Smart Cancer Diagnosis Using Machine Learning Techniques

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Shambhawi Pal (151238)

Under the supervision of

Dr. Amit Kumar

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Certificate

Candidate's Declaration

I hereby declare that the work presented in this report titled **"Smart Cancer Diagnosis using Machine Learning Techniques"** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2018 to December 2018 under the supervision of **Dr. Amit Kumar** (Assistant Professor, Department of Computer Science and Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Shambhawi Pal, 151238

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Amit Kumar

Assistant Professor

Department of Computer Science and Engineering and Information Technology

Dated:

Acknowledgement

We wish to express our profound and sincere gratitude to Dr. Amit Kumar, Assistant Professor, Department of Computer Science and Information Technology, Jaypee University of Information Technology, who guided us into the intricacies of this project non-chalantly with matchless magnanimity. He constantly co-operated and helped with the research work. He also evinced keen interest and invaluable support in the field of Machine Learning for progress of our project work.

Table of Contents

LIST OF ABBREVIATIONS

A	Adenosis
ANN	Artificial Neural Networks
BC	Breast Cancer
CLBP	Completed Local Binary Patterns
CNN	Convolutional Neural Networks
DC	Ductal Carcinoma
F	Fibroadenoma
GLCM	Gray-Level Co-Occurrence Matrix
KNN	k Nearest Neighbors
LBP	Local Binary Patterns
LC	Lobular Carcinoma
MC	Mucinous Carcinoma
ML	Machine Learning
NN	Neural Networks
ORB	Oriented FAST and Rotated Brief
PC	Papillary Carcinoma
PFTAS	Parameter Free Threshold Adjacency Statistics
PT	Phyllodes Tumor
RF	Random Forests
SVM	Support Vector Machines
TA	Tubular Adenoma

Abstract

Breast cancer (BC) is one of the most common cancers among women worldwide. According to world statistics, these are the majority of new cancers and deaths related to cancer, making them an important public health problem in today's society. Early diagnosis of breast cancer can significantly improve prognosis and survival as it promotes timely medical treatment of patients. Additional unnecessary treatments can be avoided by accurately classifying benign and malignant tumors. Therefore, the correct and correct diagnosis of breast cancer tumors and the classification into benign or malignant categories is an important field of research. Machine learning is emerging as a method of choice in the classification of breast cancer patterns and in the predictive model because of its advantages in detecting features from complex breast cancer data sets. In this project various techniques for extracting properties are used, such as: For example, the local binary pattern, scalar invariant property transformation (SIFT), and oriented FAST and rotated LETTER (ORB), GLCM, PFTAS. Thereafter, the machine learning techniques in breast cancer prognosis will be reviewed. The project provides a general description of the machine learning techniques, e.g. The support vector machine, the neural networks, the random structure and the decision tree as well as the first nearest neighbor. The primary data of the project comes from the breast cancer database BreakHis (BH).

CHAPTER 1

INTRODUCTION

1.1 Introduction

At this time, mechanical education and artificial intelligence have been included in almost all areas of our daily lives. Researchers have also tried to include it in the medical field. Healthcare improves diagnosis with the help of artificial intelligence and engineering algorithms of learning.

Our machine learning algorithm predicts that predictions or such solutions such as stock market fluctuations, weather forecasting, pattern recognition, appropriate solutions for the statement of any problem, help forecast this machine To know the various aspects of the problem statement, to get it, information and data are fed into the machine as a conversation and real world observation. The machine gradually learns from the data provided and predicts the output.

Cancer is a type of disease that has very specific characteristics. Cells become abnormal and indirectly uncontrolled. Cancer spreads to other organs of the human body, which destroys other body tissues. The main reason is that the cancer is very fatal that it can spread throughout the body. Cancer is the second place after the deadly diseases of the world. Several new treatments have been offered for the treatment of cancer. As a result of these progress, the survival rate of cancer patients is increasing. These cancer cells are respectively linked to a tumor. The formation of so many tumors changes the normal tissue in the human body. Then the volume spreads to other parts of the body and is integrated into these parts. It is called metastasis.

In women, cancer is the leading cause of death. Tumor classification is a fundamental process in the diagnosis of cancer. Tumors can be divided into two main types of malignant and benign tumors. Doctors need a reliable diagnostic method to differentiate between these tumors. Even these experts find it difficult to differentiate between these tumors. Therefore, an automatic diagnostic machine is required to diagnose these tumors. Many researchers have used machine learning algorithms to detect the rigors of human cancer and many researchers that these algorithms represent a better way to detect cancer.

Machine Learning

The science of making computers programmable without being explicitly programmed is called machine learning. The construction of algorithms that can receive data and efficiently use statistical analysis to predict production while updating the output as new data appears is the prerequisite of machine learning. The learning process begins with the observation of the data so that the patterns can be found in the data and better decisions can be made in the future based on the provided example. The main goal is to enable computers to learn without human help or interaction and to adapt their actions accordingly.

Machine learning has the following main algorithms:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Supervised learning: This is a set of predictors. These predictors are independent variables. The objective of this learning algorithm is to predict from this set of independent variables. The prediction is for the outcome variable. This is a dependent variable. With the set of independent variables, a function is generated that facilitates the allocation of our inputs to the desired outputs. To achieve a certain precision in our training data, the machine is continuously trained. Examples of supervised learning are linear regression, logistic regression, KNN decision tree, random forest, etc.

Uncontrolled learning: in this algorithm, there is no particular goal or result that can be estimated or predicted. It is used to group into different groups, which is used for segmentation into different groups for specific interventions. Some examples of unsupervised learning are K-Means, Apriori's algorithm.

Reinforcement learning: certain decisions have been made when training the machine with this algorithm. It works so that the machine is exposed to conditions in any environment. The machine is continuously trained with the trial and error method. To make accurate business decisions, the machine learns from past experience by capturing the best possible knowledge. Some examples of learning by reinforcement are the Markov decision process.

1.2 Problem Statement

Cancer is a disease associated with the reproduction and development of cells which are not controlled. In developed, developed and developing countries, approximately 8 million people die each year from cancer. There are many types of cancer, and any kind of treatment involves a deep study of action and behavior in cells. Most of the patients with breast cancer are women. Statistics show that approximately 85% of women have breast cancer.

In the case of initial treatment of patients, the probability of survival has increased significantly with early diagnosis of breast cancer. With proper tumor classification, unnecessary treatment can be avoided. Each volume should be treated differently. Therefore, if there is no proper diagnosis then there is a high risk of death for the patient. Correct diagnosis of breast cancer and classification of tumors in benign and malignant tumors is an area of investigation.

In the last decade, breast cancer cases are three times higher. The proportion of cancer patients in physicians (cancer experts) has increased. The current methods of diagnosing cancer are not fast enough to serve the growing cancer population. Therefore, it is necessary to develop a system that predicts the highest precision cancers and helps us understand all the mechanisms behind breast cancer. It will also introduce new treatments.

As mentioned above, the benefits of identifying important features of mechanical learning, complex data sets, play an important role in classification and forecasting of breast cancer. Since the best results can be achieved with engineering learning algorithms, doctors should use these techniques to diagnose cancer. This is because learning engineering algorithms can provide more accurate results. Apart from this, the results are achieved at a short time and doctors preserve complex manual work. Therefore, an intelligent health system is an important and valuable domain.

1.3 Objectives

Our project aims to diagnose and predict the type of tumor using histopathological images of tissues with a focus on improving the accuracy of prediction. This will reduce the workload on the physician and help him to provide better and fast medical treatment and service to the patient diagnosed with cancer.

Our first dataset consists of images of histopathological cells. Our main objective is to classify the given images into the following categories:

Benign:

- Adenosis
- Fibroadenoma
- Phyllodes Tumor
- Tubular Adenoma

Malignant:

- Lobular
- Papillary
- Ductal
- Mucinous

The above given tumors have to be treated differently. If this is not done, then the patient is at a high risk of dying. Thus our aim is to provide accurate result in order to provide correct diagnosis and treatment to the patient.

The second dataset consists of 4 classes.

- Normal
- Benign
- In Situ
- Invasive

1.4 Methodology

Our dataset consist of histopathological images acquired on 82 patients. The dataset has 7909 microscopic images of tissues. This data has been initially divided into 5 folds such that each fold comprises of Train Data and Test Data. For each Train Data and Test Data images in different magnification forms are present. Theses magnification forms include: 40X, 100X, 200X, 400X. All these images have been labeled according to their class names.

We have a problem statement which involves the use of Supervised Learning and comes under the category of Classification. It is a multiclass classification in which the classes given to us are: Phyllodes Tumor, Adenosis, Tubular Adenoma, Fibroadenoma, Lobular Carcinoma, Ductal Carcinoma, Papillary Carcinoma, Mucinous Carcinoma.

Our aim is to classify the given image into the above given classes. In order to do so, we have to first analyze the images given to us. For analyzing the features, we are using different techniques. These are the feature extraction techniques. These techniques will extract features from the given image and form an "n-d feature vector". The feature vector so obtained could be used to train our data. Since we can obtain feature vector from this, therefore we can have all the required parameters to train our data using any Machine Learning model.

The training of model can be done in many ways. It depends on how the data is prepared for further processing. The data can be used directly depending on the situation or the data can be used to form a histogram. After these modifications, we choose a particular model on which we will train our data. This model can be: Linear regression, Logistic Regression, SVM, Neural Networks, Decision Tress, K-Nearest Neighbors etc. Parameter tuning can also be done in order to increase our accuracy.

Once the model is trained, we can test our data by applying our algorithms on the Test Data. With the help of this we can find the learning ability of our algorithm. We can also use different performance measures such as confusion matrix, accuracy, F score, Recall, Precision etc in order to analyze the obtained result.

CHAPTER 2

LITERATURE SURVEY

2.1 Histopathological Breast Cancer Image Classification [1]

The BreakHis database consists of images of malignant and benign breast tumours. Many clinical studies was conducted in 2014 at P&D Laboratory, Brazil. Images of all referred patients were collected. The study was approved by the institutional review board and a written consent was given by all patients. All the data was anonymous.

Often used in clinical procedures, the standard paraffin process was used for the preparation of theses slides. Preserving the original molecular composition and tissue structure was the main goal of this study.

The overall method involves steps like trimming, dehydration, infiltration, clearing, embedding, and fixation. Sections of 3µm are cut using a microtome and are mounted on slides. The slides are then stained and a glass coverslip is used to cover the sections. Following this the tumoural areas in the slides are identified by the anatomopathologists through visual analysis of the section. Exams like immunohistochemistry analysis are used to confirm the final result of each case which is generated by good pathologists. Automatic exposure is set for the camera and manual focusing is done. Black borders are contained in the original images, in order to remove the errors, the final images are changed and ultimately stored.

The database comprises of 7909 images which is divided into different categories namely: benign and malignant tumours. The image distribution is shown in table(2.1).The dataset currently contains Phyllodes Tumor, Adenosis, Tubular Adenoma, Fibroadenoma, Lobular Carcinoma, Ductal Carcinoma, Papillary Carcinoma, Mucinous Carcinoma. Table (2.2) and table(2.3) show the distribution of tumours in these classes.

Table 2.1 Images distribution in BreaKHis Dataset

Magnification	Benign	Malignant	Total
40x	625	1370	1995
100x	644	1437	2081
200x	623	1390	2013
400x	588	1232	1820
Total	2840	5429	7909
#Patients	24	58	82

Table 2.2 Image distribution of Benign Tumor

Magnifications	A	F	TA	PT	Total
40x	113	252	108	148	594
100x	113	260	121	150	614
200x	111	264	108	140	594
400x	106	237	115	130	562
Total	443	1013	452	568	2363
No of Patients	5	11	5	6	27

Table 2.3 Image distribution of Malignant Tumor

Magnifications	DC	LC	MC	PC	Total
40x	864	156	205	145	1370
100x	903	170	222	142	1437
200x	896	163	196	135	1390
400x	788	137	169	138	1232
Total	3451	626	792	560	5429
#Patients	38	5	9	6	58

Feature Extractors and Classifiers

These include local binary patterns (LBP), local phase quantification (LPQ), closed LBP (CLBP), threshold neighborhood statistics (TAS), gray level co-occurrence matrix (GLCM) and a key point descriptor known as ORB. Key point descriptors are an integral part of object recognition. In addition, these descriptors of microscopic texture classification images can provide interesting results.

2.1.1 LBP [5]

The local binary pattern involves computing binary patterns that are present in the circular neighborhood of each pixel. The neighboring cells have a certain number of neighbors P, which are present in a radius R. The goal is to swell the given adjacent pixels that exist around a central pixel. If the average pixel intensity is less than the current pixel intensity, the value 1 is assigned. Otherwise, the value 0 must be assigned. Therefore, for each pixel, a binary pattern can be obtained from the provided environment. We can conclude that a total of different binary patterns, equal to 2P, can be obtained. The calculation formula can be given by:

$$\text{Local Binary Pattern (LBP)} = \sum_{i=0}^{P-1} 2^{i}.\delta(f(q_i)\text{- }f(p))$$

Here, f (p) and f (qi) represent the gray levels for pixels p and qi, respectively.

δ is called the Kronecker function.

A texture descriptor can be formed using the histogram of local binary patterns. The local binary pattern method takes into account the uniform patterns that make up the different number of local binary pattern codes that can be used for the histogram containers, and generally decreases from 36 to 10. The value of P is assumed to be 8 and this produces a 10-D feature vector.

2.1.2 CLBP

A new variant of LBP is the CLBP. It gives a complete model of LBP and has three components being extracted from the local region: sign, pixel, center and magnitude. Following thresholding where threshold is selected to be gray level, the central pixel is binary coded. Similar to LBP the number of neighbors P and a neighborhood of radius R is considered for the other two components. Specific operators are used to code the different signs and magnitudes into the binary format. It yields a 1352 dimension feature vector.

2.1.3 GLCM [6]

GLCM is mainly useful in characterizing texture images. In this experiment, they have taken four different adjacency directions which includes: 0◦, 45◦, 90◦, 135◦. Apart from this, eight different gray levels are taken into account in order to compute the value of GLCM. In the Gray Level Co-occurrence Matrix, computation of 13 Haralick parameters takes place. These features includes: correlation, angular, sum of squares, second moment, variance, contrast, sum average, difference, inverse difference moment, difference entropy, entropy, variance, information measures.

The main feature vector is obtained by taking the average of the given 13 dimension feature vector in all the four directions.

2.1.4 PFTAS [2]

They have used the parameter-free new version of TAS which is PFTAS because Breast Cancer images have a few common features with these images. Its principle is the accumulation of multiple-threshold binarized images in the histogram bins such that pixels are in accordance to their number of white neighbours. A 27T dimension feature vector is formed by concatenation of all three histograms. This results in a 81 dimension feature vector. Finally, a 162 dimension feature vector is formed by concatenation of this vector and its bitwise negated version.

2.1.5 ORB

Oriented FAST and Oriented Rotation (ORB) offers an excellent alternative to traditional methods: SURF and SIFT. These are also key point detectors, which consider performance and computational effort. It has been designed to be resistant to noise and, above all, "rotationally invariant". ORB is based on the well-known BRIEF Keypoint Descriptor and the FAST Keypoint Detector. The ORB works as follows: First, FAST is used to find the key points. Then, the first N points are selected by the detection of Harris corner. An efficient orientation is calculated and then added. This orientation helps with the rotational investment of the ORB. Finally, the image is represented by a single vector of 32 dimensions, which is the average of all existing points.

Classifiers

Four different classifiers have been used to manipulate the mentioned set of features: neural network, random forests (RF), 1-nearest neighbor (1-NN), SVMs and of decision trees. Storing of the entire training data and classification of the test samples is based on a similarity measure, which is done by the k-NN.

SVM is the most popular algorithm which comes under the category of Classification. In this a hyperplane is built in a very high-dimensional space depending upon the conditions. This hyperplane can be used for regression and classification. SVM provides an optimal hyperplane that helps in separating multiple classes. Random Forest takes an ensemble method combining decision tree predictors. Best advantage of the Random Forest is that it is very fast and has the capability to work on various unbalanced datasets.

Feature Selection

In machine learning and insights, feature selection, otherwise called variable selection, variable subset selection or property selection is the way toward choosing a subset of pertinent features (factors, indicators) for use in model development. Feature selection methods are utilized for four reasons:

1. rearrangements of models to make them simpler to translate by analysts/users
2. shorter preparing times
3. to maintain a strategic distance from the scourge of dimensionality
4. upgraded speculation by diminishing overfitting

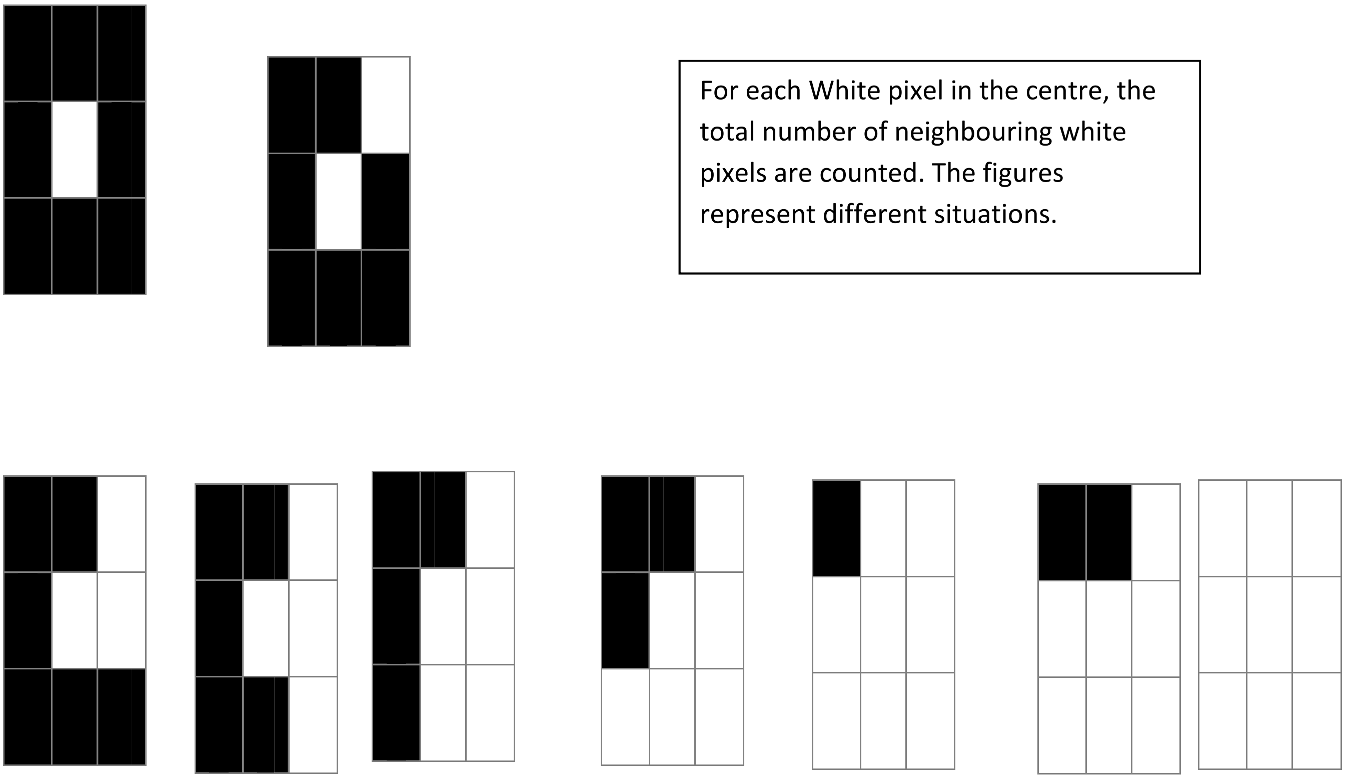
The focal reason when utilizing a feature selection system is that the information contains a few features that are either repetitive or insignificant, and would thus be able to be evacuated without causing much loss of information. Redundant and immaterial are two particular thoughts, since one important feature might be excess within the sight of another significant feature with which it is firmly correlated.

2.2 Cell Phenotype Fast Automated Image Classification [2]

Nicholas Hamilton, Kelly Hanson and Rohan D Teasdale have utilized PFTAS for picture arrangement. Limit neighborhood insights make a parallel picture by applying a chosen edge to the picture. For pixels with a power more prominent than 30, the normal force, ie μ, for the picture is computed. Utilizing the range μ-30 to μ + 30, the picture is then a paired limit. To expand the visual contrasts of the limit pictures, which had unique however outwardly comparable areas, this zone was chosen. The distinction between the edge pictures was taken from these nine insights. The quantity of neighboring white pixels is meant each white pixel. The quantity of white pixels without white neighbors is the main edge measurement. The number with a white neighbor is the second and a most extreme of eight insights are chosen.

Every measurement is partitioned by the aggregate number of white pixels in the edge picture and in this way standardizes. For pictures of parallel edges with pixels in the extents μ-30 to 255 and μ to 255, two arrangements of edge nearness measurements were figured as above, giving an aggregate of 27 insights. An assortment of other edge ranges have been tried, however yielded less execution in the arrangement.

The given image shows the nine threshold statics for cell images:



For each White pixel in the centre, the total number of neighbouring white pixels are counted. The figures represent different situations.

2.3 Breast Cancer Histopathological Image Classification using Convolutional Neural Networks [3]

Here, Fabro Spanhol and Caroline Petitjean utilized the Deep Learning Approach with CNN. Inside and out adapting continually investigates the chances and chances to gain diverse capacities straightforwardly from the given information. In the meantime many specialty capacities are stayed away from. The fundamental idea of inside and out learning is to discover and assess various layer levels that are known as portrayals. Profound Learning means to speak to the most unique semantics of the given information utilizing more elevated amount capacities.

The convolutional neural systems (CNN), which are a piece of the profound learning strategy, have made incredible progress in the field of picture grouping. This has likewise turned out to be fruitful in the examination of therapeutic pictures. Hence, we can state that a convolutional neuronal system (CNN) comprises of trainable stages, whose number is one over the other. This is frequently trailed by the utilization of a managed classifier. It likewise contains varieties of clusters called highlight maps. These component maps speak to both the information and yield of each stage.

A profound neural system is generally prepared by entering the contribution to it and afterward ascertaining the profound neural system in layer by layer design. This creates a last outcome that can be utilized to contrast and the right arrangement. In the wake of ascertaining the blunder in the yield organize, this mistake returns and streams back through the neural system through the regressive spread technique. At each retrogressive advance, the model parameters are tuned consistently toward the path to attempt to diminish the blunder however much as could be expected. This procedure consistently catches every one of the information and, consequently, enhances our model as it moves. Regularly the preparation is done as an iterative procedure. Numerous means of the given information are required. Also, it proceeds until the point when our model at long last meets up. There are three principle sorts of layers that are utilized to build convolutional neural system models: convolutional layer, gathering layer and completely associated layer. By and large, a total convolutional neuronal system (CNN) engineering is acquired when huge numbers of these layers are stacked over one another.

The CNN architecture

Finally, the CNN architecture, which gave the best results in this experiment, consists of the following parameters and layers:

Input layer: This dimension has the assignment of stacking the section and afterward making the yield that will be utilized to encourage the convolutional layers. You can likewise apply a few changes, for example, include scaling and subtraction.

Convolution layers: This layer copies the predetermined info picture with another arrangement of versatile channels. Here, each layer makes a one of a kind element outline our yield picture. There are three collapsing layers in this model. The first and second convolution layers learn 32 distinct channels. At long last, the third last layer learns 64 distinct channels.

Pooling layers: These gathering layers are for the most part in charge of decreasing the testing of the required spatial element of the predetermined information. After each new convolutional layer, there is a solitary gathering layer. Every one of these layers are arranged in a 3 × 3 gathering field (spatial augmentation) with a stage of 2.

Just little fixes of pictures are utilized for preparing to learn CNN parameters. The primary thought is to separate patches from the high-goals fixes whose measure coordinates those of the CIFAR informational index. Since these are surfaces, the most critical prerequisite is that the patches contain enough data to prepare a model with the goal that each picture can separate a suitable arrangement of patches.

The consequences of the fix were joined for the entire picture. They required a system to partition the first test pictures into patches, at that point experience the model and join the outcomes, since the models are prepared in the patches of the pictures.

The ideal outcome was gotten by removing all conceivable patches from the pictures, yet this has ended up being computationally escalated. Rather, they have chosen to extricate the frameworks from the pictures, that is, the arrangement of all non-covering fields that give a sensible harmony between the computational exertion and the order execution.

With a 40x amplification, convolutional neural systems accomplished 5% more precision than straightforward machine learning calculations.

2.4 Breast Cancer Multi-Classification from Histopathological Images with Structured Deep Learning Model [4]

A complete acknowledgment technique has been presented with the new proposed profound engineered apprehensive system (CSDCNN) in light of the class structure to give a substantial and solid answer for bosom disease multi-rating. CSDD misuses the qualities of NN and assumes an essential job in adjusting bosom malignant growth.

CSDCNN is a non-straight portrayal learning model that disposes of highlight extraction in highlight. End finishing learning mode, which consequently learns particular and brought together highlights as high as could reasonably be expected. CSDCNN conquers obstructions of different histopathology photos, which is deliberately structured and connected with the under-aura and between class properties.

Highlight space remove is utilized as a standard to gauge uniform, yet the example separation of the example in a similar classification is a lot higher than the distinctive portions. Along these lines, he has made a few hindrances of nearby highlights associated with the HSD DDNN to control the extraordinary likenesses of various sorts of histoplology photos.

The CSDCNN is precisely planned as a profound model that has many printed layers that figure out how to take in different highlights and multi-class bosom disease laws. The CSDCNN was planned as the accompanying layer.

Input layer: Input layer loads all the histoplology physicular breast cancer and rely on the first coil layer. Hospitographyology is designed to replace photos with average removal as 256 × 256. Input photos include red 2 green arrays of 8-bit arrays of blue-channel channels.

Convolution Layer: The rules are used to extract the layer characterized by the production of neurons associated with the local areas of the previous or input layer. The total weight filter or grain that weigh with the input is called. Each filter size is 3 × 3, 5 × 5 or 7 × 7. This step is the distance between the filter applications. The maximum number of pitches set is 2 which is less than the filter size.

The Caucasus home has been implemented to overlap the windows and initially has been introduced by distributing GSV with 0.01 standard deviation. The last qualifying layer consists of 64 filters that are initially in place by distributing Gioves with the standard deviation of 0.0001. All local weight values have been approved by RLL (correct linear activity).

Pooling layer: The grouping of confusion matrix is under testing on the example outline, decreasing the attributes of a similar element by one. The reasons for this layer are to decrease the commotion, diminish size and increment the field of recipients. The aftereffects of focus layers help keep up anorexia scale and decrease the quantity of parameters.

The normal mortality system with 7 × 7 resiser parts and one stage 1 is utilized by other fixation layers. The last fixation layer, on the grounds that the overall position of each component is the most extreme cleaning methodology with a 3 × 3 responsive field and one stage 2 is utilized.

The advancement technique is utilized for non-adjusted sections of rapid transmission, to lessen the proficiency of amount. The preparation unit is completely finished through the verification and testing stage utilized for true information.

The precision of the general multi-rating exactness is high and has dropped dependable execution. The normal precision at the patient's dimension is 93.2%, while the normal picture exactness is 93.8% for every one of the components. The test and approval pack is nearly a similar exactness, bringing about the CSDCNN demonstrate can keep away from over the top counteractive action and standardization.

2.5 Multi resolution Gray-Scale and Rotation Invariant Texture Classification [5]

Analysis of two-dimensional surfaces has numerous potential applications, for instance, in mechanical surface assessment, remote detecting, and biomedical picture analysis, be that as it may, just a set number of instances of effective misuse of surface exist. A noteworthy issue is that surfaces in reality are frequently not uniform due to varieties in introduction, scale, or other visual appearance. The dim scale invariance is frequently significant due to uneven brightening or incredible inside class inconstancy. In expansion, the level of computational unpredictability of most proposed surface measures is excessively high, as Randen and Husoy deduced in their ongoing broad similar consider including many diverse spatial separating strategies: "A very useful direction for future research is therefore the development of powerful texture measures that can be extracted and classified with a low-computational complexity."

This work centers around gray scale and pivot invariant surface arrangement, which has been tended to by Chen what's more, Kundu and Wu and Wei. The two examinations moved toward dim scale invariance by expecting that the grayscale change is a direct capacity. This is a fairly solid disentanglement, which may restrain the value of the proposed techniques. Chen and Kundu acknowledged gray scale invariance by worldwide standardization of the info picture utilizing histogram leveling. This isn't a general arrangement, nonetheless, as worldwide histogram leveling can't right intraimage (neighborhood) gray scale varieties.

The LBP administrator is an astounding proportion of the spatial structure of nearby picture surface, yet it, by definition, disposes of the other significant property of nearby picture surface, i.e., differentiate, since it relies upon the dim scale. In the event that just pivot invariant surface analysis is wanted, i.e., dim scale invariance isn't required, the execution of LBP can be further improved by joining it with a revolution invariant change measure VARP;R that describes the complexity of neighborhood picture surface. We present the joint dissemination of these two correlative operators,LBP =V AR, as an incredible device for revolution invariant surface characterization.

2.6 Efficient Data Mining for Local Binary Patterns in Texture Analysis [15]

LBP is a straightforward descriptor that looks at the dark dimension of a pixel and its nearby neighborhood and creates a twofold example code. Paired example codes are frequently condensed into a histogram, and a container in the histogram relates to an interesting double code. Various variations of LBP have been proposed to enhance the fundamental LBP. A few specialists have likewise proposed elective habits of misusing the binary pattern codes; for instance, "uniform" designs bunch the binary pattern codes by the quantity of bit advances. Direct or non-straight dimensionality decrease strategies tried to use just the helpful example codes.

In spite of the fact that LBP and its variations perform well, their blends regularly beat the individual descriptors; for example, a multi-goals LBP demonstrated an improvement over single goals and a joint histogram of LBP and a fluctuation measure descriptor of nearby differentiation (VAR) outflanked every one of the descriptors. Consolidating correlative descriptors in a multi-goals setting gives off an impression of being best using the capacity of LBP. Be that as it may, a basic methodology of incorporating a few LBP variations into a solitary or multidimensional histogram might be unwanted. There are numerous LBP variations and various approaches to consolidate them. Every mix is spoken to in a high-dimensional component space. Assessing the careful densities or probabilities of such highlights requires gigantic preparing pictures, and boisterous highlights would antagonistically influence surface analysis. Traditional dimensionality decrease techniques might be insufficient in light of the fact that it is as yet confined to how the underlying element pool was arranged and may prompt another issue of translating the subsequent (or changed) highlights or an extra computational weight on a testing stage. Henceforth, an option, productive, and compelling strategy to completely use LBP and its variations is required.

Proposed is an information digging approach for LBP and its variations. The essential and variations of LBP with different radii are processed, and visit design mining finds the parallel example codes that as often as possible happened inside preparing pictures. The as often as possible happened design codes can be any mix of LBP and its variations and structure the underlying element pool. Since they are visit, the thickness (or likelihood) estimation is solid. A two-arrange include choice strategy chooses the most discriminative highlights. In the main stage, highlights are requested by their pertinence with the given class marks utilizing a common data based foundation. In the second stage, forward component choice picks the best list of capabilities with the most astounding discriminative ability on the preparation pictures. A histogram is constructed utilizing the chose highlights and utilized for surface analysis. We assess our methodology on the surface pictures from the open surface databases.

So as to discover frequent pattern codes, an information mining approach is received, supposed incessant example mining. Regardless of whether a canister is visit or not is controlled by a client indicated limit. It ought to be noticed that visit design mining finds any mix of the double example codes that are visit. This implies we can at the same time inspect an individual descriptor as well as any blend of the descriptors. It is proportional to processing and investigating histograms of a solitary descriptor and joint histograms of each mix of the K descriptors. Thus, "visit receptacles" incorporate containers from single-and various descriptor histograms. Since the recurrence of a canister isn't characteristic of its discriminative capacity, a two-organize include determination is pursued to acquire a lot of discriminative containers (or highlights). In the principal organize, we request the highlights by means of mRMR (least Redundacy Maximum Relevance) foundation which augments the importance between the highlights and the class marks and limits the redundancy among the highlights. In the second stage, forward element choice picks the most discriminative highlights as indicated by the mRMR include request. At long last, order models are built utilizing the discriminative highlights and tried on the approval datasets.

Frequent patterns can be any mix of the descriptors that are joined in the examination. Non-frequent patterns create uproarious highlights and in this way crumble the examination. Utilizing frequent pattern mining, we not just stay away from the tremendous measure of uproarious highlights yet additionally look at different mixes of the descriptors and select the satisfactory mixes for the investigation. In this manner, our methodology looks to locate the most educational mixes of the descriptors and their binary pattern codes for surface picture examination. Any LBP variation and highlight determination strategy could conceivably profit by frequent pattern mining.

CHAPTER 3

SYSTEM DEVELOPMENT

3.1 Analysis

The learning procedure starts with the perception of information, so examples can be discovered in information and prevalent choices can be taken later on which depend on the precedents gave. The principle point is to enable PCs to learn without human help or collaboration and modify their activities as needs be.

The amount and size of malignant growth databases are expanding quickly, yet most are not dissected to discover covered up and profitable learning. Machine learning procedures can be utilized to find shrouded connections and examples. Models created utilizing machine learning systems enable specialists to settle on exact choices.

Accordingly, we utilize programmed learning strategies, for example, SVM, KNN, Random-Forest, and so on to prepare our machine. The gadget adjusts to the predefined information record and gains from the predetermined parameters. From that point forward, machine learning strategies have turned out to be precise in a few fields before. Along these lines, the utilization of machine learning is helpful for the conclusion of malignant growth. Collapsing neural systems work surprisingly better than SVM, KNN and Random Forests. This is on the grounds that, at every one of the dimensions, the weights proceed to return and attempt to diminish the mistake.

The most critical piece of our task is the examination of pictures when programmed learning methods are utilized. To dissect the pictures, we utilize a few descriptors, for example, nearby double examples, ORB, edge nearness measurements without parameters (PFTAS), GLCM. These element extractors help remove the usefulness of each picture. Subsequent to seeing these element vectors, we can at long last train our machine in like manner. At last, this will enable us to get an exact determination of the tumor.

The most extreme exactness was accomplished when Parameter Free Threshold Adjacency Statistics was utilized as the component extractor and SVM was utilized as the Machine

Learning Algorithm. The best outcome was accomplished when parameters were tuned in like manner. With the end goal to tune parameters and gets quick outcomes, Grid Search technique was utilized. In network seek strategy, a scope of parameters is given to the classifier and the calculation at long last takes up the best blend of all the given parameters. The best arrangement of parameters is taken with the end goal that it gives the most extreme exactness.

LBP, CLBP, ORB (descriptors) gave the best outcomes on account of 40X amplification. Though in 200X amplification, the best outcomes were given by PFTAS and GLCM descriptors. By and large execution of the descriptor was best on account of PFTAS.

3.2 System Design

The record has just been separated into train set and test set. The rate is 70% to 30%. Each picture has just been labeled. First we take the trainset organizer. Presently, picture extraction systems, for example, nearby parallel examples, ORB, neighborhood edge measurements without parameters, grayscale event frameworks, and so forth., are connected to each picture in this organizer. These descriptors extricate the attributes as a "and qualities vector". These capacities fill in as parameters for the preparation of our model.

Table 3.1 The feature vector of different descriptors

Descriptor	Feature Number
PFTAS	162
LBP	10
ORB	32
GLCM	13
CLBP	1352

We will train our model with the help of histograms. The feature so extracted is stored in a histogram. This process is done for every image in the train set. Now we will build the model of our classifiers. The classifiers which we will take into account are SVM, KNN, Random Forest and Neural Networks. With the help of our histogram, we will train our model. The most important thing to in this process is to tune thee parameters the accordingly, such that we get the most accurate results.

Once the training is complete, we will take the test set. Now for each image of test set, we will extract the features using feature extraction techniques and then compare its values with the values present in the histogram formed by train set. The output is then predicted for each test image. Now in order to calculate accuracy, we will compare the predicted value with the labeled value. The different metrics that we will use are confusion matrix, accuracy score, f1 score etc.

Proposed Flowchart

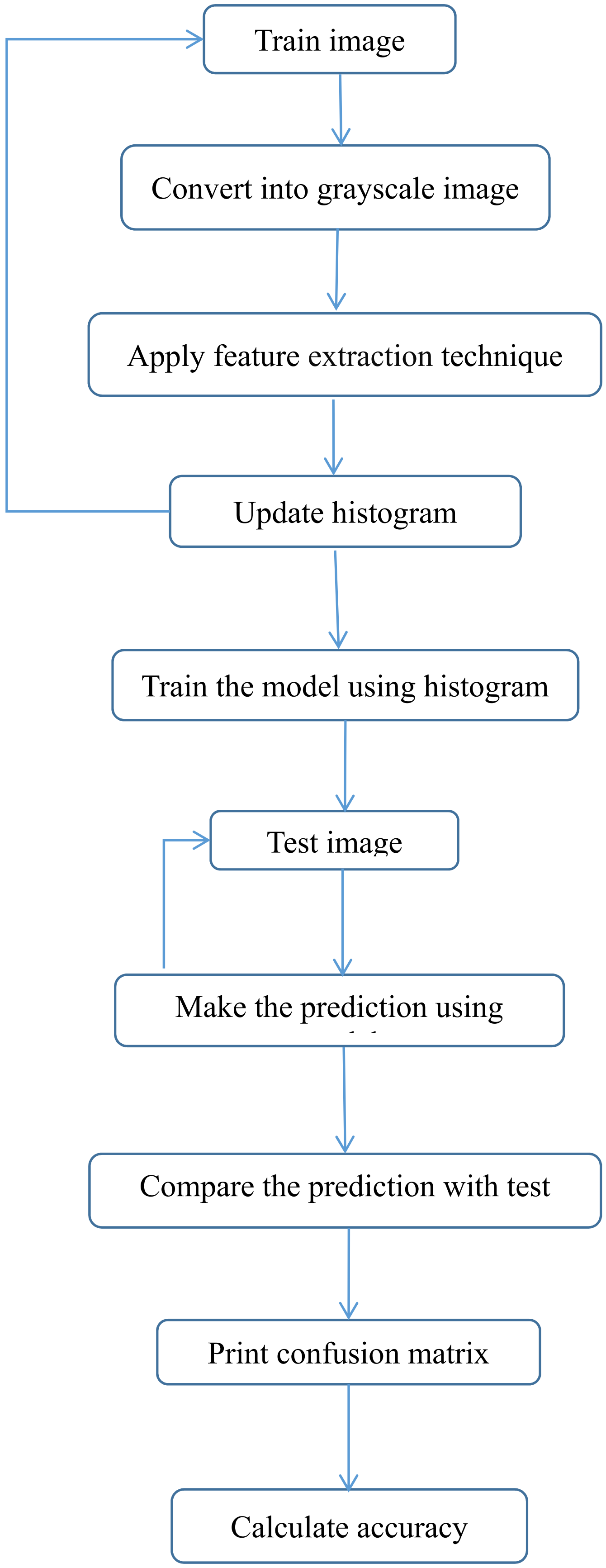

Train image
Convert into grayscale image
Apply feature extraction technique
Update histogram
Train the model using histogram
Test image
Make the prediction using
Compare the prediction with test
Print confusion matrix
Calculate accuracy

Fig 3.1 Flowchart

3.3 Model Development

Our strategy for model improvement is exploratory. The objective of our undertaking is to ensure the conclusion of malignancy with greatest accuracy. This must be accomplished by exploring different avenues regarding distinctive systems from a specific field. We have considered the programmed learning descriptors and algorithms. The highlight extractors which we are utilizing are:

- LBP
- PFTAS
- GLCM
- ORB

The Machine Learning Algorithms that we are using are:

- Support Vector Machines
- Neural Networks
- K nearest neighbors
- Random Forest

Subsequently our point is to locate the best mix which will furnish us with greatest precision. Along these lines this task is absolutely test based. In addition parameter tuning is a noteworthy piece of any Machine Learning Algorithm. Regardless of whether the calculation works exceptionally solid in specific conditions, at that point too because of terrible determination of parameters, the precision could be low. In this manner we likewise need to center around the right arrangement of parameters. Hence parameter tuning must be done in whichever show we pick.

Parameter tuning should either be possible physically or by utilizing the lattice seek technique. Network looking is the procedure in which information is checked with the end goal to discover ideal parameters for some random model. Contingent upon the kind of model that we are utilizing, tuning of specific parameters is vital. Framework seeking applies to a solitary model sort as well as number of models. Network looking can be connected in machine learning with the end goal to ascertain the best parameters for its utilization in some random model. It very well may be computationally greatly costly and may set aside a long opportunity to keep running on the machine. Matrix Search constructs a model on every conceivable parameter mix. At that point it repeats through every parameter blend lastly stores a model for each mix.

Data augmentation increases the value of base information by including data got from inner and outside sources inside a venture. Information is one of the center resources for an undertaking, making information the executives basic. Information growth can be connected to any type of information, yet might be particularly valuable for client information, deals patterns, item deals, where extra data can help give more inside and out understanding.

Information enlargement can help diminish the manual intervention required to created important data and knowledge of business information, just as fundamentally upgrade information quality.

Data augmentation is of the last advances done in big business data the board in the wake of observing, profiling and combination. A portion of the normal strategies utilized in data augmentation include:

1. Extrapolation Technique: Based on heuristics. The applicable fields are refreshed or furnished with qualities.
2. Tagging Technique: Common records are labeled to a gathering, making it more clear and separate for the gathering.
3. Aggregation Technique: Using numerical estimations of midpoints and means, values are assessed for important fields if necessary.
4. Probability Technique: Based on heuristics and logical insights, values are populated dependent on the probability of occasions.

CHAPTER 4

ALGORITHMS

4.1 Machine Learning Algorithms

4.1.1. Support Vector Machines

Support Vector Machine (SVM) is a checked machine learning calculation that is utilized for characterization and relapse difficulties. It is by and large utilized for grouping issues. In this calculation, the information components are plotted as a point in n-dimensional space (where n alludes to the quantity of elements) with the estimation of each element that frames the estimation of a specific facilitate. The order is finished by finding the hyperplane which separates the two classes exceptionally well.

Kernels in SVM: Kernel in Machine Learning is by and large utilized in alluding the piece trap. It is a technique in which a direct classifier is utilized to tackle any non-straight issue. It includes changing directly indivisible information into straightly distinct one. The part work is for the most part connected on each datum example with the end goal to outline given unique non-direct perceptions into a specific higher-dimensional space to such an extent that they wind up divisible.

The different kernels are

a. *Linear Kernel:* Any hyperplane can be written as the given set of points ***x***:

$w \quad . \, x$ - b = 0

here ***w*** is a normal vector to our given hyperplane.

This can be further divided into hard margin and soft margin.

b.*Radial Basis Function kernel:* it is represented as:

$$K(x,x') = \exp(-\frac{||x-x'||^2}{2\sigma^2})$$

$||x - x'||^2$ can be said to be as Euclidean Distance.

c. *Polynomial kernel:* In machine learning, the polynomial kernel can be specified as the kernel function commonly used with support vector machines (SVM) and many other kernel models. They represent the similarity of different vectors (the specified training patterns) in the space of characteristics provided on the polynomials of our original variables, which allows the learning of many possible non-linear models.

All the more formally, a help vector machine builds a hyperplane or set of hyperplanes in a high-or interminable dimensional space, which can be utilized for characterization, relapse, or different assignments like exceptions detection. Intuitively, a great detachment is accomplished by the hyperplane that has the biggest separation to the closest preparing information purpose of any class (alleged practical edge), since all in all the bigger the edge, the lower the speculation blunder of the classifier.

The parameters of the most extreme edge hyperplane are inferred by tackling the improvement. There exist a few specific calculations for rapidly understanding the quadratic programming (QP) issue that emerges from SVMs, for the most part depending on heuristics for separating the issue into littler, progressively reasonable lumps.

Another methodology is to utilize an inside point strategy that utilizes Newton-like cycles to discover an answer of the Karush– Kuhn– Tucker states of the basic and double issues. Rather than tackling an arrangement of separated issues, this methodology legitimately takes care of the issue out and out. To abstain from explaining a direct framework including the substantial part grid, a low-position estimation to the network is regularly utilized in the bit trap.

Another normal technique is Platt's consecutive insignificant improvement (SMO) calculation, which separates the issue into 2-dimensional sub-issues that are illuminated diagnostically, disposing of the requirement for a numerical streamlining calculation and grid stockpiling. This calculation is reasonably straightforward, simple to execute, for the most part quicker, and has better scaling properties for troublesome SVM issues.

4.1.2. K-Nearest Neighbour

The k-Nearest-Neighbors (kNN) order technique in machine learning, and gives an extraordinary method to acquaint one with machine learning and characterization all in all. At the fundamental dimension, it basically characterizes by finding the most comparable information focuses in the preparation dataset and making an informed figure which depends on their classifications.There are two critical choices that should be made before the groupings. One is the estimation of k which will be utilized. It can either be chosen self-assertively or cross-approval can be endeavored to locate an ideal esteem. The following is the separation metric that will be utilized. There are different approaches to process remove yet the correct metric to utilize is constantly dictated by the informational collection and the grouping undertaking. Two ordinarily utilized measurements are Euclidean separation and Cosine likeness. Euclidean separation is the size of the vector acquired by subtracting the preparation information point from the point to be ordered.

$$\text{E(x,y)}=\sqrt{}\sum_{i=1}^{k}(x_i - y_i)^2$$

Other matrices that can be used are Manhattan and Minkowski distance. Manhattan distance is a metric in which the distance between two points is calculated as the sum of the absolute differences of their Cartesian coordinates.

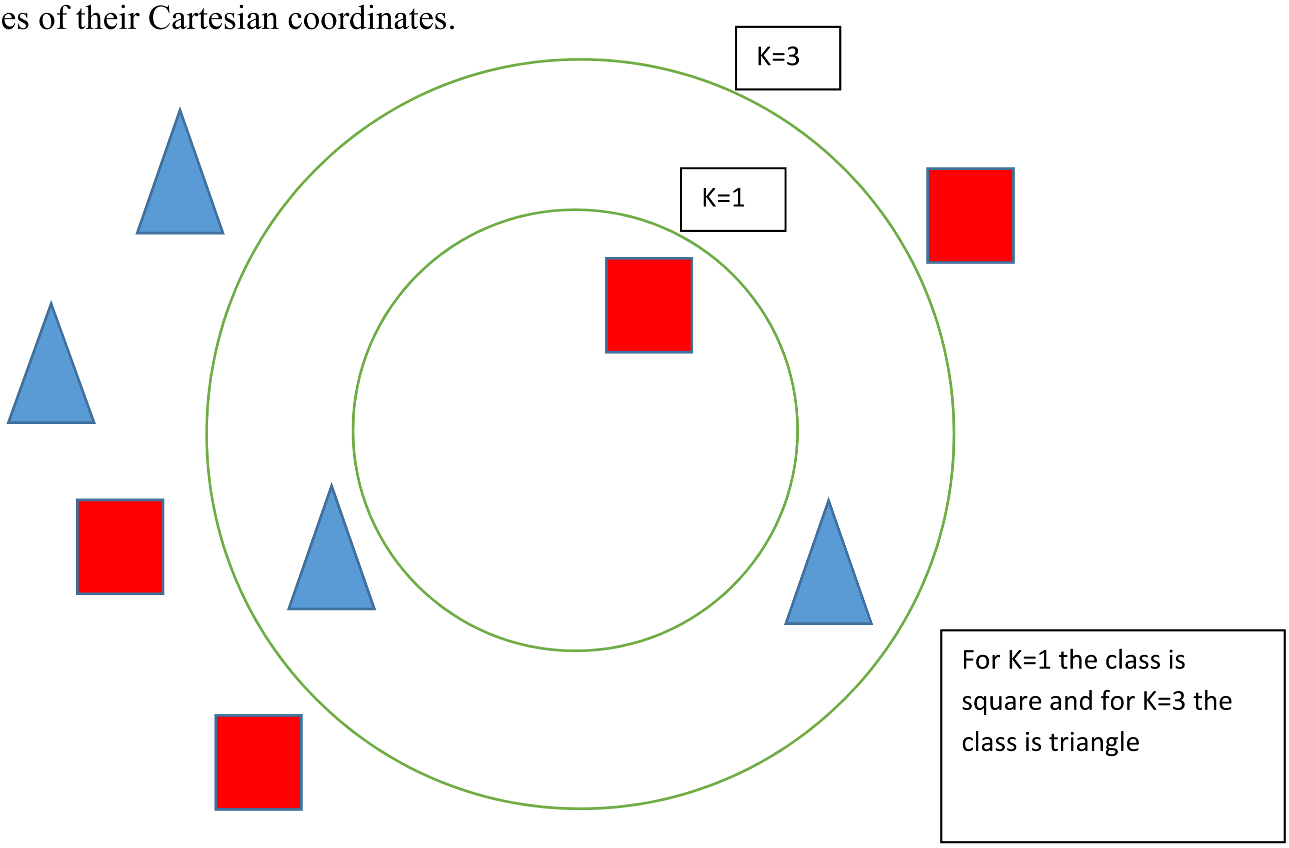

K=3
K=1
For K=1 the class is square and for K=3 the class is triangle

Fig 4.1 K-nearest neighbor classifier

4.1.3 Random Forests

Random Forest (RF) is an extremely adaptable and simple to utilize Machine Learning (ML) calculation. It creates exceptionally precise outcomes even without the high degree of hyper-parameter tuning. Irregular Forest (RF) is additionally a standout amongst the most utilized Machine Learning (ML) calculations. This is on the grounds that it is extremely basic and can likewise be utilized for both characterization and relapse tests.

Fundamentally there are two phases in Random Forest (RF) calculation. First is irregular timberland creation. Second is to play out an expectation from the recently made irregular woods classifier. The entire procedure can be given as:

i Select randomly “K” features from the total “m” features. Here k << m.

ii Now Among these “K” features, using the best split point calculate the node “d” .

iii Then split the node into its daughter nodes by using the best split.

iv Repeat all the steps from a to c until “l” number of nodes has been finally reached.

v Now build the forest by repeating steps a to d for “n” number times in order to create “n” number of trees.

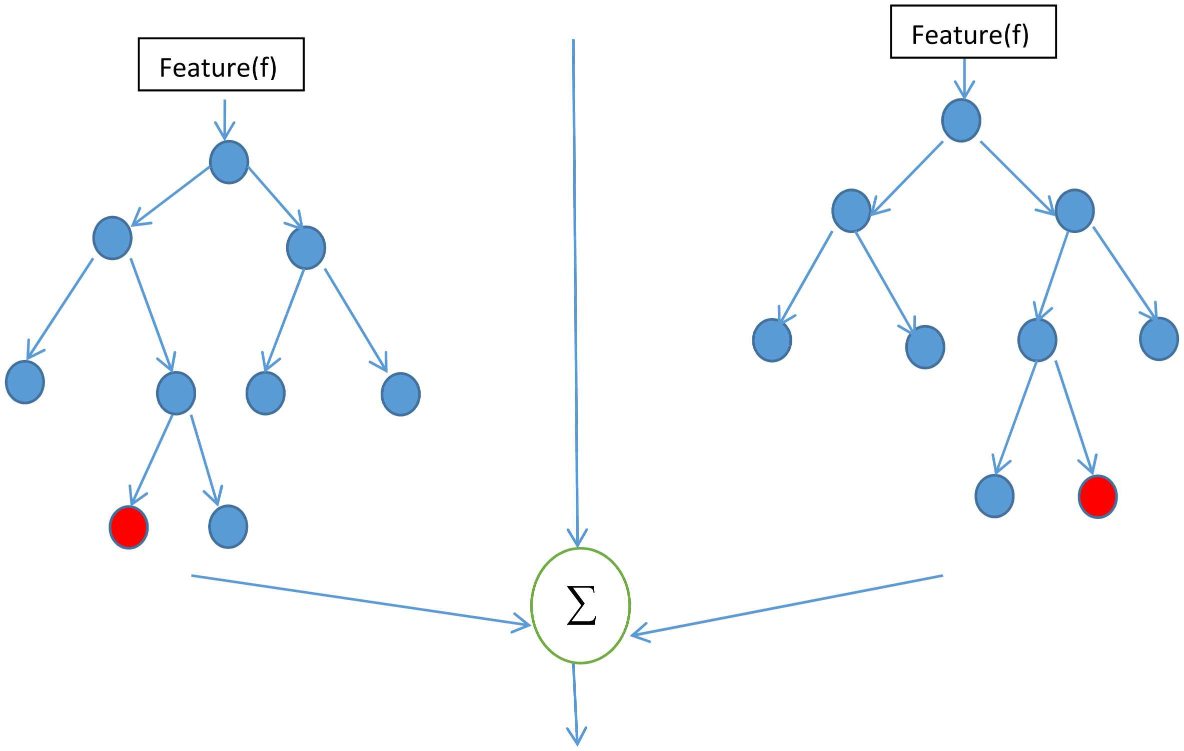


Fig 4.2 Random Forest Classifier

4.1.4 Neural Networks

A neural system (NN) is a worldview of data handling that is roused by the working of organic sensory systems, for example, our cerebrum, which forms data. The primary component of this worldview is the extraordinary novel structure of our data preparing framework. This framework comprises of countless interconnected preparing components (neurons) cooperating. Their fundamental objective is to take care of particular issues. Neural systems (NN) more often than not learn by precedent.

A Neural Networks (NN) is arranged for a specific application. This incorporates information characterization or example acknowledgment through a precise learning process. Learning in the organic frameworks by and large includes acclimations to the principle synaptic associations that typically exist between the neurons. Same is the situation with Neural Networks (NN).

Initiation Functions: Neuron can't learn with just a straight capacity that is appended to it. Any non-straight enactment capacity will dependably give it a chance to pick up as per the distinction as for mistake. Consequently initiation capacities are required.

Different types of activation functions that we will use in this project are:

4.1.4.1 Linear: This function is a line or can also be called linear. Therefore, the output of these functions will not be confined to any range.

Equation can be given as: f(x) = x

Range can be given as: (-infinity to infinity)

It never helps with the complexity or various different parameters of the usual data that is generally fed to the neural networks.

4.1.4.2 Logistic: The Sigmoid Function or the Logistic Function curve looks like a solid S-shape.

The reason why we mainly use logistic function is because of its it existence between (0 to1). Hence, it is particularly used for models where the output to be predicted is probability. Since the probability exists only in the range of 0 and 1, logistic is the right choice. The function is also differentiable. Thus we can find the slope of the logistic function curve at any two given points. The logistic function is monotonic but its derivative is not. The softmax function can be said as a more generalized logistic activation function as it is used for multiclass classification.

4.1.4.3 tanh: tanh is also similar to logistic sigmoid but better. The range of this function is from (-1 to 1) and it is also sigmoidal (s - shaped). The advantage in tanh is that the negative inputs will be strongly mapped negative and zero inputs will be mapped close to zero in the tanh graph. This function is also differentiable. The function is aslo monotonic whereas its derivative is not. tanh is generally used in classification between two classes.

4.1.4.4 Rectified Linear Unit (ReLU): The Regulated Linear Unit (ReLU) is currently the most used activation function in the world. Since then, it has been used in almost all convolutional neuronal networks or deep learning. Its range can be specified as follows: [0 to infinity] The function is monotonic and also its derivative. However, the problem is that all negative values become zero immediately, which quickly reduces our model's ability to properly adjust or train the given data. This means that as soon as the negative entries that are passed to the ReLU activation function, the value in the graph immediately to zero, the resulting graph is affected because the negative values are not assigned accordingly.

4.2 Feature Extractor Algorithms

4.2.1 Local Binary Patterns

Local binary pattern (LBP) is a visual descriptor utilized for grouping in counterfeit vision. You see a ground-breaking surface arrangement highlight. Joining LBP with the histogram of the slope arranged descriptor (HOG) will effectively enhance acknowledgment execution in some datasets. The LBP include vector is made as pursues: partitioned into cells. For every pixel in a cell, the pixel is contrasted with every one of its 8 neighbors (upper left, focus left, base left, upper right, and so forth.). On the off chance that the estimation of the center pixel is more prominent than the estimation of the neighbor, state "0". Something else, express "1". This will give you a 8-digit double number. The histogram is ascertained. This histogram is viewed as an element vector with 256 measurements. The histogram is standardized and the histograms of all cells are connected. This will give you an element vector for the whole window.

$$\text{Local Binary Pattern (LBP)} = \sum_{i=0}^{P-1} 2^{i}.\delta(f(q_i)\text{- } f(p))$$

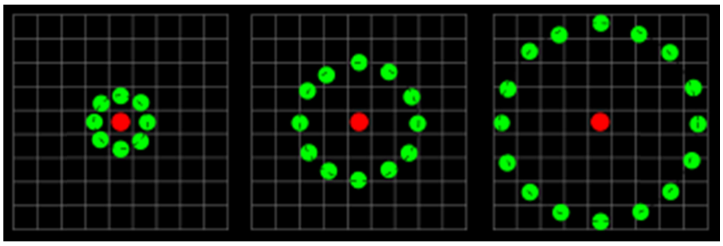


Fig 4.3 Radius and number of points in Local Binary Patterns

4.2.2 Gray-Level Co-Occurrence Matrix

The GLCM is a table of the recurrence with which diverse mixes of pixel splendor esteems (dim dimensions) are created in a picture. It is a factual technique for analyzing the surface that considers the spatial relationship of the pixels. It is additionally called a spatial reliance lattice of dim dimensions. Since I am a grayscale picture and N is the aggregate number of dark dimensions in the picture. The dim dimension coordinating network is a quadratic lattice G of request N, where (I, j) the passage in G speaks to the occasions a pixel of force I is nearby a pixel of power j. The standardized co-event grid is acquired when every component of G is partitioned by the aggregate number of co-event matches in G. The area happens in every one of the four headings (even, vertical, left and right diagonals). For every one of these area bearings, the properties of the Haralick surface are figured.

By averaging the four coordinated concurrence lattices, the properties of the surface are ascertained. To stretch out the ideas to n-dimensional Euclidean space, the grayscale pictures in n-dimensional space and the neighboring headings made reference to above in n-dimensional pictures are exactly characterized.

4.2.3 Oriented FAST and Rotated BRIEF

ORB is a combination of the FAST key point detector and BRIEF descriptor with a few modifications. In order to determine the key points, it uses FAST. Then it applies a Harris corner measure to find top N points. The orientation and rotation variant is not computed by FAST. For the patch with located corner at center,it computes the intensity weighted centroid. The orientation given by the direction of the vector from this corner point to centroid. In order to improve the rotation invariance moments are computed. If there is an in-plane rotation, the descriptor BRIEF poorly performs. ORB computes a rotation matrix using the orientation of patch followed by steering of the descriptors according to the orientation by the BRIEF

4.2.4 Scale Invariant Feature Transform

The Invariant Characteristic Transform (SIFT) change is a calculation for perceiving highlights in PC vision that perceives and portrays the nearby properties in various pictures. Applications incorporate protest acknowledgment, signal acknowledgment, robot mapping and route, 3D demonstrating, video following, and so forth.

Critical SIFTs of the item were taken from various references and put away in the information. The component is perceived by another picture by looking at any new close to home picture of the information and deciding the attributes of the game dependent on the Euclidean thickness of the visual components. The captions of the key focuses relating to the item and the position, weight and course of the new picture are characterized in the incorporated circuits. Altered gatherings are immediately controlled by executing a straightforward difference in the regular chilly table. Each gathering of at least 3 comparative things relates to the itemized examinations and point by point advertisements. At long last, the likelihood is figured in particular arrangements that demonstrate the presence of the protest. All examinations for these tests will be perceived effectively.

4.2.5 PFTAS [2]

They have used the parameter-free new version of TAS which is PFTAS because Breast Cancer images have a few common features with these images. Its principle is the accumulation of multiple-threshold binarized images in the histogram bins such that pixels are in accordance to their number of white neighbours. A 27 dimension feature vector is formed by concatenation of all three histograms. This results in a 81 dimension feature vector. Finally, a 162 dimension feature vector is formed by concatenation of this vector and its bitwise negated version.

4.3 Feature Selection

We as a whole may have confronted this issue of distinguishing the related features from a lot of information and evacuating the immaterial or less significant features with don't contribute a lot to our objective variable so as to accomplish better exactness for our model. Feature Selection is one of the centre ideas in machine learning which enormously impacts the execution of our model. The information features that we use to prepare your machine learning models impact the execution we can accomplish. Immaterial or in part important features can contrarily affect model execution. Feature determination and Data cleaning ought to be the first and most significant advance of your model planning.

Feature selection procedures ought to be recognized from feature extraction. Feature extraction makes new features from elements of the first features, though feature selection restores a subset of the features. Feature selection methods are regularly utilized in areas where there are numerous features and relatively few examples (or information focuses). Original cases for the use of feature selection incorporate the examination of composed writings and DNA microarray information, where there are a huge number of features, and a couple of tens to many examples.

Feature Selection is where we consequently or physically select those features which contribute most to our expectation variable or yield in which we are keen on. Having unessential features in our information can diminish the exactness of the models and influence our model to learn dependent on unimportant features.

The different feature selection techniques used here are:

1. Univariate Selection
2. Feature importance
3. Correlation matrix with heatmap
4. Principal Component Analysis

4.3.1 Univariate Selection

Univariate feature selection analyzes each feature independently to decide the quality of the relationship of the feature with the reaction variable. These techniques are easy to run and comprehend and are when all is said in done especially useful for picking up a superior comprehension of information (yet not really for upgrading the feature set for better speculation). There are part of various choices for univariate selection. Statistical tests can be utilized to choose those features that have the most grounded association with the yield variable. The scikit-learn library gives the SelectKBest class that can be utilized with a suite of various statistical tests to choose a particular number of features.

4.3.2 Feature Importance

We get the feature significance of each feature of our dataset by utilizing the feature significance property of the model. Feature significance gives us a score for each feature of our information, the higher the score increasingly significant or important is the feature towards our yield variable.

Feature importance is an inbuilt class that accompanies Tree Based Classifiers.

4.3.3 Correlation heatmap

Correlation states how the highlights are identified with one another or the objective variable. Correlation can be certain (increment in one estimation of highlight builds the estimation of the objective variable) or negative (increment in one estimation of highlight diminishes the estimation of the objective variable). Heatmap makes it simple to distinguish which highlights are most identified with the objective variable.

4.3.4 Principal Component Analysis

Principal component analysis (PCA) is a factual system that utilizes a symmetrical change to change over a lot of perceptions of conceivably related factors (elements every one of which takes on different numerical qualities) into a lot of estimations of directly uncorrelated factors called principal components. On the off chance that there are n perceptions with p factors, at that point the quantity of unmistakable principal components is. This change is characterized so that the primary principal component has the biggest conceivable fluctuation (that is, represents however much of the inconstancy in the information as could be expected), and each succeeding component thus has the most elevated difference conceivable under the requirement that it is symmetrical to the previous components. The subsequent vectors (each being a straight blend of the factors and containing n perceptions) are an uncorrelated symmetrical premise set. PCA is delicate to the overall scaling of the first factors.

Principal Component Analysis (or PCA) utilizes straight polynomial math to change the dataset into a packed structure. For the most part this is known as an information decrease procedure. A property of PCA is that you can pick the quantity of measurements or principal component in the changed outcome.

PCA is generally utilized as a device in exploratory information analysis and for making prescient models. Usually used to imagine hereditary separation and relatedness between populaces. PCA should be possible by eigenvalue deterioration of an information covariance (or connection) framework or particular esteem disintegration of an information network, as a rule after a standardization venture of the underlying information. The standardization of each characteristic comprises of mean focusing – subtracting every datum esteem from its variable's deliberate mean with the goal that its exact mean (normal) is zero – and, potentially, normalizing every factor's fluctuation to make it equivalent to 1. The consequences of a PCA are normally talked about as far as component scores, in some cases called factor scores (the changed variable qualities relating to a specific information point), and loadings (the weight by which each institutionalized unique variable ought to be duplicated to get the component score). If component scores are institutionalized to unit fluctuation, loadings must contain the information difference in them (and that is the extent of eigenvalues). On the off chance that component scores are not institutionalized (consequently they contain the information change) at that point loadings must be unit-scaled, ("standardized") and these loads are called

eigenvectors; they are the cosines of symmetrical revolution of factors into principal components or back.
PCA can be thought of as fitting a p-dimensional ellipsoid to the information, where every pivot of the ellipsoid speaks to a principal component. In the event that some pivot of the ellipsoid is little, at that point the fluctuation along that hub is likewise little, and by excluding that hub and its relating principal component from our portrayal of the dataset, we lose just a proportionately little measure of data.

To discover the tomahawks of the ellipsoid, we should initially subtract the mean of every factor from the dataset to base the information on the beginning. At that point, we figure the covariance lattice of the information, and compute the eigenvalues and relating eigenvectors of this covariance grid. At that point we should standardize each of the symmetrical eigenvectors to move toward becoming unit vectors. When this is done, each of the commonly symmetrical, unit eigenvectors can be deciphered as a pivot of the ellipsoid fitted to the information. This decision of premise will change our covariance lattice into a diagonalised structure with the inclining components speaking to the fluctuation of every pivot. The extent of the fluctuation that each eigenvector speaks to can be determined by isolating the eigenvalue relating to that eigenvector by the aggregate all things considered.

This system is delicate to the scaling of the information, and there is no accord regarding how to best scale the information to get ideal outcomes.

4.4 Hyperparameter Tuning

In machine learning, a hyperparameter is a parameter whose esteem is set before the learning procedure starts. Paradoxically, the estimations of different parameters are determined by means of preparing.

Diverse model preparing calculations require distinctive hyperparameters, some basic calculations, (for example, common least squares relapse) require none. Given these hyperparameters, the preparation calculation takes in the parameters from the information. For example, LASSO is a calculation that adds a regularization hyperparameter to customary least squares relapse, which must be set before assessing the parameters through the preparation calculation.

Most execution variety can be credited to only a couple of hyperparameters. The tunability of a calculation, hyperparameter, or collaborating hyperparameters is a proportion of how much execution can be picked up by tuning it. For a LSTM, while the learning rate pursued by the system estimate are its most critical hyperparameters, though grouping and energy have no noteworthy impact on its execution.

Albeit some examination has upheld the utilization of small group sizes in the thousands, other work has discovered the best execution with smaller than usual clump sizes somewhere in the range of 2 and 32.

An inborn stochasticity in learning straightforwardly suggests that the observational hyperparameter execution isn't really its actual performance. Methods that are not vigorous to basic changes in hyperparameters, arbitrary seeds, or even various usage of a similar calculation can't be incorporated into mission basic control frameworks without huge improvement and robustification.

Fortification learning calculations, specifically, require estimating their execution over an extensive number of irregular seeds, and furthermore estimating their affectability to decisions of hyperparameters. Their assessment with few arbitrary seeds does not catch execution sufficiently because of high variance. Some support learning techniques, for example DDPG (Deep Deterministic Policy Gradient), are more touchy to hyperparameter decisions than others

In machine learning, hyperparameter improvement or tuning is the issue of picking a lot of ideal hyperparameters for a learning calculation. A hyperparameter is a parameter whose esteem is utilized to control the learning procedure. On the other hand, the estimations of different parameters (ordinarily hub loads) are found out.

A similar sort of machine learning model can require various limitations, loads or learning rates to sum up various information designs. These measures are called hyperparameters, and must be tuned with the goal that the model can ideally take care of the machine learning issue. Hyperparameter streamlining finds a tuple of hyperparameters that yields an ideal model which limits a predefined misfortune work on given free data. The target work takes a tuple of hyperparameters and returns the related loss. Cross-approval is frequently used to assess this speculation execution.

4.4.1 Grid Search

The customary method for performing hyperparameter enhancement has been network look, or a parameter clear, which is essentially a comprehensive seeking through a physically indicated subset of the hyperparameter space of a learning calculation. A lattice seek calculation must be guided by some execution metric, ordinarily estimated by cross-approval on the preparation set or assessment on a held-out approval set.

Since the parameter space of a machine student may incorporate genuine esteemed or unbounded esteem spaces for specific parameters, physically set limits and discretization might be important before applying matrix seek.

For instance, a common delicate edge SVM classifier outfitted with a RBF bit has no less than two hyperparameters that should be tuned for good execution on concealed information: a regularization steady C and a portion hyperparameter γ. The two parameters are nonstop, so to perform network seek, one chooses a limited arrangement of "sensible" values for each, state:

$C = \{10, 1000, 100000\}$

$\gamma = \{0.001, 0.01, 0.1, 1, 10, 100\}$

Framework seek at that point prepares a SVM with each pair (C, γ) in the Cartesian result of these two sets and assesses their execution on a held-out approval set (or by inward cross-approval on the preparation set, in which case different SVMs are prepared per pair). At long last, the network look calculation yields the settings that accomplished the most elevated score in the approval system.

Network look experiences the scourge of dimensionality, however is regularly embarrassingly parallel in light of the fact that normally the hyperparameter settings it assesses are autonomous of one another.

4.4.2 Random search

Random Search replaces the thorough list of all mixes by choosing them haphazardly. This can be just connected to the discrete setting portrayed above, yet additionally sums up to nonstop and blended spaces. It can beat Grid look, particularly when just few hyperparameters influences the last execution of the machine learning calculation. For this situation, the improvement issue is said to have a low natural dimensionality. Arbitrary Search is likewise embarrassingly parallel, and also permits the incorporation of earlier information by indicating the dispersion from which to test.

Arbitrary Search proposes arrangements haphazardly from your parameter space. While less normal in machine learning practice than lattice seek, arbitrary hunt has been appeared to discover equivalent or preferred qualities over matrix look inside less capacity assessments for specific sorts of issues. To upgrade with irregular pursuit, assess your capacity at some number of arbitrary setups in the parameter space; note that it might be hazy how to decide the quantity of capacity assessments required for your specific issue.

There are numerous hypothetical and pragmatic concerns while assessing improvement techniques. The best system for your concern is the one that finds the best esteem the fastest– with the least capacity assessments. The best discovered esteem and the Area Under the Curve are two estimations to analyze streamlining techniques. These estimations joined with the Mann-Whitney U Test enable us to own thorough factual expressions about which streamlining systems performs better. SigOpt utilizes Bayesian streamlining to enable you to locate the best parameters for a given capacity or model the quickest, and we as often as possible utilize these techniques to decide our very own execution against famous enhancement methodologies.

4.5 Data Augmentation

Data augmentation increases the value of base information by including data got from inner and outside sources inside a venture. Information is one of the center resources for an undertaking, making information the executives basic. Information growth can be connected to any type of information, yet might be particularly valuable for client information, deals patterns, item deals, where extra data can help give more inside and out understanding.

Information enlargement can help diminish the manual intervention required to created important data and knowledge of business information, just as fundamentally upgrade information quality.

Data augmentation is of the last advances done in big business data the board in the wake of observing, profiling and combination. A portion of the normal strategies utilized in data augmentation include:

1. Extrapolation Technique: Based on heuristics. The applicable fields are refreshed or furnished with qualities.
2. Tagging Technique: Common records are labeled to a gathering, making it more clear and separate for the gathering.
3. Aggregation Technique: Using numerical estimations of midpoints and means, values are assessed for important fields if necessary.
4. Probability Technique: Based on heuristics and logical insights, values are populated dependent on the probability of occasions.

Data augmentation techniques:

Position Augmentation: One type of growth influences the situation of pixel esteems.
Utilizing mixes of slicing, scaling, translating, rotating and flipping the estimations of the first picture can be moved to make new pictures. A few activities (like scaling and turn) require introduction as pixels in the new picture are blends of pixels in the first picture.

CHAPTER 5

TEST PLAN

5.1 Dataset

The Breast Cancer Histopathological Image Classification (BreakHis) comprises of 9,109 images of tumor tissue which has been collected from multiple patients using various magnifying factors (40X, 100X, 200X, and 400X). Till date it consists of 5,429 malignant and 2,480 benign samples (700X460 pixels).

Characteristics

BreaKHis dataset is divided into two groups: malignant tumors and benign tumors. Histologically the term benign refers to a lesion that does not match any criteria of malignancy e.g. disruption of basement membranes, marked cellular atypia, metastasize, mitosis, etc. Normally, benign tumors are “innocents” i.e they remain localized and are slow growing. Malignant is a synonym for cancer: lesion can invade and destroy adjacent structures and spreads to distant sites which causes death.

In the current version of this dataset, samples that are present in the dataset have been collected by SOB method, also named excisional biopsy or partial mastectomy .This procedure when compared to other methods of needle biopsy, removes the larger size of the tissue sample. This is done in a hospital with general anesthetic. The sorting is based on the way the tumor looks under the microscope. Various types of breast tumors can have a different type of prognoses and treatment implications. The dataset contains four distinct types of benign breast tumors: Phyllodes Tumor, Adenosis, Tubular Adenoma, Fibroadenoma, and four distinct types of malignant breast tumors: Lobular Carcinoma, Ductal Carcinoma, Papillary Carcinoma, Mucinous Carcinoma.

BreaKHis dataset is as follows:

Table 5.1 BreakHis dataset image distribution

Magnification	Benign	Malignant	Total
40X	652	1370	1995
100X	644	1437	2081
200X	623	1390	2013
400X	588	1232	1820
Total Images	2480	5429	7909

5.2 Test Metrics

5.2.1 Confusion Matrix

A confusion matrix is a procedure which abridges the execution of an order calculation. It is a synopsis of forecast results on a characterization issue. Characterization precision can be deceiving there are an unequal number of perceptions in each class or if there are in excess of two classes in the dataset. Ascertaining a disarray lattice gives a superior thought of what the characterization demonstrate is getting right and the sorts of blunders it is making.

The quantity of off base forecasts and right expectations are outlined with check esteems and are separated by each class. This goes about as the way to the disarray grid. The manners by which the order display is befuddled when it makes forecasts is appeared by the disarray lattice. It gives an understanding into the mistakes being made by your classifier. It is this breakdown that conquers the restriction of utilizing characterization exactness alone.

Calculation of Confusion Matrix

The method for computing a confusion matrix is demonstrated as follows.

An arrangement of test information or an approval informational index is required with the normal outcome esteems. The forecast is made for each line in the test informational collection. From the normal outcomes and conjectures, coming up next are considered: The quantity of erroneous estimates for every classification. The quantity of right expectations for each class sorted out by the class gave. These numbers are sorted out into a table as pursues:

It is normal from the side: each line of the framework compares to an anticipated class. Forecast at the best: Each section of the table relates to a genuine class. Right and off base grouping numbers are finished in the table. The line esteem is normal for this class and the anticipated section an incentive for this class is loaded up with the aggregate number of right forecasts for a class.

Additionally, the request expected for that class esteem and the anticipated an incentive for this segment class is loaded up with the aggregate number of erroneous expectations for a class. Practically speaking, a parallel classifier like this can complete two kinds of mistakes: it is erroneously credited to a man who has not showed up in the predefined classification or is wrongly ascribed to a man who has not showed up in the predefined class. Deciding these two sorts of blunders is frequently a region of intrigue. A disarray framework is an advantageous method to show this kind of data.

This grid can without much of a stretch be utilized for issues in two classes where it is straightforward, however it can likewise be connected to issues with at least 3 class esteems, adding more lines and segments to network perplexity.

Accuracy

Accuracy is one of the measures to evaluate classification models. The precision is the fraction of the predictions given by the classification model. The precision has the following definition: Accuracy = Total no. of the correct forecasts / From predictions

For the binary classification, the accuracy can be calculated as negative and positive in the following way:

$$\text{Accuracy}=((TP+TN)/(TP+TN+FP+FN))$$

TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

Table 5.2 Representation of Confusion Matrix

	Predicted Value	**Predicted Value**
Real Value	True Positive(TP) Reality: Malignant ML model predicted: Malignant	False Positive(FP) Reality: Benign ML model predicted: Malignant
Real Value	False Negative(FN) Reality: Malignant ML model predicted: Benign	True Negative(TN) Reality: Benign ML model predicted: Benign

Precision

The precision determines how often it is correct when the model predicts positive. Accuracy helps determine when the cost of false positives is high.

$$\text{Precision} = TP / (TP + FP)$$

where TP is the number of real positives and FP the number of false positives. Precision refers to the ability of the classifier not to designate a positive sample as negative.

Recall

Recall it helps to determine how much the false negative cost is.

$$\text{Recall} = TP / (TP + FN)$$

Where TP is true positive and the number of FN is false negative number. Recall refers to the classification capability to find all the classified samples.

F1 Score

F1 is a measure of purity of the test. It checks both accuracy and memory. This is considered right when F1 score is 1 and there is a total failure of 0.

$$F1 = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

5.3 Test Setup

The test process is already in-built in our system. The testing process taking place just after the model is trained. After the completion of the training process, we analyze each image present in the test set. In order to analyze each image, we use descriptors to extract features. Now we compare these feature values with the feature values which were initially retained using the train set. The comparison is done according to the Machine Learning model used and finally the output for each image is received. Since each image is already labeled, we can compute accuracy by comparing the predicted value with the received values.

CHAPTER 6

RESULTS

We implemented the previously stated procedure using Local Binary Pattern as feature extractor and SVM as our machine learning model. The major problem faced by us was that of **class imbalance.** The train and test images of Ductal Carcinoma were thrice to that of other classes which led to the class imbalance problem. Implementation of various feature extracting techniques combined with machine learning models and determining the best combination is our main area of concern. Now in order to solve the class imbalance problem, various data augmentation techniques are used. Flipping of images in order to create a new image, rotation of images and zooming of image helped the model to learn in a more efficient way. Better results were achieved using Grid Search method. Experimental method is used to get the results in this project.

Here I have worked on two datasets. The first dataset has been defined earlier as having 8 classes. Ductal Carcinoma is always large in number as it is the most common form of breast cancer. In this dataset, Support Vector Machine and Random Forest gave better outputs with Local Binary Pattern as feature extractor. A range of parameter was fed to the model in order to implement the grid search method.

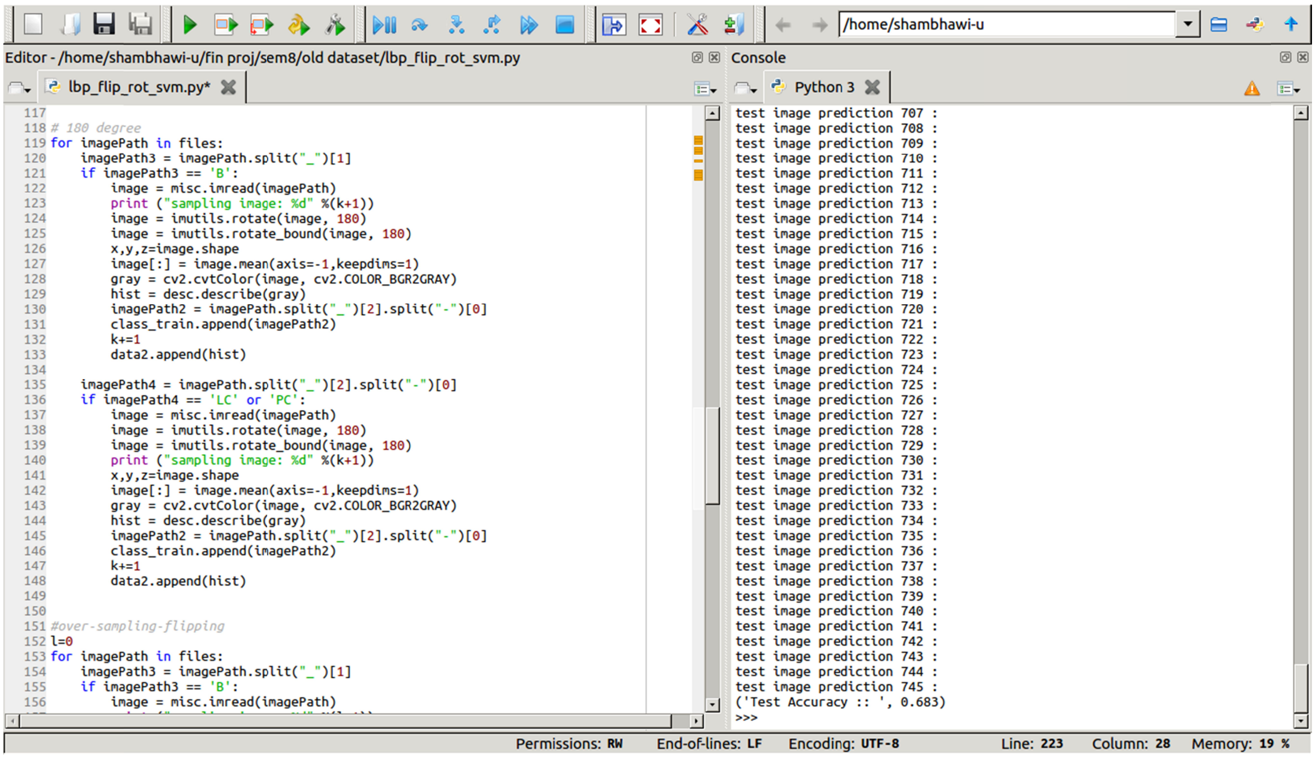
In the first dataset Support Vector Machine proved to be the best with Local Binary Patterns using the Grid Search Method. Grid Search method helped us to find out the most suitable parameters in order to find the best accuracy. Moreover the data augmentation techniques helped to increase our data thereby improving the learning of our model. Here random forest and neural networks gave average results. The results were average even when augmented data was used combined with grid search method.

In the second dataset, the images were first normalized and then the features were extracted. The best result was achieved by neural networks classifier. The most effective activation function was tanh. But in this, the learning was not good. When grid search method was combined with decision tree, the train accuracy was 100%, thereby proving that the learning was perfect. The best accuracy was achieved in the same. Support Vector Machines proved to be the worst in this dataset.

Table 6.1 Result on dataset 1 using LBP

Classifier	Normal Test Accuracy		Grid Search Accuracy
SVM	40.134%		68.3%
Random Forest	40%		40%
Neural Network	Identity	37.85%	-
	Relu	40%	
	Logistic	39%	
	Tanh	39.9%	

Fig 6.1 SVM Grid Search Method on dataset 1



/home/shambhawi-u
Editor - /home/shambhawi-u/fin proj/sem8/old dataset/lbp_flip_rot_svm.py
lbp_flip_rot_svm.py*
Console
Python 3

```
# 180 degree
for imagePath in files:
    imagePath3 = imagePath.split("_")[1]
    if imagePath3 == 'B':
        image = misc.imread(imagePath)
        print ("sampling image: %d" %(k+1))
        image = imutils.rotate(image, 180)
        image = imutils.rotate_bound(image, 180)
        x,y,z=image.shape
        image[:] = image.mean(axis=-1,keepdims=1)
        gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        hist = desc.describe(gray)
        imagePath2 = imagePath.split("_")[2].split("-")[0]
        class_train.append(imagePath2)
        k+=1
        data2.append(hist)

    imagePath4 = imagePath.split("_")[2].split("-")[0]
    if imagePath4 == 'LC' or 'PC':
        image = misc.imread(imagePath)
        image = imutils.rotate(image, 180)
        image = imutils.rotate_bound(image, 180)
        print ("sampling image: %d" %(k+1))
        x,y,z=image.shape
        image[:] = image.mean(axis=-1,keepdims=1)
        gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        hist = desc.describe(gray)
        imagePath2 = imagePath.split("_")[2].split("-")[0]
        class_train.append(imagePath2)
        k+=1
        data2.append(hist)


#over-sampling-flipping
l=0
for imagePath in files:
    imagePath3 = imagePath.split("_")[1]
    if imagePath3 == 'B':
        image = misc.imread(imagePath)
```

```
test image prediction 707 :
test image prediction 708 :
test image prediction 709 :
test image prediction 710 :
test image prediction 711 :
test image prediction 712 :
test image prediction 713 :
test image prediction 714 :
test image prediction 715 :
test image prediction 716 :
test image prediction 717 :
test image prediction 718 :
test image prediction 719 :
test image prediction 720 :
test image prediction 721 :
test image prediction 722 :
test image prediction 723 :
test image prediction 724 :
test image prediction 725 :
test image prediction 726 :
test image prediction 727 :
test image prediction 728 :
test image prediction 729 :
test image prediction 730 :
test image prediction 731 :
test image prediction 732 :
test image prediction 733 :
test image prediction 734 :
test image prediction 735 :
test image prediction 736 :
test image prediction 737 :
test image prediction 738 :
test image prediction 739 :
test image prediction 740 :
test image prediction 741 :
test image prediction 742 :
test image prediction 743 :
test image prediction 744 :
test image prediction 745 :
('Test Accuracy :: ', 0.683)
>>>
```

Permissions: RW   End-of-lines: LF   Encoding: UTF-8   Line: 223   Column: 28   Memory: 19 %

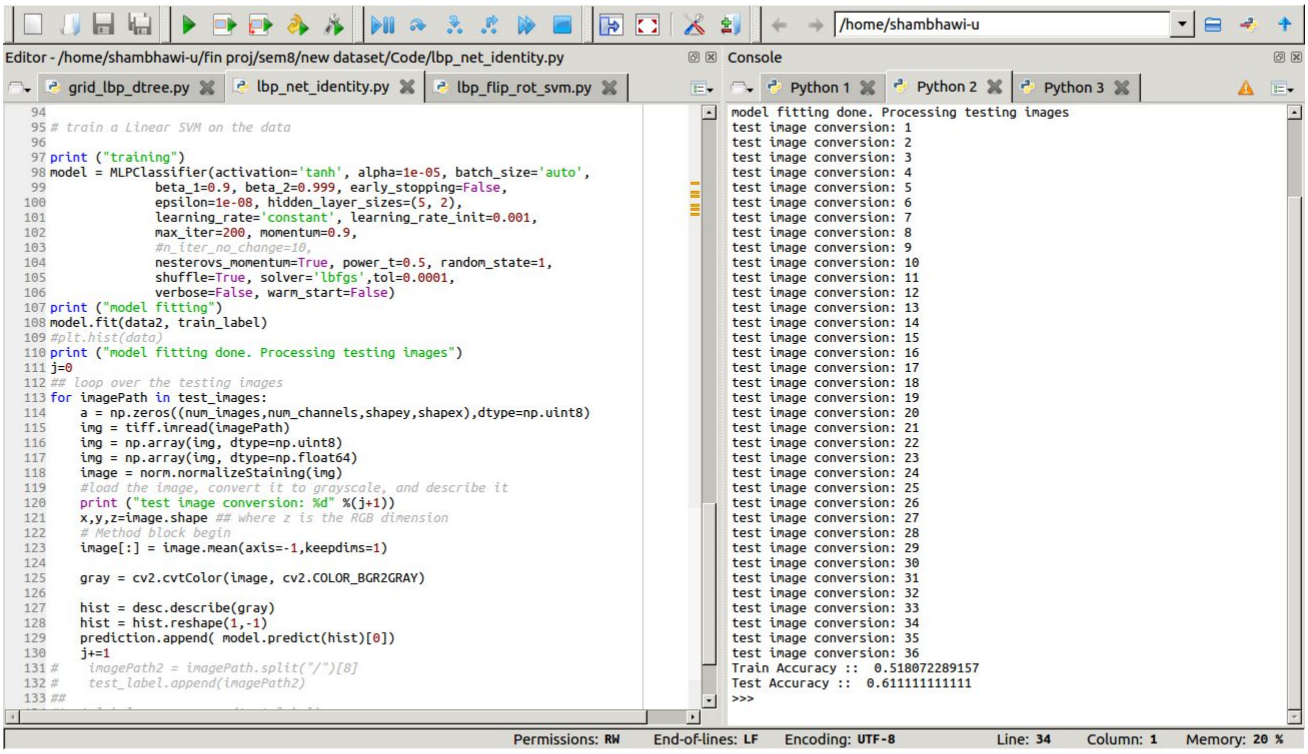
The second dataset has less number of images so data augmentation plays an important part here. Moreover the images have been normalized before the features are being extracted. Here neural networks with activation function identity and tanh gave the best results.

The results for this dataset is given below:

Table 6.2 Result on Dataset 2 using LBP

Classifier		Train Accuracy	Test Accuracy
SVM		38%	25%
Random Forest		53.8%	44.44%
Decision Tree		79%	36%
Neural Network	Identity	55.42	52.78%
	Tanh	51.80%	61.11%

Fig 6.2 Neural Networks result on dataset 2



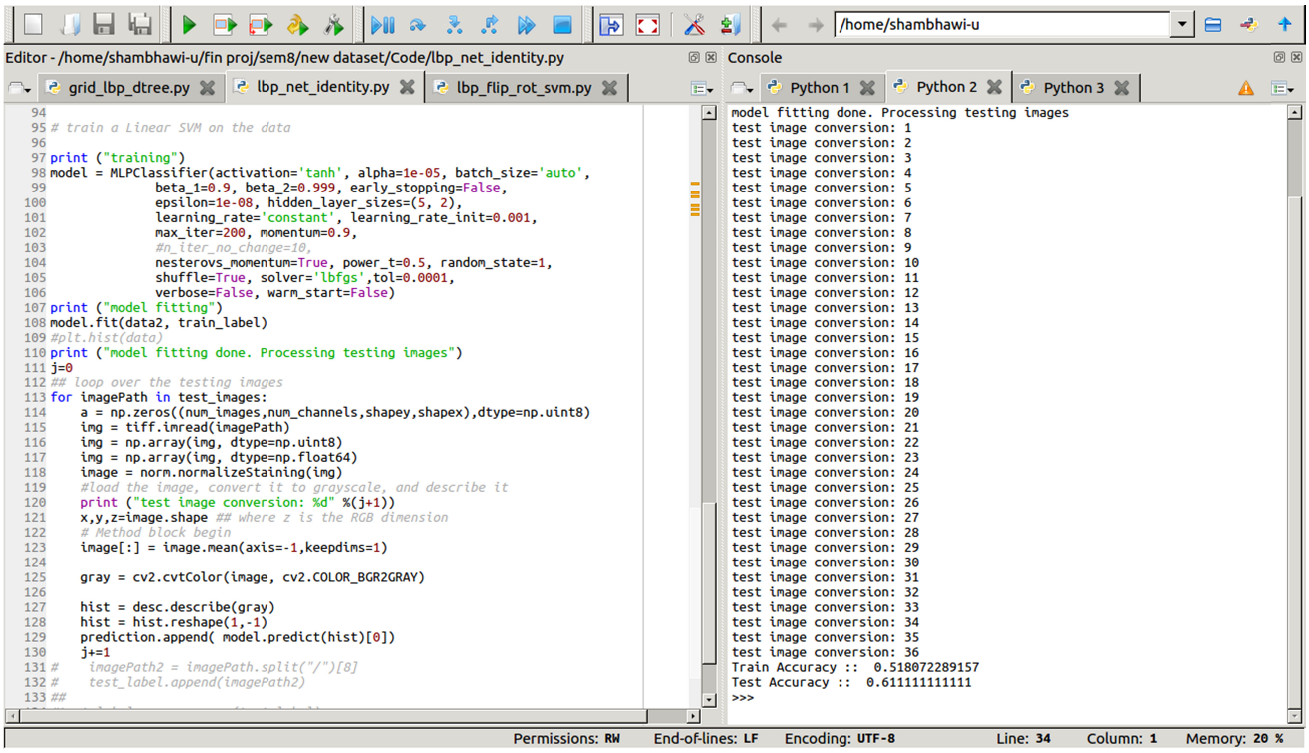
```python

# train a Linear SVM on the data

print ("training")
model = MLPClassifier(activation='tanh', alpha=1e-05, batch_size='auto',
               beta_1=0.9, beta_2=0.999, early_stopping=False,
               epsilon=1e-08, hidden_layer_sizes=(5, 2),
               learning_rate='constant', learning_rate_init=0.001,
               max_iter=200, momentum=0.9,
               #n_iter_no_change=10,
               nesterovs_momentum=True, power_t=0.5, random_state=1,
               shuffle=True, solver='lbfgs',tol=0.0001,
               verbose=False, warm_start=False)
print ("model fitting")
model.fit(data2, train_label)
#plt.hist(data)
print ("model fitting done. Processing testing images")
j=0
## loop over the testing images
for imagePath in test_images:
    a = np.zeros((num_images,num_channels,shapey,shapex),dtype=np.uint8)
    img = tiff.imread(imagePath)
    img = np.array(img, dtype=np.uint8)
    img = np.array(img, dtype=np.float64)
    image = norm.normalizeStaining(img)
    #load the image, convert it to grayscale, and describe it
    print ("test image conversion: %d" %(j+1))
    x,y,z=image.shape ## where z is the RGB dimension
    # Method block begin
    image[:] = image.mean(axis=-1,keepdims=1)

    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    hist = desc.describe(gray)
    hist = hist.reshape(1,-1)
    prediction.append( model.predict(hist)[0])
    j+=1
#    imagePath2 = imagePath.split("/")[8]
#    test_label.append(imagePath2)
##
```

```
model fitting done. Processing testing images
test image conversion: 1
test image conversion: 2
test image conversion: 3
test image conversion: 4
test image conversion: 5
test image conversion: 6
test image conversion: 7
test image conversion: 8
test image conversion: 9
test image conversion: 10
test image conversion: 11
test image conversion: 12
test image conversion: 13
test image conversion: 14
test image conversion: 15
test image conversion: 16
test image conversion: 17
test image conversion: 18
test image conversion: 19
test image conversion: 20
test image conversion: 21
test image conversion: 22
test image conversion: 23
test image conversion: 24
test image conversion: 25
test image conversion: 26
test image conversion: 27
test image conversion: 28
test image conversion: 29
test image conversion: 30
test image conversion: 31
test image conversion: 32
test image conversion: 33
test image conversion: 34
test image conversion: 35
test image conversion: 36
Train Accuracy ::  0.518072289157
Test Accuracy ::  0.611111111111
>>>
```

Table 6.3 Result on Dataset 2 using LBP with Grid Search

Classifier	Train Accuracy	Test Accuracy
SVM	27%	50%
Random Forest	53.81%	30%
Decision Tree	100%	47.22%

Fig 6.3 Decision Tree Output with grid search



```

# train a Linear SVM on the data

print ("training")
model = MLPClassifier(activation='tanh', alpha=1e-05, batch_size='auto',
              beta_1=0.9, beta_2=0.999, early_stopping=False,
              epsilon=1e-08, hidden_layer_sizes=(5, 2),
              learning_rate='constant', learning_rate_init=0.001,
              max_iter=200, momentum=0.9,
              #n_iter_no_change=10,
              nesterovs_momentum=True, power_t=0.5, random_state=1,
              shuffle=True, solver='lbfgs',tol=0.0001,
              verbose=False, warm_start=False)
print ("model fitting")
model.fit(data2, train_label)
#plt.hist(data)
print ("model fitting done. Processing testing images")
j=0
## loop over the testing images
for imagePath in test_images:
    a = np.zeros((num_images,num_channels,shapey,shapex),dtype=np.uint8)
    img = tiff.imread(imagePath)
    img = np.array(img, dtype=np.uint8)
    img = np.array(img, dtype=np.float64)
    image = norm.normalizeStaining(img)
    #load the image, convert it to grayscale, and describe it
    print ("test image conversion: %d" %(j+1))
    x,y,z=image.shape ## where z is the RGB dimension
    # Method block begin
    image[:] = image.mean(axis=-1,keepdims=1)

    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    hist = desc.describe(gray)
    hist = hist.reshape(1,-1)
    prediction.append( model.predict(hist)[0])
    j+=1
#    imagePath2 = imagePath.split("/")[8]
#    test_label.append(imagePath2)
##
```

```
model fitting done. Processing testing images
test image conversion: 1
test image conversion: 2
test image conversion: 3
test image conversion: 4
test image conversion: 5
test image conversion: 6
test image conversion: 7
test image conversion: 8
test image conversion: 9
test image conversion: 10
test image conversion: 11
test image conversion: 12
test image conversion: 13
test image conversion: 14
test image conversion: 15
test image conversion: 16
test image conversion: 17
test image conversion: 18
test image conversion: 19
test image conversion: 20
test image conversion: 21
test image conversion: 22
test image conversion: 23
test image conversion: 24
test image conversion: 25
test image conversion: 26
test image conversion: 27
test image conversion: 28
test image conversion: 29
test image conversion: 30
test image conversion: 31
test image conversion: 32
test image conversion: 33
test image conversion: 34
test image conversion: 35
test image conversion: 36
Train Accuracy ::  0.518072289157
Test Accuracy ::  0.611111111111
>>>
```

CHAPTER 7
CONCLUSION

The project tries to attempt to explain, compare and assess the performance of various machine learning techniques that can be applied to cancer prediction and prognosis. A number of trends will be recognized a with respect to the different types of machine learning techniques being used, the given dataset being integrated, the type predictions being made, the various classes of cancers being studied and the performance of these different methods in predicting cancer susceptibility.

When machine learning techniques are compared to expert-based systems or conventional statistical systems, it is found that machine learning techniques generally improve the performance with increased accuracy of most prognoses.

Most of the studies are generally well constructed and well validated, there is a need for greater attention to experimental design and implementation, especially with respect to the quality and quantity of biological data. The overall quality and reproducibility of many machine-based classifiers will be enhanced by improvements in experimental design accompanied by improved biological validation.

If quality of studies continues to improve, that day is not far away where use of machine learning classifier will become a common scenario in various clinical and hospital settings.

In the coming years, if the correct predictions are made by the machine then diagnosis of breast cancer will become very easy, thereby reducing the chances of wrong treatments. This will reduce death rate due to cancer.

REFERENCES

[1] Fabio A. Spanhol, Luiz S. Oliveira, Caroline Petitjean, and Laurent Heutte, “A Dataset for Breast Cancer Histopathological Image Classification”, IEEE TRANSACTION ON BIOMEDICAL ENGINEERING, VOL. 63, NO. 7, JULY 2016.

[2] Nicholas A Hamilton Radosav S Pantelic, Kelly Hanson and Rohan D Teasdale1, “Fast automated cell phenotype image classification”, BMC Bioinformatics, Published: 30 March 2007.

[3] Fabio Spanhol, Caroline Petitjean, “Breast Cancer Histopathologicl Image Classification using Convolutional Neural Networks”, IEEE Transaction, Published: July 2016.

[4] Zhongyi Han, Benzheng Wei1, Yuanjie Zheng, Yilong Yin, Kejian Li & Shuo Li, “Breast Cancer Multi-classification from Histopathological Images with Structured Deep Learning Model”, Scientific Reports, Published: Feb 2017.

[5] T. Ojala et al., “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” IEEE Trans. Pattern Anal.Mach.Intell., vol. 24, no. 7, pp. 971–987, Jul. 2002.

[6] R. Haralick et al., “Textural features for image classification,” IEEE Trans. Syst. Man Cybern., vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[7] E. Rublee et al., “ORB: An efficient alternative to SIFT or SURF,” in Proc. IEEE Int. Conf. Comput. Vision, 2011, pp. 2564–2571.

[8] L. P. Coelho et al., “Structured literature image finder: extracting information from text and images in biomedical literature,” in Linking Literature, Information, and Knowledge for Biology (ser. LNCS) vol. 6004, C. Blaschke and H. Shatkay, Eds. New York, NY, USA: Springer, 2010, pp. 23–32.

[9] C. Cortes and V. Vapnik, “Suport-vector networks,” Mach. Learning, vol. 20, pp. 273–297, 1995.

[10] T. Fawcett, “An introduction to ROC analysis,” Pattern Recog. Lett., vol. 27, pp. 861–874, 2006.

[11] Z. Guo et al., "A completed modeling of local binary pattern operator for texture classification," IEEE Trans. Image Process., vol. 19, no. 6, pp. 1657–1663, Jun. 2010.

[12] J. Paivarinta et al., "Volume local phase quantization for blur-insensitive dynamic texture classification," in Proc. 17th Scandinavian Conf. Image Anal., 2011, pp. 360–369.

[13] P. Boyle and B. Levin, Eds., World Cancer Report 2008. Lyon: IARC, 2008. [Online]. Available: http://www.iarc.fr/en/publications/pdfs-online/wcr/2008/wcr_2008.pdf

[14] G. Bradski, "The OpenCV library," Dr. Dobb's Journal of Software Tools, 2000, vol. 25(11), pp. 120–125.

[15] Jin Tae Kwak, Sheng Xu, Bradford J. Wood, "Effective Data Mining for Local Binary Pattern in Texture Image Anlysis" NCBI Expert System, June,2015.

APPENDICES

1. The first implementation is based on the use of Local Binary Patterns as descriptor and Support Vector Machines as the Machine Learning Model.

   Code:

```
# -*- coding: utf-8 -*-
from pyimagesearch.localbinarypatterns import LocalBinaryPatterns
from sklearn.metrics import roc_curve, auc
from sklearn import svm,grid_search
from sklearn.svm import LinearSVC
from imutils import paths
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt
import cv2
from scipy import misc
from skimage import data
import glob
import os
import numpy as np
from sklearn.metrics import accuracy_score,confusion_matrix
import seaborn as sns
train_path = "/home/shambhawi-u/summer internship/40X/lbp/fold1/images/train"
test_path = "/home/shambhawi-u/summer internship/40X/lbp/fold1/images/test"
# initializing the local binary patterns descriptor along with
# the data and label lists
#p-number of points in the neighborhood
#r-radius of circle
desc = LocalBinaryPatterns(8, 1)
data2 = np.load("train_image.npy")
imagePath2 = []
class_train = np.load("train_label.npy")
class_test = np.load("test_label.npy")
```

```
prediction = []
#train_image = []
#train_label = []
#test_image = []
#test_label = []
data_path1 = os.path.join(train_path,'*g') #for joining path for train and test datasets
files = glob.glob(data_path1)
i=0;
# loop over the training images
#for imagePath in files:
#       #load the image first. Then convert it to grayscale. Finally use the descriptor to
extract features
#       image = misc.imread(imagePath)
#       print ("train image conversion: %d" %(i+1))
#       x,y,z=image.shape ## where z is the RGB dimension
#       ## Method block begin
#       image[:] = image.mean(axis=-1,keepdims=1)
#       gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
#       hist = desc.describe(gray)
#       #extract the label from the image path, then update the label and data lists
#       imagePath2 = imagePath.split("_")[2].split("-")[0]
#       class_train.append(imagePath2)
#       i+=1
#       data2.append(hist)
#data2 = np.asarray(data2)
#np.save("train_image.npy",data2)
#class_train = np.asarray(class_train)
#np.save("train_label.npy",class_train)
# train a Linear SVM on the data
print "training....................."
model = svm.SVC(C=10000, cache_size=200, class_weight=None, coef0=0.0,
decision_function_shape='ovr', degree=3, gamma=0.001, kernel='rbf',max_iter=-1,
probability=False, random_state=None, shrinking=True,
                         tol=0.001, verbose=False)
model.fit(data2, class_train)
data_path2 = os.path.join(test_path, '*g')
files2 = glob.glob(data_path2)
j=0
# loop over the testing images
for imagePath in files2:
	# load the image. Then convert it to grayscale and finally describe it,
  # and classify it
	image = misc.imread(imagePath)
	x,y,z=image.shape ## where z is the RGB dimension
	### Method block begin
	image[:] = image.mean(axis=-1,keepdims=1)
	gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
	print ("test image prediction %d : " %(j+1))
	hist = desc.describe(gray)
	hist = hist.reshape(1,-1)
	prediction.append( model.predict(hist)[0])
	#imagePath2 = imagePath.split("_")[2].split("-")[0]
	#class_test.append(imagePath2)
	j+=1
#class_test = np.asarray(class_test)
#np.save("test_label.npy",class_test)
#evaluation of accuracy
#print "Train Accuracy :: ", (model.fit(data2,class_train).score(data2,class_train))
print "Test Accuracy :: ", (accuracy_score(class_test,prediction))
print "confusion matrix: "
mat = confusion_matrix(class_test, prediction)
sns.heatmap(mat.T, square=True, annot=True, fmt='d', cbar=False)
plt.xlabel('true label')
plt.ylabel('predicted label');
```

2. The second implementation is based on the use of Local Binary Patterns with Support Vector Machines. Here grid search method is used to enhance the accuracy.

Code:

```
# -*- coding: utf-8 -*-
from pyimagesearch.localbinarypatterns import LocalBinaryPatterns
from sklearn import svm,grid_search
from sklearn.svm import LinearSVC
from imutils import paths
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt
import cv2
from scipy import misc
from skimage import data
import glob
import os
import numpy as np
from sklearn.metrics import accuracy_score
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
#/home/shambhawi-u/summer internship/mk_fold/fold1/train/40X
train_path = "/home/shambhawi-u/summer internship/40X/lbp/fold1/images/train"
test_path = "/home/shambhawi-u/summer internship/40X/lbp/fold1/images/test"
args = {"training":train_path, "testing":test_path}
# firstly initialize the local binary patterns descriptor along with
# the data and label lists
#p-number of points in the neighborhood
#r-radius of circle
desc = LocalBinaryPatterns(8, 1)
data2 = np.load("train_image.npy")
class_train = np.load("train_label.npy")
class_test = np.load("test_label.npy")
```

```
imagePath2 = []
prediction = []
data_path1 = os.path.join(train_path,'*g')
files = glob.glob(data_path1)
i=0;
# loop over the training images
#for imagePath in files:
#       # load the image. Then convert it to grayscale, and finally describe it
#       image = misc.imread(imagePath)
#       print ("train image conversion: %d" %(i+1))
#       x,y,z=image.shape ## where z is the RGB dimension
#       ### Method block begin
#       image[:] = image.mean(axis=-1,keepdims=1)
#       gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
#       hist = desc.describe(gray)
#       # finally extract the label from the given image path, and then update the
#       # label and data lists
#       imagePath2 = imagePath.split("_")[2].split("-")[0]
#       class_train.append(imagePath2)
#       i+=1
#       data2.append(hist)
print "training...................."
tuned_parameters = [{'kernel': ['rbf'], 'gamma': [1e-1,1e-2,1e-3,1e-4],
                                'C': [0.0001,0.001,0.01,1, 10, 100,
1000,10000],'tol':[0.00001,0.0001,0.001,0.01,0.1]},
                                {'kernel': ['linear'], 'C': [1, 10, 100,
1000],'tol':[0.00001,0.0001,0.001,0.01,0.1]},
                                {'kernel':['poly'], 'degree':
[3,4,5,6,7,8,9,10],'C':[0.0001,0.001,0.01,1, 10, 100,
1000,10000],'tol':[0.00001,0.0001,0.001,0.01,0.1]}]
model = svm.SVC(cache_size=200, class_weight=None, coef0=0.0,
      decision_function_shape='ovr',max_iter=-1, probability=False,
random_state=None, shrinking=True, verbose=False)
```

```
scores = ['precision', 'recall']
for score in scores:
    print("# Tuning hyper-parameters for %s" % score)
    print()
    clf = GridSearchCV(model, tuned_parameters, cv=5,scoring='%s_macro' % score)
    clf.fit(data2, class_train)
    print("Best parameters set found on development set:")
    print()
    print(clf.best_params_)
    print()
    print("Grid scores on development set:")
    print()
    means = clf.cv_results_['mean_test_score']
    stds = clf.cv_results_['std_test_score']
    for mean, std, params in zip(means, stds, clf.cv_results_['params']):
        print("%0.3f (+/-%0.03f) for %r" % (mean, std * 2, params))
    print()
data_path2 = os.path.join(test_path, '*g')
files2 = glob.glob(data_path2)
j=0
# loop over the testing images
for imagePath in files2:
    # load the image, then convert it to grayscale, and finally describe it and classify it
    image = misc.imread(imagePath)
    x,y,z=image.shape ## where z is the RGB dimension
    image[:] = image.mean(axis=-1,keepdims=1)
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    print ("test image conversion %d : " %(j+1))
    hist = desc.describe(gray)
    hist = hist.reshape(1,-1)
    prediction.append( clf.predict(hist)[0])
```

```
#       imagePath2 = imagePath.split("_")[2].split("-")[0]
#       class_test.append(imagePath2)
        j+=1

# imagePath2 = imagePath.split("_")[2].split("-")[0]
print (accuracy_score(class_test,prediction))
print "confusion matrix: "

cm = confusion_matrix(class_test, prediction)
plt.clf()
plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Wistia)
classNames = ['a','f', 'pt','ta', 'dc','lc','mc','pc']
plt.title('Breast Cancer Histopathological Image Classification')
plt.ylabel('True label')
plt.xlabel('Predicted label')
tick_marks = np.arange(len(classNames))
plt.xticks(tick_marks, classNames, rotation=45)
plt.yticks(tick_marks, classNames)
s = [['TN','FP'], ['FN', 'TP']]

for i in range(2):
        for j in range(2):
                plt.text(j,i, str(s[i][j])+" = "+str(cm[i][j]))
plt.show()
```

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 09/05/2019

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | [✓ B.Tech Project Report] | Paper

Name: SHAMBHAWI PAL Department: CSE Enrolment No 151238

Contact No. 8629010255 E-mail. shambhawipal@gmail.com

Name of the Supervisor: Dr. AMIT KUMAR

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): SMART CANCER DIAGNOSIS USING MACHINE LEARNING TECHNIQUES

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 65
- Total No. of Preliminary pages = 5
- Total No. of pages accommodate bibliography/references = 9

Shambhawi Pal

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at 13 (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

9/5/19

(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
09.05.2019	• All Preliminary Pages • Bibliography/Images/Quotes • ~~14~~ 20 Words String	29%	Word Counts	13,424
Report Generated on			Character Counts	73,159
09.05.2019		Submission ID	Total Pages Scanned	53
		1127528630	File Size	1.05M

Checked by
Name & Signature Ashok

Chandan 09.05.2019
Librarian

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com