications.* IEEE, 2010, pp. 604–609.

[3] F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre, and A. Jemal, "Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: a cancer journal for clinicians*, vol. 68, no. 6, pp. 394–424, 2018.

[4] T. Araújo, G. Aresta, E. Castro, J. Rouco, P. Aguiar, C. Eloy, A. Polónia, and A. Campilho, "Classification of breast cancer histology images using convolutional neural networks," *PloS one*, vol. 12, no. 6, p. e0177544, 2017.

[5] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte, "Breast cancer histopathological image classification using convolutional neural networks," in

2016 international joint conference on neural networks (IJCNN). IEEE, 2016, pp. 2560–2567.

[6] C. Zhu, F. Song, Y. Wang, H. Dong, Y. Guo, and J. Liu, "Breast cancer histopathology image classification through assembling multiple compact cnns," *BMC medical informatics and decision making*, vol. 19, no. 1, p. 198, 2019.

[7] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and H. Laurent, "A dataset for breast cancer histopathological image classification," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 7, pp. 1455–1462, 2015.

[8] J. Xie, R. Liu, J. Luttrell IV, and C. Zhang, "Deep learning based analysis of histopathological images of breast cancer," *Frontiers in genetics*, vol. 10, p. 80, 2019.

[9] J. G. Elmore, G. M. Longton, P. A. Carney, B. M. Geller, T. Onega, A. N. Tosteson, H. D. Nelson, M. S. Pepe, K. H. Allison, S. J. Schnitt *et al.*, "Diagnostic concordance among pathologists interpreting breast biopsy specimens," *Jama*, vol. 313, no. 11, pp. 1122–1132, 2015.

[10] M. Nawaz, A. A. Sewissy, and T. H. A. Soliman, "Multi-class breast cancer classification using deep learning convolutional neural network," *Int. J. Adv. Comput. Sci. Appl*, vol. 9, no. 6, pp. 316–332, 2018.

[11] B. Halalli and A. Makandar, "Computer aided diagnosis-medical image analysis techniques," *Breast Imaging*, 2017.

[12] M. Li and Z.-H. Zhou, “Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 6, pp. 1088–1098, 2007.

[13] A. Jalalian, S. B. Mashohor, H. R. Mahmud, M. I. B. Saripan, A. R. B. Ramli, and B. Karasfi, “Computer-aided detection/diagnosis of breast cancer in mammography and ultrasound: a review,” *Clinical imaging*, vol. 37, no. 3, pp. 420–426, 2013.

[14] N. I. Yassin, S. Omran, E. M. El Houby, and H. Allam, “Machine learning techniques for breast cancer computer aided diagnosis using different image modalities: A systematic review,” *Computer methods and programs in biomedicine*, vol. 156, no. 3, pp. 25–45, 2018.

[15] S. A. Medjahed, T. A. Saadi, and A. Benyettou, “Breast cancer diagnosis by using k-nearest neighbor with different distances and classification rules,” *International Journal of Computer Applications*, vol. 62, no. 1, 2013.

[16] M. F. Akay, “Support vector machines combined with feature selection for breast cancer diagnosis,” *Expert systems with applications*, vol. 36, no. 2, pp. 3240–3247, 2009.

[17] Y. Wu, M. L. Giger, K. Doi, C. J. Vyborny, R. A. Schmidt, and C. E. Metz, “Artificial neural networks in mammography: application to decision making in the diagnosis of breast cancer.” *Radiology*, vol. 187, no. 1, pp. 81–87, 1993.

[18] V. Chaurasia and S. Pal, "A novel approach for breast cancer detection using data mining techniques," *International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol*, vol. 2, 2017.

[19] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[20] Z. Han, B. Wei, Y. Zheng, Y. Yin, K. Li, and S. Li, "Breast cancer multi-classification from histopathological images with structured deep learning model," *Scientific reports*, vol. 7, no. 1, pp. 1–10, 2017.

[21] M. I. Razzak, S. Naz, and A. Zaib, "Deep learning for medical image processing: Overview, challenges and the future," in *Classification in BioApps.* Springer, 2018, pp. 323–350.

[22] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*, 2016, pp. 173–182.

[23] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing [exploratory dsp]," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 145–154, 2010.

[24] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Mul-

timodal deep learning for robust rgb-d object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 681–687.

[25] A. A. Mohamed, W. A. Berg, H. Peng, Y. Luo, R. C. Jankowitz, and S. Wu, "A deep learning method for classifying mammographic breast density categories," *Medical physics*, vol. 45, no. 1, pp. 314–321, 2018.

[26] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck, "Deep learning for identifying metastatic breast cancer," *arXiv preprint arXiv:1606.05718*, 2016.

[27] J. Li, J.-h. Cheng, J.-y. Shi, and F. Huang, "Brief introduction of back propagation (bp) neural network algorithm and its improvement," in *Advances in computer science and information engineering*. Springer, 2012, pp. 553–558.

[28] D. J. Montana and L. Davis, "Training feedforward neural networks using genetic algorithms." in *IJCAI*, vol. 89, 1989, pp. 762–767.

[29] S. Belciug and F. Gorunescu, "A hybrid neural network/genetic algorithm applied to breast cancer detection and recurrence," *Expert Systems*, vol. 30, no. 3, pp. 243–254, 2013.

[30] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[31] A. F. Gad, Gad, and S. John, *Practical Computer Vision Applications Using Deep Learning with CNNs*. Springer, 2018.

[32] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Evolving deep convolutional neural

networks for image classification," *IEEE Transactions on Evolutionary Computation*, 2019.

[33] L. Xie and A. Yuille, "Genetic cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1379–1388.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[35] S. Deepa and B. A. Devi, "A survey on artificial intelligence approaches for medical image classification," *Indian Journal of Science and Technology*, vol. 4, no. 11, pp. 1583–1595, 2011.

[36] M. Amrane, S. Oukid, I. Gagaoua, and T. Ensarİ, "Breast cancer classification using machine learning," in *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT).* IEEE, 2018, pp. 1–4.

[37] A.-A. Nahid and Y. Kong, "Involvement of machine learning for breast cancer image classification: a survey," *Computational and mathematical methods in medicine*, vol. 2017, 2017.

[38] T. C. Cahoon, M. A. Sutton, and J. C. Bezdek, "Breast cancer detection using image processing techniques," in *Ninth IEEE International Conference on Fuzzy Systems. FUZZ-IEEE 2000 (Cat. No. 00CH37063)*, vol. 2. IEEE, 2000, pp. 973–976.

[39] N. Singh, A. G. Mohapatra, and G. Kanungo, "Breast cancer mass detection

in mammograms using k-means and fuzzy c-means clustering," *International Journal of Computer Applications*, vol. 22, no. 2, pp. 15–21, 2011.

[40] A. E. Lashkari and M. Firouzmand, "Early breast cancer detection in thermogram images using adaboost classifier and fuzzy c-means clustering algorithm," 2016.

[41] M. Kowal, P. Filipczuk, A. Obuchowicz, J. Korbicz, and R. Monczak, "Computer-aided diagnosis of breast cancer based on fine needle biopsy microscopic images," *Computers in biology and medicine*, vol. 43, no. 10, pp. 1563–1572, 2013.

[42] A. K. Dubey, U. Gupta, and S. Jain, "Comparative study of k-means and fuzzy c-means algorithms on the breast cancer data," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 8, no. 1, pp. 18–29, 2018.

[43] C. Wring and X. Liu, "Face detection using spectral histograms and svms," *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS*, vol. 35, no. 3, June 2005.

[44] C. Bahlmann, B. Haasdonk, and H. Burkhardt, "Online handwriting recognition with support vector machines - a kernel approach," in *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*, Niagara on the Lake, Ontario, Canada, Canada, August 2002.

[45] B. E and S. G., “Support vector machine applications in bioinformatics,” *Applied Bioinformatics*, vol. 2, no. 2, pp. 67–77, 2003.

[46] H. Frohlich and O. Chapelle, “Feature selection for support vector machines by means of genetic algorithms,” in *IEEE international conference on tools with artificial intelligence*, Sacramento, CA, 2003, pp. 142–148.

[47] D. Dua and C. Graff, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[48] J. Ding, H.-D. Cheng, J. Huang, J. Liu, and Y. Zhang, “Breast ultrasound image classification based on multiple-instance learning,” *Journal of digital imaging*, vol. 25, no. 5, pp. 620–627, 2012.

[49] F. Shirazi and E. Rashedi, “Detection of cancer tumors in mammography images using support vector machine and mixed gravitational search algorithm,” in *2016 1st conference on swarm intelligence and evolutionary computation (CSIEC)*. IEEE, 2016, pp. 98–101.

[50] M. Taheri, G. Hamer, S. H. Son, and S. Y. Shin, “Enhanced breast cancer classification with automatic thresholding using svm and harris corner detection,” in *Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, 2016, pp. 56–60.

[51] B. Zheng, S. W. Yoon, and S. S. Lam, “Breast cancer diagnosis based on feature extraction using a hybrid of k-means and support vector machine algorithms,” *Expert Systems with Applications*, vol. 41, no. 4, pp. 1476–1482, 2014.

[52] G. Verma and H. Verma, “Predicting breast cancer using linear kernel support vector machine,” *Available at SSRN 3350254*, 2019.

[53] H. Wang, B. Zheng, S. W. Yoon, and H. S. Ko, “A support vector machine-based ensemble algorithm for breast cancer diagnosis,” *European Journal of Operational Research*, vol. 267, no. 2, pp. 687–699, 2018.

[54] A. Tsymbal and S. Puuronen, “Bagging and boosting with dynamic integration of classifiers,” in *European Conference on Principles of Data Mining and Knowledge Discovery.* Springer, 2000, pp. 116–125.

[55] M. E. Mavroforakis, H. V. Georgiou, N. Dimitropoulos, D. Cavouras, and S. Theodoridis, “Mammographic masses characterization based on localized texture and dataset fractal analysis using linear, neural and support vector machine classifiers,” *Artificial intelligence in medicine*, vol. 37, no. 2, pp. 145–162, 2006.

[56] E. Fischer, J. Lo, and M. Markey, “Bayesian networks of bi-rads/spl trade/descriptors for breast lesion classification,” in *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 2. IEEE, 2004, pp. 3031–3034.

[57] D. Soria, J. M. Garibaldi, E. Biganzoli, and I. O. Ellis, “A comparison of three different methods for classification of breast cancer data,” in *2008 Seventh International Conference on Machine Learning and Applications.* IEEE, 2008, pp. 619–624.

[58] V. Rodríguez-López and R. Cruz-Barbosa, "Improving bayesian networks breast mass diagnosis by using clinical data," in *Mexican Conference on Pattern Recognition.* Springer, 2015, pp. 292–301.

[59] K. Rajakeerthana, C. Velayutham, and K. Thangavel, "Mammogram image classification using rough neural network," in *Computational Intelligence, Cyber Security and Computational Models.* Springer, 2014, pp. 133–138.

[60] K. B. Nahato, K. N. Harichandran, and K. Arputharaj, "Knowledge mining from clinical datasets using rough sets and backpropagation neural network," *Computational and mathematical methods in medicine*, vol. 2015, 2015.

[61] A. Bhattacherjee, S. Roy, S. Paul, P. Roy, N. Kausar, and N. Dey, "Classification approach for breast cancer detection using back propagation neural network: a study," in *Biomedical image analysis and mining techniques for improved health outcomes.* IGI Global, 2016, pp. 210–221.

[62] E. E. E. Ali and W. Z. Feng, "Breast cancer classification using support vector machine and neural network," *International Journal of Science and Research*, vol. 5, no. 3, pp. 1–6, 2016.

[63] S. Kaymak, A. Helwan, and D. Uzun, "Breast cancer image classification using artificial neural networks," *Procedia computer science*, vol. 120, pp. 126–131, 2017.

[64] M. A. Mohammed, B. Al-Khateeb, A. N. Rashid, D. A. Ibrahim, M. K. A. Ghani, and S. A. Mostafa, "Neural network and multi-fractal dimension features

for breast cancer classification from ultrasound images," *Computers & Electrical Engineering*, vol. 70, pp. 871–882, 2018.

[65] M. M. Saritas and A. Yasar, "Performance analysis of ann and naive bayes classification algorithm for data classification," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 7, no. 2, pp. 88–91, 2019.

[66] M. M. Abdelsamea, M. H. Mohamed, and M. Bamatraf, "Automated classification of malignant and benign breast cancer lesions using neural networks on digitized mammograms," *Cancer informatics*, vol. 18, p. 1176935119857570, 2019.

[67] R. Fakoor, F. Ladhak, A. Nazi, and M. Huber, "Using deep learning to enhance cancer diagnosis and classification," in *Proceedings of the international conference on machine learning*, vol. 28. ACM New York, USA, 2013.

[68] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis detection in breast cancer histology images with deep neural networks," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2013, pp. 411–418.

[69] A. M. Abdel-Zaher and A. M. Eldeib, "Breast cancer classification using deep belief networks," *Expert Systems with Applications*, vol. 46, pp. 139–144, 2016.

[70] N. Bayramoglu, J. Kannala, and J. Heikkilä, "Deep learning for magnification independent breast cancer histopathology image classification," in *2016 23rd*

International conference on pattern recognition (ICPR). IEEE, 2016, pp. 2440–2445.

[71] D. Bardou, K. Zhang, and S. M. Ahmad, "Classification of breast cancer based on histology images using convolutional neural networks," *IEEE Access*, vol. 6, pp. 24 680–24 693, 2018.

[72] S. H. Kassani, P. H. Kassani, M. J. Wesolowski, K. A. Schneider, and R. Deters, "Classification of histopathological biopsy images using ensemble of deep learning networks," *arXiv preprint arXiv:1909.11870*, 2019.

[73] M. Veta, J. P. Pluim, P. J. Van Diest, and M. A. Viergever, "Breast cancer histopathology image analysis: A review," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 5, pp. 1400–1411, 2014.

[74] A. Rakhlin, A. Shvets, V. Iglovikov, and A. A. Kalinin, "Deep convolutional neural networks for breast cancer histology image analysis," in *International Conference Image Analysis and Recognition.* Springer, 2018, pp. 737–744.

[75] Y. Zheng, Z. Jiang, F. Xie, H. Zhang, Y. Ma, H. Shi, and Y. Zhao, "Feature extraction from histopathological images based on nucleus-guided convolutional neural network for breast lesion classification," *Pattern Recognition*, vol. 71, pp. 14–25, 2017.

[76] A.-A. Nahid, M. A. Mehrabi, and Y. Kong, "Histopathological breast cancer image classification by deep neural network techniques guided by local clustering," *BioMed research international*, vol. 2018, 2018.

[77] M. Z. Alom, C. Yakopcic, M. S. Nasrin, T. M. Taha, and V. K. Asari, “Breast cancer classification from histopathological images with inception recurrent residual convolutional neural network,” *Journal of digital imaging*, vol. 32, no. 4, pp. 605–617, 2019.

[78] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley, and J. Clune, “Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning,” *arXiv preprint arXiv:1712.06567*, 2017.

[79] A. Bhardwaj and A. Tiwari, “Breast cancer diagnosis using genetically optimized neural network model,” *Expert Systems with Applications*, vol. 42, no. 10, pp. 4611–4620, 2015.

[80] R. S. Sexton and R. E. Dorsey, “Reliable classification using neural networks: a genetic algorithm and backpropagation comparison,” *Decision Support Systems*, vol. 30, no. 1, pp. 11–22, 2000.

[81] G. Das, P. K. Pattnaik, and S. K. Padhy, “Artificial neural network trained by particle swarm optimization for non-linear channel equalization,” *Expert Systems with Applications*, vol. 41, no. 7, pp. 3491–3496, 2014.

[82] I. Aljarah, H. Faris, and S. Mirjalili, “Optimizing connection weights in neural networks using the whale optimization algorithm,” *Soft Computing*, vol. 22, no. 1, pp. 1–15, 2018.

[83] Q. Cao and M. E. Parry, “Neural network earnings per share forecasting models:

A comparison of backward propagation and the genetic algorithm," *Decision Support Systems*, vol. 47, no. 1, pp. 32–41, 2009.

[84] J. N. Gupta and R. S. Sexton, "Comparing backpropagation with a genetic algorithm for neural network training," *Omega*, vol. 27, no. 6, pp. 679–684, 1999.

[85] A. Ghaffari, H. Abdollahi, M. Khoshayand, I. S. Bozchalooi, A. Dadgar, and M. Rafiee-Tehrani, "Performance comparison of neural network training algorithms in modeling of bimodal drug delivery," *International journal of pharmaceutics*, vol. 327, no. 1-2, pp. 126–138, 2006.

[86] E. Alba and J. F. Chicano, "Training neural networks with ga hybrid algorithms," in *Genetic and Evolutionary Computation Conference.* Springer, 2004, pp. 852–863.

[87] Z. Liu, A. Liu, C. Wang, and Z. Niu, "Evolving neural network using real coded genetic algorithm (ga) for multispectral image classification," *Future Generation Computer Systems*, vol. 20, no. 7, pp. 1119–1129, 2004.

[88] S. R. Young, D. C. Rose, T. P. Karnowski, S.-H. Lim, and R. M. Patton, "Optimizing deep learning hyper-parameters through an evolutionary algorithm," in *Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments*, 2015, pp. 1–5.

[89] E. P. Ijjina and K. M. Chalavadi, "Human action recognition using genetic

algorithms and convolutional neural networks," *Pattern recognition*, vol. 59, pp. 199–212, 2016.

[90] A. Martín, R. Lara-Cabrera, F. Fuentes-Hurtado, V. Naranjo, and D. Camacho, "Evodeep: a new evolutionary approach for automatic deep neural networks parametrisation," *Journal of Parallel and Distributed Computing*, vol. 117, pp. 180–191, 2018.

[91] H. Su, F. Liu, Y. Xie, F. Xing, S. Meyyappan, and L. Yang, "Region segmentation in histopathological breast cancer images using deep convolutional neural network," in *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI).* IEEE, 2015, pp. 55–58.

[92] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[93] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural networks: Tricks of the trade.* Springer, 2012, pp. 437–478.

[94] J. Zhang and F. B. Gouza, "Gadam: genetic-evolutionary adam for deep neural network optimization," *arXiv preprint arXiv:1805.07500*, 2018.

[95] D. Whitley, "A genetic algorithm tutorial," *Statistics and computing*, vol. 4, no. 2, pp. 65–85, 1994.

[96] K. Deb, R. B. Agrawal *et al.*, "Simulated binary crossover for continuous search space," *Complex systems*, vol. 9, no. 2, pp. 115–148, 1995.

[97] I. Korejo, S. Yang, K. Brohi, and Z. Khuhro, "Multi-population methods with adaptive mutation for multi-modal optimization problems," 2013.

[98] N. Soni and T. Kumar, "Study of various mutation operators in genetic algorithms," *(IJCSIT) International Journal of Computer Science and Information Technologies*, vol. 5, no. 3, pp. 115–148, 2014.

[99] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information processing & management*, vol. 45, no. 4, pp. 427–437, 2009.

[100] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.

[101] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 464–472.

[102] Y. Wu, L. Liu, J. Bae, K.-H. Chow, A. Iyengar, C. Pu, W. Wei, L. Yu, and Q. Zhang, "Demystifying learning rate polices for high accuracy training of deep neural networks," *arXiv preprint arXiv:1908.06477*, 2019.

[103] P. M. Radiuk, "Impact of training set batch size on the performance of convolutional neural networks for diverse datasets," *Information Technology and Management Science*, vol. 20, no. 1, pp. 20–24, 2017.

[104] M. Angelova and T. Pencheva, "Tuning genetic algorithm parameters to im-

prove convergence time," *International Journal of Chemical Engineering*, vol. 2011, 2011.

[105] S. G. B. Rylander and B. Gotshall, "Optimal population size and the genetic algorithm," *Population*, vol. 100, no. 400, p. 900, 2002.

[106] D. Whitley, S. Rana, and R. B. Heckendorn, "The island model genetic algorithm: On separability, population size and convergence," *Journal of computing and information technology*, vol. 7, no. 1, pp. 33–47, 1999.

[107] P. Pospichal, J. Jaros, and J. Schwarz, "Parallel genetic algorithm on the cuda architecture," in *European conference on the applications of evolutionary computation*. Springer, 2010, pp. 442–451.

[108] E. K. Burke, S. Gustafson, and G. Kendall, "Diversity in genetic programming: An analysis of measures and correlation with fitness," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 1, pp. 47–62, 2004.

[109] P. A. Diaz-Gomez and D. F. Hougen, "Initial population for genetic algorithms: A metric approach." in *Gem*, 2007, pp. 43–49.

[110] R. H. Herring III and M. R. Eden, "Evolutionary algorithm for de novo molecular design with multi-dimensional constraints," *Computers & Chemical Engineering*, vol. 83, pp. 267–277, 2015.

[111] M. Olteanu and N. Paraschiv, "The influence of random numbers generators upon genetic algorithms," in *2013 IEEE INISTA*. IEEE, 2013, pp. 1–5.

[112] H. Maaranen, K. Miettinen, and M. M. Mäkelä, "Quasi-random initial population for genetic algorithms," *Computers & Mathematics with Applications*, vol. 47, no. 12, pp. 1885–1895, 2004.

[113] M. Lynch, M. S. Ackerman, J.-F. Gout, H. Long, W. Sung, W. K. Thomas, and P. L. Foster, "Genetic drift, selection and the evolution of the mutation rate," *Natural Reviews Genetics*, no. 17, pp. 704–714, 2016.