

THYROID CANCER DETECTION FROM CLINICAL DATA BY USING MACHINE LEARNING ALGORITHM

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology degree in Biotechnology
Interdisciplinary project

BY

SARITHRA. A. P (41230054)
JENNIE ROSHINI P (41240015)



DEPARTMENT OF BIOTECHNOLOGY
SCHOOL OF BIO AND CHEMICAL ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
CATEGORY -1 UNIVERSITY BY UGC

Accredited with grade “A++” by NAAC | 12 B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119

MAY 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

[DEEMED TO BE UNIVERSITY]

CATEGORY-I UNIVERSITY BY UGC

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **SARITHRA.A.P(41230054)** and **JENNIE ROSHINI P(41240015)** who carried out the project entitled "**THYROID CANCER DETECTION FROM CLINICAL DATA BY USING MACHINE LEARNING ALGORITHM**" under our supervision from January 2024 to April 2024 .

R. Thyagarajan

INTERNAL GUIDE

DR. R THYAGARAJAN, M.Phil., PhD.,

V. Ramesh Kumar

HEAD OF THE DEPARTMENT

DR. V RAMESH KUMAR, M.Phil., PhD.,

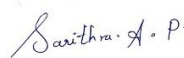
Submitted for Viva voce Examination held on 06.05.2024

Jagadeesh
Internal Examiner

S. H. Saeed
External Examiner

DECLARATION

We, **SARITHRA.A.P (Reg.No.41230054)** and **JENNIE ROSHINI Reg.No.41240015)** hereby declare that the Project Report entitled “**THYROID CANCER DETECTION FROM CLINICAL DATA USING MACHINE LEARNING ALGORITHM**” done under the guidance of **Dr. R THYAGARAJAN M.Phil., PhD.,Associate Professor, Department of Biotechnology** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Biotechnology.



Sarithra A. P.

DATE: 06.05.2024

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleasure to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully.

I convey my sincere thanks **Dr V Ramesh Kumar, M.Phil., Ph.D., Head of the Department, Department of Biotechnology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my project Guide **Dr. R Thyagarajan ,M.Phil.,Ph.D., Associate professor, Department of biotechnology** for their valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my project work.

I also express my thanks to **all Teaching and Non-teaching staff members** of the Department of Biotechnology who were helpful in many ways for the completion of the project work.

ABSTRACT

Thyroid cancer is cancer that develops from the tissues of the thyroid gland. .The Ultrasound is performed to confirm the presence of the thyroid nodule and assess the status of the whole gland . Some ultrasound results may report a TI-RADS or TIRADS score to categorize the risk of malignancy. Thyroid cancer can be benign or malignant based on the tumor size, vascularity and elasticity by the radiologists. Recent advancements of ML algorithms such as decision trees classifiers and Deep neural networks are used in prediction and detection of various diseases to overcome these disadvantages .In this study we have approached a Machine learning algorithm that can be used to interpret the ultrasound images of the thyroid gland to locate the malignant cells. The model is developed using CNN and Bagged CART algorithm. CNN architecture that is Convolutional Neural Network, it's a type of deep learning algorithm commonly used in machine learning tasks related to image recognition and classification. CNNs are particularly effective in analyzing ultrasound images of the thyroid to differentiate between benign and malignant nodules. Bagged CART that is Bootstrap Aggregation is a general procedure that can be used to reduce the variance for those algorithms that have high variance. The Bagged CART algorithm is applied to make accurate predictions from multiple ML algorithms than the individual model. The bagged Cart is sensitive to the datasets and changes the decision trees based on the changes in the trained data ,this increases the efficiency of the tumor detection and minimizes the error. The performance of the model is evaluated with the measures such as Accuracy ,Precision ,Recall F1 score and confusion matrix .The evaluated model id used to detect and classify the thyroid tumor as Benign or Malignant from the test data. This study ensures to classify the benign and malignant thyroid tumors of large Datasets of the ultrasound images of the thyroid gland and can be applied for the diagnostic and research purpose where large datasets are analyzed and predicted accurately.

Keywords: Benign, Malignant, Ultrasound Images, Convolutional Neural Network, Bagged CART,Accuracy,Precision,Recall,F1 score, Confusion Matrix

TABLE OF CONTENTS

S.NO	CONTENTS	PAGE.NO
	ABSTRACT	i
	LIST OF FIGURES	iii
	LIST OF TABLES	iv
	LIST OF ABBREVIATIONS	v
1	INTRODUCTION	1
	1.1 THYROID TUMOR	1
	1.2 CLINICAL IMAGES	2
	1.3 BAGGED CART ALGORITHM	3
	1.4 THYROID CANCER DETECTION USING MACHINE LEARNING	4
2	LITERATURE REVIEW	5
3	AIM AND SCOPE	10
	3.1 AIM	10
	3.2 SCOPE	10
4	MATERIALS AND METHODS	11
	4.1 MATERIALS	11
	4.2 METHODOLOGY	11
	4.2.1 DATA COLLECTION	11
	4.2.2 DATA PREPROCESSING	12
	4.2.3 FEATURE SELECTION	13
	4.2.4 MODEL SELECTION	14
	4.2.5 MODEL TRAINING	15
	4.2.6 MODEL EVALUATION	17
5	RESULT AND DISCUSSION	20
	5.1 TRAINING AND TESTING ACCURACY	20
	5.2 CONFUSION MATRIX	21
	5.2.1 COMPONENTS OF CONFUSION MATRIX	22
	5.2.2 CONFUSION MATRIX PARAMETRES	23
	5.3 DETECTION OF THYROID CANCER	24
6	SUMMARY AND CONCLUSION	26
	REFERENCE	29
	APPENDIX	31

LIST OF ABBREVIATIONS

ANN	Artificial neural network
BACC	Balanced accuracy
CART	Classification and Regresssion tree
CNN	Convolutional neural network
FN	False negative
FP	False positive
IHC	Immunohistology chemistry
KNN	K nearest neighbour
NPV	Negative predictive value
PCA	Principle component analysis
SGD	Stochastic gradient descent
SVM	Support vector machine
TN	True negative
TP	True positive

LIST OF FIGURES`

FIGURE NO	TITLE	PAGE NO
1.1	Thyroid Tumor	1
1.2	MRI of thyroid gland	2
1.3	CT of thyroid gland	2
1.4	Ultrasound of thyroid gland	3
4.1	Block diagram of Methodology	11
4.1	Benign thyroid tumor	12
4.2	Malignant thyroid tumor	13
4.3	Normal thyroid gland	13
4.4	Feature selection for embedded trees	13
4.5	Precision, accuracy, recall, F1 score ,confusion matrix	19
5.1	Graphical representation of validation accuracy and validation loss	21
5.2	Visualization of confusion matrix	23
5.3	Normal thyroid detection for test image	24
5.4	Malignant thyroid detection for test image	24

LIST Of TABLES

TABLE.NO	TITLE	PAGE.NO
5.1	Test Accuracy and Test Loss	20
5.2	Precision,Recall,F1 score ,Support	23

CHAPTER -1

INTRODUCTION

1.1 THYROID TUMOR:

The thyroid is a butterfly-shaped gland located at the base of the neck. The thyroid produces hormones that regulate heart rate, blood pressure, body temperature and weight. Thyroid cancer is a growth of cells that starts in the thyroid .It can cause signs and symptoms, such as swelling in your neck, voice changes and difficulty swallowing.



Fig 1.1 Thyroid Tumor

(Source: Uofl health)

The tumor can grow and can spread (metastasize) to the lymph nodes in the neck. Various studies have shown that India has been reported with 42 million cases of thyroid cancer as of 2020. Sometimes the cancer cells can spread beyond the neck to the lungs, bones and other parts of the body. The thyroid cancer can be either benign or malignant based on the nature of the tumor cells to spread through the other parts of the body. Benign tumors are non-progressive and non-metastatic whereas malignant tumor is metastatic in nature.

1.2 CLINICAL IMAGES:

Various imaging techniques are used for the diagnosis of the thyroid tumor such as computed tomography (CT) , magnetic resonance imaging (MRI) and ultrasound (US). Ultrasound (US) has become an important diagnostic tool in the assessment of thyroid nodules. It is highly sensitive for detecting nodules, and the sonographic features of the nodules can be used to determine the need for further investigation. The ultrasound images is analyzed by the Radiologists , trained technicians and medical professionals. The thyroid cancer is classified based on the size and shape of the nodule such as cystic or spongiform and echogenic behavior. The wrong diagnosis or overdiagnosis by the practitioners is solved by the use of the ML algorithms where there is a accurate prediction of the tumor based on the input ultrasound image.

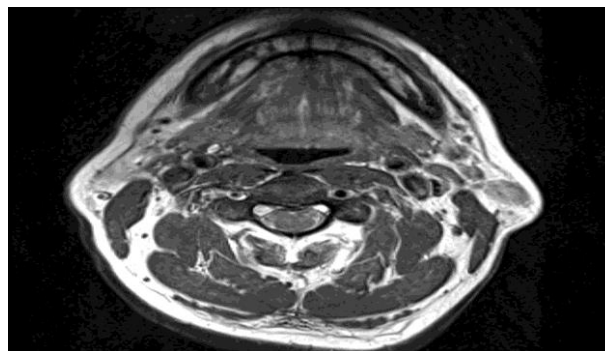


Fig 1.2.MRI of thyroid gland

(Source: Wiki Doc)

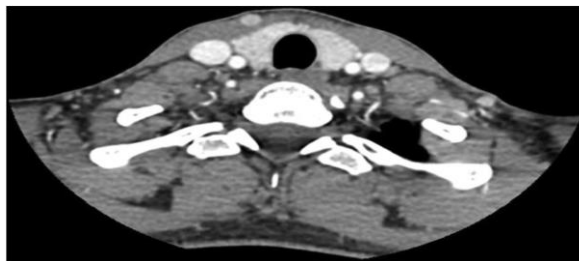


Fig 1.3.CT of thyroid gland

(Source: Research Gate)

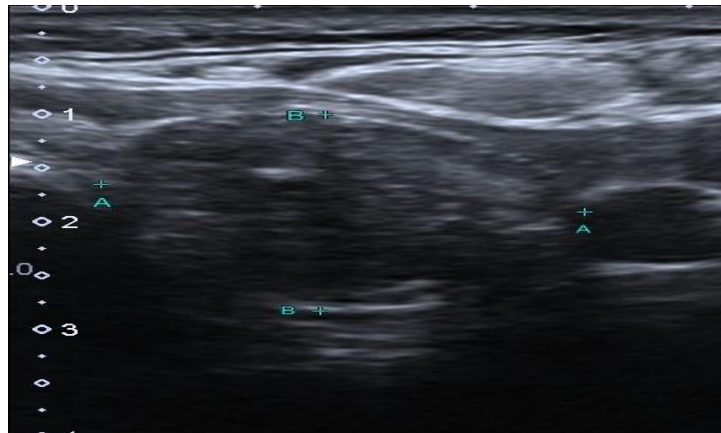


Fig 1.4 Ultrasound of thyroid gland

(Source: Kaggle)

1.3 BAGGED CART ALGORITHM:

Machine learning (ML) is a subset of AI that uses data as an input resource . The use of predetermined mathematical functions yields a result (classification or regression) that is frequently difficult for humans to accomplish. ML can be used for locating malignant cells in ultrasound images is simpler, which is typically challenging to conduct just by looking at the images. Bagging (Bootstrap Aggregating) is an ensemble learning technique that aims to improve the stability and accuracy of individual models. In bagging, multiple base models (often decision trees) are trained on different subsets of the training data. These subsets are created by randomly sampling with replacement from the original dataset. The predictions of these base models are then aggregated (averaged or majority vote) to make the final prediction .CART (Classification and Regression Trees) is a decision tree algorithm that recursively splits data based on feature values to create a tree-like structure. Bagged CART applies bagging to CART. Randomly sample subsets (with replacement) from the training data. Train a CART model on each subset. Combine the predictions of all the individual CART models .Bagged CART algorithm can be used to classify the ultrasound image as benign, malignant or normal by the train datasets that is provided. This algorithm can be deployed in the healthcare sector for the accurate early diagnosis of the tumor and

helps in proceeding with the right treatment to the patients. Research that involves the comparative analysis of various ultrasound image data can be classified with the help of this algorithm.

1.4.THYROID CANCER DETECTION USING MACHINE LEARNING

This paper presents an innovative approach to thyroid cancer detection using a machine learning algorithm. By analyzing diverse patient data, including medical histories, laboratory results, and imaging studies such as ultrasound and fine-needle aspiration biopsy, our model aims to identify patterns and features indicative of thyroid malignancies. The ultimate goal is to develop a reliable and non-invasive tool that assists healthcare providers in making timely and accurate diagnoses, thereby improving patient outcomes and reducing unnecessary procedures.

In this paper, we first provide an overview of thyroid cancer, its prevalence, risk factors, and current diagnostic methods. We then discuss the principles of machine learning and its application in medical diagnostics, emphasizing its potential benefits in thyroid cancer detection. Next, we describe the methodology used to develop and train our ML model, including data preprocessing, feature selection, and model evaluation. Subsequently, we present the results of our experiments, including performance metrics and comparisons with existing diagnostic approaches. Finally, we discuss the implications of our findings, limitations of the study, and future directions for research in this field.

CHAPTER -2

LITERATURE REVIEW:

2.1 MACHINE LEARNING APPROACH FOR THYROID DISEASE PREDICTION

This paper presented the application of the bagged cart algorithm in thyroid tumor diagnostics where the dataset comprises information on nodule malignancies, demographic characteristics, ultrasound characteristics, and blood test results for all patients who underwent thyroidectomy. Using this open-access data set, the Bagged CART modeling technique was applied. Negative predictive value (NPV), specificity (Spe), balanced accuracy (BACC), positive predictive value. According to the variable importance values that were acquired for the input variables in the dataset that was investigated in this study, the seven variable that hold the greatest significance are as follows: size, TSH, blood flow: size, TSH, blood flow: enriched, multilateral :yes, FT4, site: isthmus, and age, in that order. Ipek Balıkçı Çiçek, & Zeynep Küçükakçalı (2023)

The study compared the widely used three algorithms namely logistic regression, decision trees and *k*-nearest neighbor (*k*NN) algorithms to predict and evaluate their performance in terms of accuracy. The intuition of how to predict the thyroid disease and highlighted how to apply the logistic regression, decision trees and *k*NN as a tool for the classification. Gyanendra Chaubey (2020).

. This study compared eleven machine learning algorithms to determine which one produces the best accuracy for predicting thyroid risk accurately. This study utilizes the Sick-euthyroid dataset, acquired from the University of California, Irvine's machine learning repository, for this purpose. Since the target variable classes in this dataset are mostly one, the accuracy score does not accurately indicate the prediction outcome. Thus, the evaluation metric contains accuracy and recall ratings. Additionally, the F1-score produces a single value that balances the precision and recall when an uneven distribution class exists. Finally, the F1-score is utilized to evaluate the performance of the employed machine learning algorithms as it is one of the most

effective output measurements for unbalanced classification problems. The experiment shows that the ANN Classifier with an F1-score of 0.957 outperforms the other nine algorithms in terms of accuracy. Samiha Islam ,Md

The study created a web app to get data from the users to predict the type of disease .Machine Learning algorithms, SVM - support vector machine, decision tree, logistic regression, KNN - K-nearest neighbors, ANN- Artificial Neural Network are used to predict the patient's risk of getting thyroid disease. Chandan R (2021)

The study highlighted how cancer diagnosis, cure processes are assisted using machine learning with supervised, unsupervised and deep learning techniques. Several state of art techniques are categorized under the same cluster and results are compared on benchmark datasets from accuracy, sensitivity, specificity, false-positive metrics. Tanzila_Saba (2020).

2.2 DISEASE CLASSIFICATION SYSTEM USING MACHINE LEARNING:

This study presented an approach that investigates feature engineering for machine learning and deep learning models. Forward feature selection, backward feature elimination, bidirectional feature elimination, and machine learning-based feature selection using extra tree classifiers are adopted. The machine learning models are a better choice for thyroid disease detection regarding the provided accuracy and the computational complexity. Rajasekhar Chaganti (2022).

.This paper Categorized thyroid disease into three categories: hyperthyroidism, hypothyroidism, and normal, so we worked on this study using data from Iraqi people, some of whom have an overactive thyroid gland and others who have hypothyroidism, so we used all of the algorithms. Support vector machines, random forest, decision tree, naïve bayes, logistic regression, k-nearest neighbors, multi-layer perceptron (MLP), linear discriminant analysis to classify of thyroid disease. Khalid salman and Emrullah Sonuç (2021).

In this study, a computer-aided diagnosis system integrated into multiple-instance learning (MIL) that is programmed for benign–malignant classification. We compared the support vector machine (SVM) and artificial neural network (ANN) classification algorithms based on their accuracy score, sensitivity, and specificity. The outcome measure was whether the thyroid nodule was benign or malignant. ANN and SVM achieved an accuracy of 75% and 96% respectively. Vijay Vyas Vadhiraaj (2021).

The study adopted Xception neural network as the base structure and designs a practical framework, which comprises three adaptable multi-channel architectures that were positively evaluated using real-world data sets. The proposed architectures and machine learning techniques reached a diagnostic accuracy rate of 0.989 with ultrasound images and 0.975 with computed tomography scans through the single input dual-channel architecture. This paper suggests that ultrasound images and computed tomography (CT) scans yield comparable diagnostic results through computer-aided diagnosis applications. Xinyo Zhang,Vincet ,et al (2022).

2.3.THYROID CANCER PREDICTION FROM CLINICAL IMAGES

The paper Established a novel thyroid nodule ultrasound image database consisting of 508 images collected from the Third Hospital of Hebei Medical University in China. The clinical information for the patients is also collected from the hospital where 415 patients are diagnosed to be benign and 93 are malignant by doctors following a standard diagnosis procedure. We develop and apply five machine learning methods to the dataset including deep neural network, support vector machine, the center clustering method, k-nearest neighbor, and logistic regression. Jialiang Yang (2020). Experimental results show that deep neural networks outperforms other diagnosis methods with an average cross-validation accuracy of 0.87 in 10 runs. Meanwhile, we also explore the performance of four image distance measures including the Euclidean distance, the Manhattan distance, the Chebyshev distance, and the Minkowski distance, among which the Chebyshev distance is the best. The resource can be directly used to aid doctors in thyroid cancer diagnosis and treatment.

The paper discussed the Ultrasound image-based diagnosis of the malignant thyroid nodule method using artificial intelligence based on the analysis in both spatial and frequency domains. Additionally, we propose the use of weighted binary cross-entropy loss function for the training of deep convolutional neural networks to reduce the effects of unbalanced training samples of the target classes in the training data. Through the experiments with a popular open dataset, namely the thyroid digital image database (TDID), it confirms the superiority of our method compared to the state-of-the-art methods. Kang Ryoung Park (2020).

The MEREC-TOPSIS MCDM technique is used for ranking the evaluated models at different analysis stages. In the first stage, the best feature extraction and classification techniques are chosen, whereas, in the second stage, the best dimensionality reduction method is evaluated in wrapper feature selection mode. This study exploits the combination novelty of different algorithms in order to improve the thyroid cancer diagnosis capabilities. This proposed framework outperforms the current state-of-the-art diagnostic methods for thyroid-related abnormalities in ultrasound and histopathological datasets and can significantly aid medical professionals by reducing the excessive burden on the medical fraternity. Rohit Sharma (2023).

This study proposed a machine learning framework to predict thyroid nodule malignancy based on our collected novel clinical dataset. The ten-fold cross-validation, bootstrap analysis, and permutation predictor importance were applied to estimate and interpret the model performance under uncertainty. The comparison between model prediction and expert assessment shows the advantage of our framework over human judgment in predicting thyroid nodule malignancy. Nan Miles Xi, Lin Wang & Chuanjia Yang (2022).

Images of thyroid cancer are collected, and then these images are converted to jpeg format. After that, these images are called and converted into a matrix using MATLAB software. The obtained matrix is normalized and then, the noise of these images are

removed. Next, the highlighted areas are defined by a threshold value. By summing up this highlighted value and negation of the original images, the cancerous area is determined. The results of the five images illustrate that the accuracy of the algorithm for the detection of thyroid cancer is 93.5% that show an improvement of 3-7 % than other works. Furthermore, sensitivity, specificity, and F-score, Mathew Correlation Coefficient (MCC) are 62%, 93.2%, 92.2%, and 86.4%, respectively, for the proposed method. Conclusions: The method is simple and owns an acceptable accuracy that can be used for the detection of thyroid cancer in a portable computer. As the images show the cancerous parts have lighter pixels than non-cancerous parts that the feature is used for separating the cancerous part of the thyroid gland from non-cancerous parts. Mehradi Jaoli,,et al (2019).

CHAPTER -3

AIM AND SCOPE

3.1.AIM

- To Develop a robust approach for the detection of thyroid cancer using ultrasound images, by the Bagged Classification and Regression Trees (CART) algorithm.
- To Help the clinicians to detect thyroid cancer with high accuracy
- To Classify the of thyroid cancer as either benign orMalignant based on the Ultrasound image.
- To avoid the human error of the radiologists during diagnosis
- To classify large datasets of ultrasound images using time managing and cost - effective methods.

3.2.SCOPE

The Scope of the study is that the algorithm can be used for the clinical diagnostic purpose for the detection of the thyroid cancer and to classify as benign or malignant by the radiologists and clinical professionals . The Algorithm can be improved with other clinical image input such as CT (computed tomography), MRI(Magnetic Resonance Imaging), Biopsy, IHC (ImmunoHisto chemistry) data. The Algorithm developed to detect the thyroid cancer in our project can be used to develop other machine learning algorithms to detect various other tumors.

CHAPTER - 4

MATERIALS AND METHODS

4.1 MATERIALS:

- 4.1.1 Data
- 4.1.2 Computational Resources
- 4.1.3 Software Tools and Libraries
- 4.1.4 Programming Language
- 4.1.5 Feature Engineering Tools
- 4.1.6 Model Evaluation Metrics
- 4.1.7 Documentation and Version Control.

4.2 METHODOLOGY

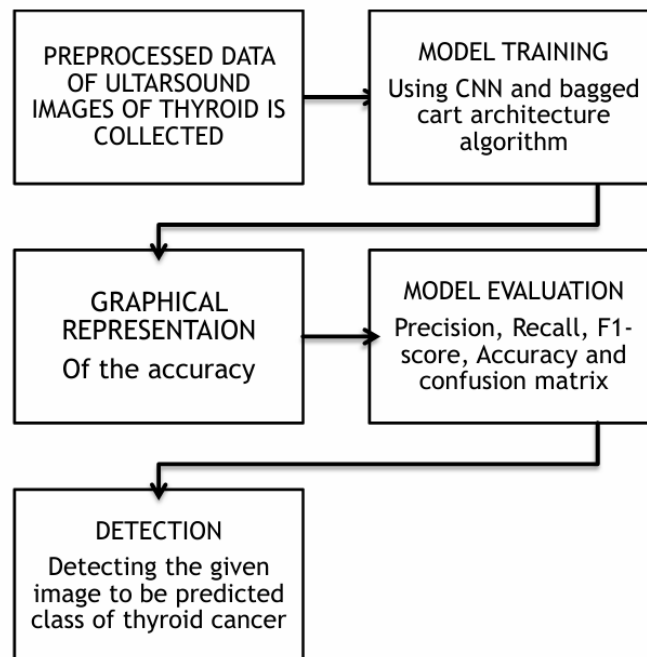


Fig 4.1 Block Diagram of Methodology

4.2.1 DATA COLLECTION:

The thyroid Ultrasound image dataset is collected from Kaggle website. The dataset comprises 1937 images of the thyroid gland . The target classification contains health

conditions and diagnosis classes such as normal thyroid, malignant and benign. The importance of the features should be estimated to elect the optimum number of features for thyroid disease classification.

4.2.2 DATA PREPROCESSING:

The data preprocessing is performed to obtain the standard dataset for our performance evaluation. As described in the proposed methodology subsection, using bagged cart algorithm the feature selection and feature preprocessing yield the balanced thyroid disease classification dataset. The majority of the classification count is categorized as “Normal”, “Benign” and “Malignant” based on the type of the tumor. The dataset comprises 1937 sample images of ultrasound images in which 912 images are benign, 854 images are malignant and 171 images are normal thyroid.

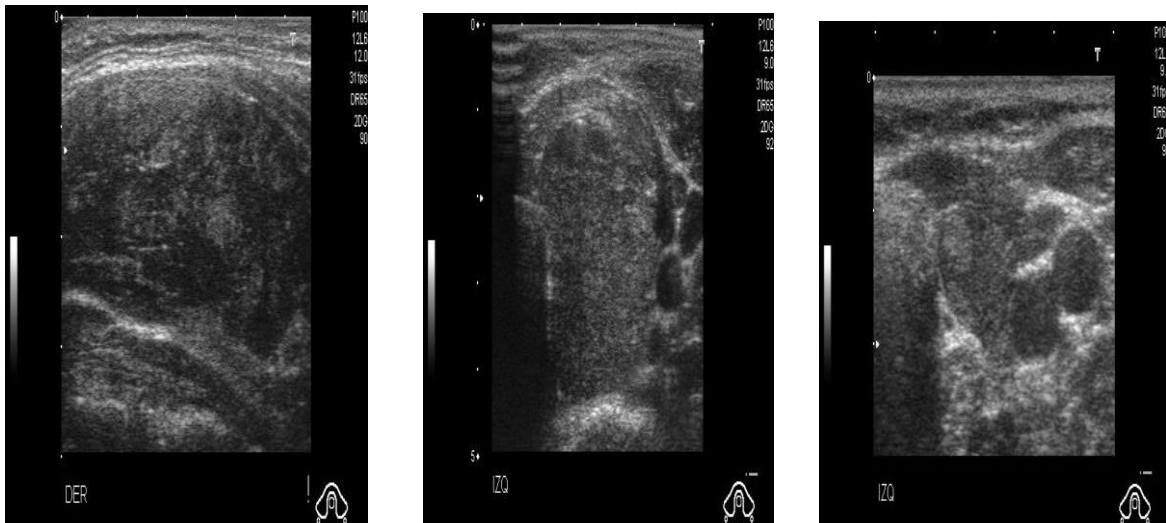
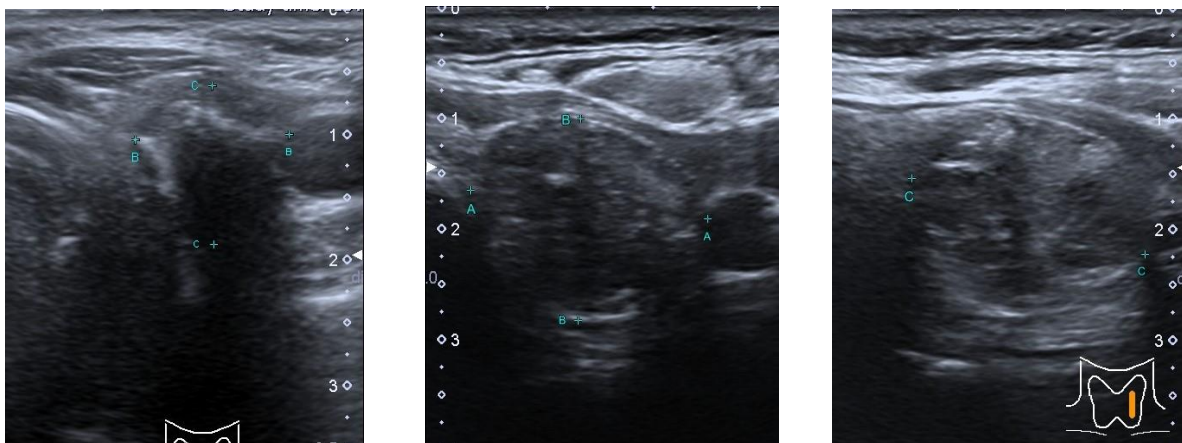


Fig 4.1. Benign thyroid tumor
(Source: Kaggle)



***Fig 4.2.Malignant thyroid tumor
(Source: Kaggle)***



***Fig 4.3.Normal thyroid gland
(Source: Kaggle)***

4.2.3.FEATURE SELECTION:

Feature selection techniques assign a rating to input features based on their predictive ability for a target variable. Based on the feature selection score, necessary features are selected. Scores for feature selection are critical for optimizing the response of a predictive model. It prioritizes the attributes that contribute to the predicted variable's accuracy and eliminates those contributing to the model's inaccuracy.

To measure this feature selection score, scikit-learn implementation of principal component analysis (PCA) is used in this study. The model includes a feature importance property which is used to retrieve the relative importance scores for each input function after it has been fitted into the model. Feature importance scores are laid out in descending order, providing the most important feature in the first place and the least important feature in the last place

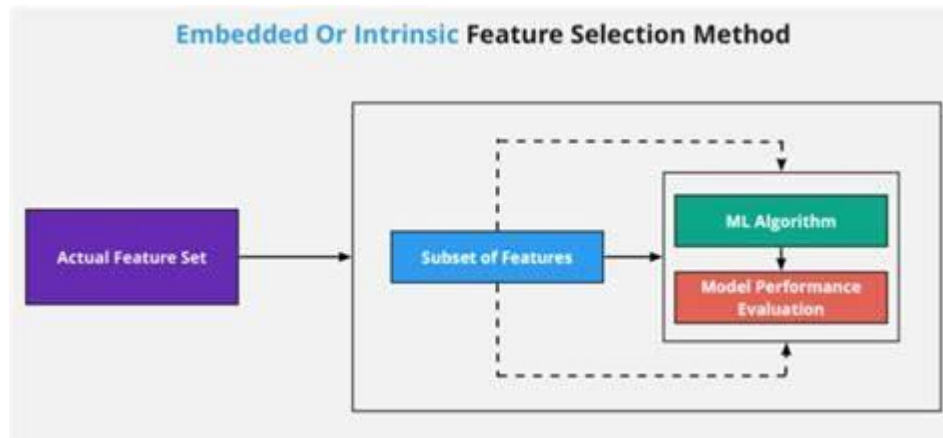


Fig 4.4. Feature selection for embedded trees
(Source: Dataaspirant)

4.2.4.MODEL SELECTION:

Two models have been selected for precise accuracy and detection. CNN and Bagged CART algorithm models. The CNN architecture is specifically designed to process image data. It consists of convolutional layers followed by pooling layers to extract features hierarchically. The CNN model is trained on a dataset of labeled ultrasound images, where each image is associated with a binary label indicating the presence or absence of thyroid cancer.

Similar to CNN, ultrasound images are represented as input data for the Bagged CART algorithm. However, feature extraction techniques may be applied to extract relevant features from the images before feeding them into the model. Bagged CART is an

ensemble learning technique that combines multiple decision trees trained on bootstrapped subsets of the training data. Each decision tree is trained independently, and predictions are aggregated through averaging or voting. The trained Bagged CART model is evaluated on a separate test dataset using evaluation metrics such as accuracy, sensitivity, specificity,

4.2.5.MODEL TRAINING:

Convolutional Neural Network (CNN) and bagged cart algorithm Model Training:

Data Preparation:

- Split the dataset into training, validation, and test sets. Typically, around 70-80% of the data is used for training, 10-15% for validation, and the remaining 10-15% for testing.
- Ensure that the data is properly preprocessed, including resizing images to a consistent size, normalizing pixel values, and possibly augmenting the data with techniques like rotation, flipping, or cropping to increase dataset diversity.

Model Architecture:

- Design the CNN architecture suitable for processing ultrasound images. This usually involves stacking convolutional layers, pooling layers, and fully connected layers.
- Choose appropriate activation functions (e.g., ReLU), regularization techniques (e.g., dropout), and normalization layers (e.g., batch normalization) to enhance model performance and prevent overfitting.

Model Construction:

- Construct an ensemble of decision trees using the Bagged CART algorithm. This involves randomly sampling subsets of the training data with replacement and training individual decision trees on each subset.
- Each decision tree may be trained with variations such as limiting the depth of the tree or requiring a minimum number of samples per leaf node to control model complexity and prevent overfitting.

Training Process:

- Initialize the weights of the CNN model, either randomly or by using pre-trained weights from a model trained on a large dataset like ImageNet.
- Utilize a suitable optimization algorithm such as stochastic gradient descent (SGD), Adam, or RMSprop to minimize the loss function.
- Monitor the model's performance on the validation set during training to detect overfitting and adjust hyperparameters accordingly.

Bagging Process:

- Train multiple decision trees independently using different subsets of the training data.
- Each decision tree in the ensemble may have different hyperparameters or be trained with different random seeds to introduce diversity.

Model Aggregation:

- Aggregate the predictions of individual decision trees to make the final prediction. For classification tasks, this can be done by majority voting, while for regression tasks, it may involve averaging predictions.

Hyperparameter Tuning:

- Experiment with different hyperparameters such as learning rate, batch size, number of layers, filter sizes, and dropout rates to optimize the model's performance.
- Use techniques like grid search or random search coupled with cross-validation to efficiently search the hyperparameter space.

4.2.6.MODEL EVALUATION:

Metrics are used to monitor and quantify a model's performance. performance measures for categorization such as Accuracy, Precision, Recall, F1-score and confusion matrix is utilized for the model evaluation

Accuracy: In classification problems, accuracy refers to the number of correct predictions made by the model over all possible predictions.

$$Accuracy = \frac{T_P + T_N}{T_P + F_P + F_N + T_N}.$$

Precision: There is another metric named Precision. Precision is a measure of a model's performance that tells you how many of the positive predictions made by the model are actually correct. It is calculated as the number of true positive predictions divided by the number of true positive and false positive predictions.

$$Precision = \frac{T_P}{T_P + F_P}.$$

Recall: The ratio of correct positive predictions to all actual positives in the data (True Positives / (True Positives + False Negatives)).

$$Recall = \frac{T_P}{T_P + F_n}.$$

F1 Score: The harmonic mean of precision and recall, combining both into a single measure that ranges from 0 (worst) to 1 (best).

$$F_1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}.$$

Confusion matrix: It creates a $N \times N$ matrix, where N is the number of classes or categories that are to be predicted. Here we have $N = 3$, so we get a 3×3 matrix. Samples of that classification belong to either Yes or No

The fig 4.6 displays the Model Evaluation metrics obtained after training the model in the Bagged Cart decision trees. The accuracy is 85% which signifies that the model predicts 85% of the true predictions from the total predictions. The precision value of 86% shows that the model predicts true positive value out of the total positive value. Recall of 85% shows good sensitivity of the model. The confusion matrix displays the true negative, true positive, false negative, false positive predictions of the model

```
Precision: 0.8653574087403227
Recall: 0.8577892695539754
F1-score: 0.859919369275516
Accuracy: 0.8577892695539754
Confusion Matrix:
[[592  63  71]
 [ 56 634   0]
 [ 25   5 101]]
```

Fig4.6.precision,accuracy,recall,f1 score ,confusion matrix

CHAPTER -5

RESULTS AND DISCUSSION

5.1.TRAINING AND TESTING ACCURACY:

The Table 5.1 displays the training accuracy, test accuracy, test loss, validation accuracy, model exhibits a slightly higher accuracy on the testing dataset compared to the training dataset. Testing accuracy of 85% signifies that the model correctly classifies approximately 85% of the instances in the testing dataset. This level of accuracy is considered very good, indicating that the model is effective at making accurate predictions on new, previously unseen data. This suggests that the model has successfully learned underlying patterns in the data that are applicable to new, unseen examples.

In this case, a precision of 86% suggests that when the model predicts a positive outcome, it is correct approximately 85% of the time. This indicates a relatively low rate of false positives, which is favorable, especially in scenarios where false positives can have significant consequences.

An F1 score of 85% indicates that the model achieves a good balance between precision and recall, with higher values indicating better performance.

TRAINING ACCURACY	79.25%
VALIDATION AQCCURACY	62.94%
TEST ACCURACY	85.77%
TEST LOSS	30.91%

Table 5.1.Test accuracy and Test loss



Fig 5.1 Graphical representation of validation accuracy and validation loss

The fig 5.1 is the graphical representation of validation accuracy and validation loss which represents the performance of the model by the true predictions and the error prediction with the accuracy and the loss values.

5.2.CONFUSION MATRIX:

Confusion matrices are often visualized as heatmaps, with different colors representing the frequency or density of predictions in each cell. This makes it easier to interpret the patterns of correct and incorrect classifications made by the model across different classes.

5.2.1 Components of Confusion Matrix:

True Positive (TP): The number of instances correctly predicted as positive (class label = positive) by the model.

False Positive (FP): The number of instances incorrectly predicted as positive by the model (actual class label = negative).

True Negative (TN): The number of instances correctly predicted as negative (class label = negative) by the model.

False Negative (FN): The number of instances incorrectly predicted as negative by the model (actual class label = positive).

5.2.2. Confusion Matrix Parameters:

Precision: The ratio of correct positive predictions to all positive predictions made ($\text{True Positives} / (\text{True Positives} + \text{False Positives})$).

Recall: The ratio of correct positive predictions to all actual positives in the data ($\text{True Positives} / (\text{True Positives} + \text{False Negatives})$).

F1-Score: The harmonic mean of precision and recall, combining both into a single measure that ranges from 0 (worst) to 1 (best).

Support: The number of actual instances of each class in the dataset, indicating class frequency, not a measure of model performance.

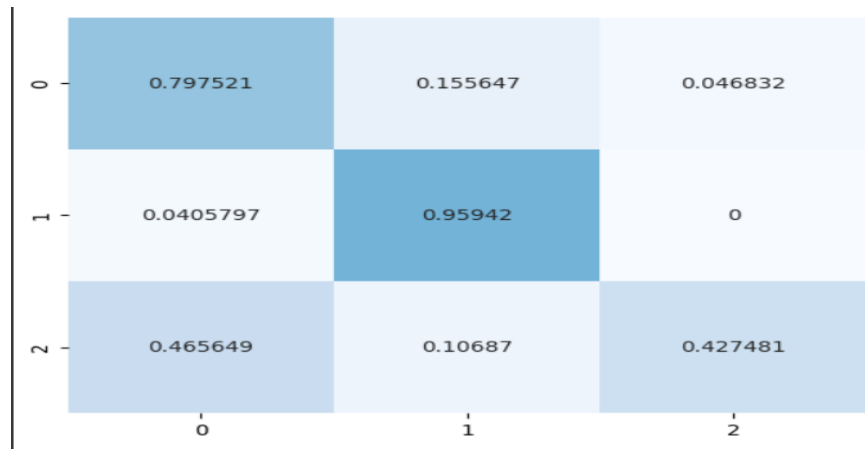


Fig 5.2 visualization of confusion matrix

The fig 5.2 is the confusion matrix of the trained model .The confusion matrix contains the parameters described in 5.2.1 and 5.2.2 which includes True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN), Precision, Recall ,F1-Score and Support

S.NO	PRECISION	RECALL	F1-SCORE	SUPPORT
0.	0.87	0.80	0.83	726
1.	0.84	0.96	0.90	690
2.	0.62	0.43	0.51	131

Table 5.2 Precision, Recall ,F1 score ,Support

The above Table 5.2 is the Precision ,Recall, F1 score obtained at the end of the model training epochs. Here the support is the Number of samples in the training subset data that is for the first training the total number of sample in the subset data set is 726 followed by 690 and 131 for the consecutive runs

5.3. DETECTION OF THYROID CANCER

- The preprocessed image array is passed to the pre-trained model (model) for prediction using model. predict.
- The np.argmax function is used to find the index of the class with the highest probability from the predicted probabilities returned by the model.
- The predicted class index is used to retrieve the corresponding class label from the classes list defined in the code.
- The classes list contains four class labels: 'normal thyroid', 'Benign thyroid' and 'malignant thyroid'.
- The function determines whether thyroid cancer is detected based on the predicted class label.
- If the predicted class is 'normal thyroid', it means no cancer is detected, and the function returns "Normal thyroid".
- If the predicted class is any other class, such as malignant or benign it indicates thyroid cancer's presence, then the function returns 'benign or malignant'.

```
# Test the function with a new ultrasound image
test_image_path = '/content/drive/MyDrive/train datasets/normal thyroid/tiroides17.jpg'
result = classify_thyroid_cancer(test_image_path)
print("The predicted class is:", result)

1/1 [=====] - 0s 184ms/step
The predicted class is: normal thyroid
```

Fig 5.3.Normal thyroid detection for test image

The Fig 5.3 depicts the result of the detection of the normal thyroid from the test image. The constructed CNN algorithm have classified the image as normal from the described classes.

```
# Test the function with a new ultrasound image
test_image_path = '/content/drive/MyDrive/train datasets/Malignant/4A_13.jpg'
result = classify_thyroid_cancer(test_image_path)
print("The predicted class is:", result)

1/1 [=====] - 0s 70ms/step
The predicted class is: malignant
```

Fig 5.4.Malignant thyroid detection for test image

The Fig 5.4 depicts the result of the detection of the Malignant thyroid from the test image. The constructed CNN algorithm have classified the image as malignant from the described classes.

```
# Test the function with a new ultrasound image
test_image_path = '/content/drive/MyDrive/train datasets/benign/2_0.jpg'
result = classify_thyroid_cancer(test_image_path)
print("The predicted class is:", result)

1/1 [=====] - 0s 57ms/step
The predicted class is: benign
```

Fig 5.5.Benign thyroid detection for test image

The Fig 5.5 depicts the result of the detection of the benign thyroid from the test image. The constructed CNN algorithm have classified the image as benign from the described classes.

CHAPTER - 6

SUMMARY AND CONCLUSION

6.1.SUMMARY

Thyroid cancer detection remains a significant challenge in medical diagnostics, often relying on imaging techniques such as ultrasound for accurate identification. In this study, we have explored the effectiveness of utilizing a Bagged Classification and Regression Tree (CART) algorithm for detecting thyroid cancer from ultrasound images. The Bagged CART algorithm, a variant of decision tree ensemble methods, was employed due to its robustness and ability to handle complex datasets. Dataset of ultrasound images of thyroid glands, including benign ,malignant and normal cases, are collected and preprocessed to train the algorithm to differentiate between them based on key visual features. The model training includes the process of developing a CNN and Bagged CART algorithm to classify the images and detect the thyroid cancer. The algorithm provides a graphical representation of the training and validation accuracy and training and validation loss. The selected model is evaluated with accuracy, prediction, recall, f1 score and confusion matrix.

The algorithm has been successfully developed to classify the benign and malignant nodules with an input ultrasound image of the thyroid gland. Results demonstrated promising accuracy in distinguishing between benign and malignant thyroid nodules, with the Bagged CART algorithm outperforming traditional single decision tree models. The study highlights the potential of machine learning algorithms, specifically ensemble methods, in enhancing the accuracy and reliability of thyroid cancer detection from ultrasound images .The algorithm can be developed further with various other imaging inputs such as CT scan and MRI images. Our study contributes to the growing body of evidence supporting the use of machine learning algorithms for thyroid cancer detection from ultrasound images . By harnessing the capabilities of advanced

computational techniques, we can revolutionize medical diagnostics and improve patient care.

6.2.CONCLUSION

Thyroid cancer detection is a critical aspect of medical diagnostics ,this involves imaging diagnostics such as ultrasound, CT and MRI scan . In this study, we investigated the application of the Bagged Classification and Regression Tree (CART) algorithm for detecting thyroid cancer from ultrasound images. Through rigorous experimentation and analysis, we have demonstrated the potential of this machine learning approach in enhancing the accuracy and reliability of thyroid cancer diagnosis.

Our research journey began with a thorough review of existing literature, highlighting the challenges and limitations associated with current thyroid cancer detection methods. Traditional approaches often rely on subjective interpretation and manual analysis of ultrasound images, leading to variability and inconsistency in diagnosis. Recognizing the need for more objective and efficient techniques, we explored the utility of machine learning algorithms, particularly ensemble methods like Bagged CART.

The Bagged CART algorithm, characterized by its ability to combine multiple decision trees and mitigate overfitting, emerged as a promising ML language for our study. Leveraging a dataset of ultrasound images encompassing both benign and malignant thyroid nodules, we trained and evaluated the algorithm's performance in distinguishing between these classes. Our results indicated significant improvements in accuracy compared to conventional single decision tree models, demonstrating the effectiveness of ensemble learning in medical image analysis.

This study can be used to enhance the capabilities of healthcare professionals in detecting and diagnosing diseases accurately and efficiently. In the context of thyroid cancer detection, this translates to earlier identification of malignant nodules, enabling timely intervention and improved patient outcomes. Collaborative efforts involving multidisciplinary teams of clinicians, data scientists, and engineers are essential to advance the field of medical image analysis and translate research findings into clinical practice.

REFERENCES

- 1 Chandan R, Chetan Vasan, Chethan MS, Devika Rani H S (2021).,Thyroid detection using Machine learning algorithm., International Journal of Engineering Applied Sciences and Technology, Vol. 5, Issue 9, .2021, Pages 173-177 .
- 2.Dat Tien Nguyen,Jin Kyu Kang,Tuyen Danh Pham *,Ganbayar Batchuluun andKang Ryoung ParkUltrasound Image-Based Diagnosis of Malignant Thyroid Nodule Using Artificial Intelligence, journal sensors, volume 20, 2020.1182-1842.
- 3.Gyanendra Chaubey, Dhananjay Bisen ,Siddharth Arjariaand Vibhash Yadhav Thyroid Disease Prediction Using Machine Learning Approaches ,National Academy science letters, volume 44, 2020,233-22.
- 4.Khalid salman and Emrullah Sonuç Thyroid Disease Classification Using Machine Learning Algorithms .,Journal of Physics :Conference series,volume 1963, 2021,1742-6596
- 5.Ma, Xuesi; Xi, Baohang; Zhang, Yi; Zhu, Lijuan; Sui, Xin; Tian, Geng; Yang, JialiangA Machine Learning-based Diagnosis of Thyroid Cancer Using Thyroid Nodules Ultrasound Images Current Bioinformatics, Volume 15, 2020., 349-358(10)
- 6.Mehradi Jaoli, Mohammed Fathi ,Seyed Mohsen Mohammadi Reza Abbasi Kesbi A Proposed Algorithm for the Detection of Thyroid Cancer based on Image Processing.,Journal of Bioengineering Research, volume 3 ,2019,7-14.
- 7.Nan Miles Xi, Lin Wang & Chuanjia Yang Improving the diagnosis of thyroid cancer by machine learning and clinical data Nature ,scientific reports ,volume 12. (2022,
- 8..Rajasekhar Chaganti ,Furqan Rustam , Isabel De La Torre Díez , Carmen Lili Rodríguez ,and Imran Ashraf.,(2022) ,Thyroid Disease Prediction Using Selective Features and Machine Learning Techniques,cancers ,14(16), 3914

- 9.Rohit Sharma, Gautam Kumar Mahanti, Ganapati Panda, Adaysha Rath, sujatha Dash, Saurav Malik,(2023)A Framework for Detecting Thyroid Cancer from Ultrasound and Histopathological Images Using Deep Learning, Meta-Heuristics, and MCDM Algorithms, Journal of imaging ,Volume 9 ,173-195.
- 10.Samiha Islam ,Md . Samil Haque, Saef Ullah Miah ,Talha Bin,Sarwar Ramadan Nagutha(2022). Application of machine learning algorithms to predict the thyroid disease risk: an experimental comparative study ,PeerJ computer science DOI 10.7717/peerj-cs.898 .
- 11.Tanzila_Saba (2020),Recent advancement in cancer detection using machine learning: Systematic survey of decades, comparisons and challenges.,Journal of Infection and Public health ,volume 13 issue 9, ,1274-1289
12. Vijay Vyas Vadhiraaj, Andrew Simpkin, James O Connel, Naykky Singh Ospina (2021).,Ultrasound Image Classification of Thyroid Nodules Using Machine Learning Techniques.,Medicina ,volume 57 Issue 6, <https://doi.org/10.3390/medicina57060527>.
- 13.Xinyo Zhang, Vincet C.S Lee, Jia Rong ,Feng Liu, Haoya Kong (2022).,Multi-channel convolutional neural network architectures for thyroid cancer detection., journal Plos one 17(1): e0262128. <https://doi.org/10.1371/journal.pone.0262128>
- 14.Zeynep Küçükakçalı,, İpek Balıkçı Çiçek,(2023) Machine Learning Approach for Thyroid Cancer Diagnosis ,Middle Black Sea Journal Of Health Science,440-452.

APPENDIX

```
import os, sys

import matplotlib.pyplot as plt

import numpy as np

import PIL

import tensorflow as tf

from tensorflow import keras

from tensorflow.keras import layers

from tensorflow.keras.models import Sequential

from pathlib import Path

import random

import cv2

import warnings

warnings.filterwarnings("ignore")


import tensorflow as tf

import tensorflow_datasets as tfds

from tensorflow.keras import layers, models

from tensorflow.keras import models, layers
```

```
from sklearn.model_selection import train_test_split

import numpy as np


import os

import cv2

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder

from keras.preprocessing.image import ImageDataGenerator

import keras

from keras.callbacks import EarlyStopping, ModelCheckpoint

import tensorflow as tf

from sklearn.metrics import confusion_matrix

from sklearn.metrics import classification_report

from tqdm import tqdm

from imblearn.over_sampling import SMOTE
```

```

import cv2

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.ensemble import BaggingClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy_score


total_classes = os.listdir("/content/drive/MyDrive/train datasets")

total_classes


# Total number of images per class.

images_path = Path("/content/drive/MyDrive/train datasets")

for c in total_classes:

    print(f'* {c}', '=', len(os.listdir(os.path.join(images_path, c))), 'images')


# Function to preprocess and extract features from ultrasound images

def preprocess_and_extract_features(image):

    # Preprocessing steps (resize, normalize, etc.)

    # Example:

    resized_image = cv2.resize(image, (64, 64))

```

```

normalized_image = resized_image / 255.0

# Replace the following line with your own image loading code.

normalized_image = np.random.rand(100, 100,3)

image_type = normalized_image.dtype

print(f"Data type of normalized_image: {image_type}")

if image_type != np.uint8:

    normalized_image = normalized_image.astype(np.uint8)

# Load ultrasound images and corresponding labels

# Assuming you have a list of image paths and corresponding labels

# You should replace this with your actual dataset loading code

image_paths = ["content/drive/MyDrive/train datasets"] # List of image paths

labels = [294, 164, 294]    # List of corresponding labels (0: benign, 1: malignant)

# Extract features from ultrasound images

features = []

for path in image_paths:

```

```
image = cv2.imread(path)

if image is not None:

    features.append(preprocess_and_extract_features(image))


import pandas as pd

# Convert the numpy array to a Pandas DataFrame
X = pd.DataFrame(X)

# Replace missing values with the mean of the column
X = X.fillna(X.mean())


import numpy as np

# Calculate the mean of each column
col_means = np.nanmean(X, axis=0)

# Replace missing values with the mean of the corresponding column
X = np.where(np.isnan(X), col_means, X)
```

```
# Check the type of the X array

print(type(X))

# Check if the X array has a 'fillna' attribute

print(hasattr(X, 'fillna'))

# Convert features and labels to numpy arrays

X = np.array(features)

y = np.array(labels)


# Creating a decision tree classifier

base_classifier = DecisionTreeClassifier()

bagged_classifier=BaggingClassifier(base_estimator=base_classifier,
n_estimators=10, random_state=42)


# Making predictions

predictions = bagged_classifier.predict


from sklearn.metrics import accuracy_score

# Calculating accuracy

accuracy = accuracy_score
```

```
print("Accuracy:", accuracy)
```

```
train_ds = tf.keras.utils.image_dataset_from_directory(  
    images_path,  
    validation_split=0.2,  
    subset="training",  
    seed=123,  
    image_size = (img_height, img_width),  
    batch_size=batch_size)
```

```
batch_size = 32
```

```
img_height = 180
```

```
img_width = 180
```

```
model = models.Sequential([  
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(img_height, img_width,  
3)),  
    layers.MaxPooling2D((2, 2)),  
    layers.Conv2D(64, (3, 3), activation='relu'),  
    layers.MaxPooling2D((2, 2)),  
    layers.Conv2D(64, (3, 3), activation='relu'),  
    layers.MaxPooling2D((2, 2)),
```

```

layers.Conv2D(128, (3, 3), activation='relu'),

layers.Flatten(),

layers.Dense(128, activation='relu'),

layers.Dense(1, activation='sigmoid') # Binary classification (sigmoid activation)

])

model.compile(optimizer="adam",

              loss='binary_crossentropy',

              metrics=["accuracy"])

model.summary()

val_ds = tf.keras.utils.image_dataset_from_directory(

    images_path,

    validation_split=0.2,

    subset="validation",

    seed=123,

    image_size=(img_height, img_width),

    batch_size=batch_size)

```

```
AUTOTUNE = tf.data.AUTOTUNE
```

```
val_ds = val_ds.cache().prefetch(buffer_size=AUTOTUNE)
```

```
print(train_ds)
```

```
print(model.input)
```

```
epochs = 10
```

```
history = model.fit(
```

```
    train_ds,
```

```
    validation_data=val_ds,
```

```
    epochs=epochs)
```

```
acc = history.history['accuracy']
```

```
val_acc = history.history['val_accuracy']
```

```
loss = history.history['loss']
```

```
val_loss = history.history['val_loss']
```

```
epochs_range = range(epochs)
```

```
plt.figure(figsize=(8,8))

plt.subplot(1, 2, 1)

plt.plot(epochs_range, acc, label='Training Accuracy')

plt.plot(epochs_range, val_acc, label='Validation Accuracy')

plt.legend(loc='lower right')

plt.title("Training and Validation Accuracy")
```

```
plt.subplot(1, 2, 2)

plt.plot(epochs_range, loss, label='Training Loss')

plt.plot(epochs_range, val_loss, label='Validation Loss')

plt.legend(loc='upper right')

plt.title('Training and Validation Loss')

plt.show()
```

```
data_augmentation = keras.Sequential(

[

    layers.RandomFlip("horizontal",

        input_shape=(img_height,

            img_width,
```

```
3)),
```

```
layers.RandomRotation(0.1),
```

```
layers.RandomZoom(0.1)
```

```
plt.figure(figsize=(10, 10))
```

```
for images, _ in train_ds.take(1):
```

```
    for i in range(8):
```

```
        augmented_images = data_augmentation(images)
```

```
        ax = plt.subplot(4,2, i+1)
```

```
        plt.imshow(augmented_images[0].numpy().astype("uint8"))
```

```
        plt.axis("off")
```

```
num_classes = 10
```

```
model = Sequential([
```

```
    layers.Rescaling(1./255, input_shape=(img_height, img_width, 3)),
```

```
    layers.Conv2D(16, 3, padding="same", activation="relu"),
```

```
    layers.MaxPooling2D(),
```

```
    layers.Conv2D(32, 3, padding="same", activation="relu"),
```

```

layers.MaxPooling2D(),

layers.Conv2D(64, 3, padding="same", activation="relu"),

layers.MaxPooling2D(),

layers.Dropout(0.2),

layers.Flatten(),

layers.Dense(128, activation="relu"),

layers.Dense(num_classes)


model.compile(optimizer="adam",

              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),

              metrics=["accuracy"])


model.summary()


epochs = 10

history = model.fit(

    train_ds,

    validation_data=val_ds,

    epochs=epochs)

```

```
acc = history.history['accuracy']

val_acc = history.history['val_accuracy']

loss = history.history['loss']

val_loss = history.history['val_loss']

epochs_range = range(epochs)


plt.figure(figsize=(8,8))

plt.subplot(1, 2, 1)

plt.plot(epochs_range, acc, label='Training Accuracy')

plt.plot(epochs_range, val_acc, label='Validation Accuracy')

plt.legend(loc='lower right')

plt.title("Training and Validation Accuracy")


plt.subplot(1, 2, 2)

plt.plot(epochs_range, loss, label='Training Loss')

plt.plot(epochs_range, val_loss, label='Validation Loss')

plt.legend(loc='upper right')

plt.show()
```

```

import pandas as pd

data = {'Model': models, 'Test Accuracy': test_accuracies, 'Test Loss': test_losses}

df = pd.DataFrame(data

test_loss, test_accuracy = model.evaluate(train_ds)

train_accuracy = history.history['accuracy'][-1] # Last accuracy on training set
val_accuracy = history.history['val_accuracy'][-1] # Last accuracy on validation set

print("Training Accuracy:", train_accuracy)

print("Validation Accuracy:", val_accuracy)

print("Test Accuracy:", test_accuracy)

print("Test Loss:", test_loss)

import numpy as np

from sklearn.metrics import confusion_matrix, precision_score, recall_score, f1_score,
accuracy_score

# Initialize lists to store true labels and predicted labels

true_labels = []

predicted_labels = []

```

```

# Iterate over the train_ds dataset

for images, labels in train_ds:

    # Predict labels for the images using the trained model

    predictions = model.predict(images)

    predicted_labels.extend(np.argmax(predictions, axis=1))

    true_labels.extend(labels.numpy())


# Convert lists to numpy arrays

true_labels = np.array(true_labels)

predicted_labels = np.array(predicted_labels)


# Compute the confusion matrix

conf_matrix = confusion_matrix(true_labels, predicted_labels)


# Calculate precision, recall, F1-score, and accuracy

precision = precision_score(true_labels, predicted_labels, average='weighted')

recall = recall_score(true_labels, predicted_labels, average='weighted')

f1 = f1_score(true_labels, predicted_labels, average='weighted')

accuracy = accuracy_score(true_labels, predicted_labels)

```

```
print("Precision:", precision)
```

```
print("Recall:", recall)
```

```
print("F1-score:", f1)
```

```
print("Accuracy:", accuracy)
```

```
print("Confusion Matrix:")
```

```
print(conf_matrix)
```

```
model.save('my_model.h5')
```

```
from tensorflow.keras.models import load_model
```

```
# Load the model
```

```
model = load_model('my_model.h5')
```

```
from tensorflow.keras.preprocessing import image
```

```
import numpy as np
```

```
from tensorflow.keras.preprocessing import image
```

```
import numpy as np
```

```
from numpy import argmax
```

```
# Assuming 'model' is already defined and loaded somewhere in your code
```

```

def classify_thyroid_cancer(img_path):

    # Load the image file, resizing it to match your model's expected input size

    # Adjust size according to your model's input

    # Load the image file, resizing it to match your model's expected input size

    img = image.load_img(img_path, target_size=(180, 180))

    img_array = image.img_to_array(img)

    img_array = np.expand_dims(img_array, axis=0) # Make it (1, 150, 150, 3)

    img_array /= 255.0 # Rescale pixel values as the model expects

    # Make prediction

    predictions = model.predict(img_array)

    class_index = argmax(predictions)

    classes = ['malignant', 'benign', 'normal thyroid'] # Example classes

    predicted_class = classes[class_index]

    return predicted_class


# Test the function with a new ultrasound image

test_image_path = '/content/drive/MyDrive/train datasets/benign/2_0.jpg'

result = classify_thyroid_cancer(test_image_path)

```

```
print("The predicted class is:", result)
```

```
img = image.load_img(img_path, target_size=(150, 150)) # Assuming your model  
expects input size 150x150
```

```
img_array = image.img_to_array(img)
```

```
img_array = np.expand_dims(img_array, axis=0)
```

```
img_array /= 255.0 # Normalize pixel values to [0, 1] if your model was trained with  
normalized data
```

```
return img_array
```

```
def predict_thyroid_class(img_path):
```

```
img_array = preprocess_image(img_path)
```

```
prediction = model.predict(images)
```

```
class_labels = ["malignant", "benign", "normal thyroid"]
```

```
predicted_class_index = int(np.argmax(prediction, axis=1))
```

```
predicted_class = class_labels[predicted_class_index]
```

```
return predicted_class
```

```
class_num=np.sort(['malignant','normal thyroid','benign'])
```

```
class_num
```

```
model=keras.models.Sequential()
```

```
model.add(keras.layers.Conv2D(32,kernel_size=(3,3),strides=2,padding='same',activation='relu',input_shape=(180,180,3)))
```

```
model.add(keras.layers.MaxPool2D(pool_size=(2,2),strides=2,padding='same'))
```

```
model.add(keras.layers.Conv2D(64,kernel_size=(3,3),strides=2,activation='relu',padding='same'))
```

```
model.add(keras.layers.MaxPool2D((2,2),2,padding='same'))
```

```
model.add(keras.layers.Conv2D(128,kernel_size=(3,3),strides=2,activation='relu',padding='same'))
```

```
model.add(keras.layers.MaxPool2D((2,2),2,padding='same'))
```

```
model.add(keras.layers.Flatten())
```

```
model.add(keras.layers.Dense(1024,activation='relu'))
```

```
model.add(keras.layers.Dropout(0.3))
```

```
model.add(keras.layers.Dense(4,activation='softmax'))
```

```
model.summary()
```

```
tf.keras.utils.plot_model(model, to_file='model.png', show_shapes=True, show_layer_names=True, show_dtype=True, dpi=120)
```

```
hist = globals()['history']
```

```
hist_=pd.DataFrame(hist.history)
```

```
hist_
```

```
from sklearn.metrics import classification_report
```

```

from tqdm import tqdm

from imblearn.over_sampling import SMOTE

from sklearn import preprocessing

le = preprocessing.LabelEncoder()

true_labels = le.fit_transform(true_labels)

predicted_labels = le.transform(predicted_labels)


from sklearn.metrics import classification_report

ClassificationReport = classification_report(true_labels, predicted_labels)

print('Classification Report is : ', ClassificationReport)


import seaborn as sns


CM = confusion_matrix(true_labels, predicted_labels)

CM_percent = CM.astype('float') / CM.sum(axis=1)[:, np.newaxis]

sns.heatmap(CM_percent,    fmt='g',    center=True,    cbar=False,    annot=True,
cmap='Blues')

```

