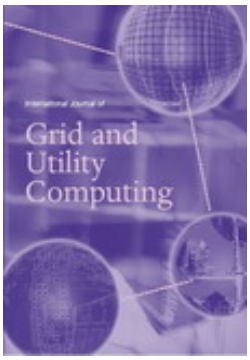


International Journal of Grid and Utility Computing

ISSN online: 1741-8488 - ISSN print: 1741-847X
https://www.inderscience.com/ijguc

Performance comparison of various machine learning classifiers using fusion of LBP, intensity and GLCM feature extraction techniques for thyroid nodules classification

Rajshree Srivastava, Pardeep Kumar

DOI: 10.1504/IJGUC.2023.10055313

Article History:

Received:	12 December 2022
Last revised:	11 February 2023
Accepted:	25 February 2023
Published online:	19 February 2024

Performance comparison of various machine learning classifiers using fusion of LBP, intensity and GLCM feature extraction techniques for thyroid nodules classification

Rajshree Srivastava and Pardeep Kumar*

Department of Computer Science and Engineering,
Jaypee University of Information Technology,
Waknaghat, Solan, Himachal Pradesh, India
Email: rajshree.srivastava27@gmail.com
Email: pardeepkumarkhokhar@gmail.com
*Corresponding author

Abstract: Machine Learning (ML) and feature extraction techniques have shown a great potential in medical imaging field. This work presents an effective approach for the identification and classification of thyroid nodules. In the proposed model, various features are extracted using Grey Level Co-occurrence Matrix (GLCM), Local Binary Pattern (LBP) and intensity-based matrix. These features are fed to various ML classifiers like K-Nearest Neighbour (KNN), Decision-Tree (DT), Artificial Neural Network (ANN), Naïve Bayes, Extreme Gradient Boosting (XGBoost), Random Forest (RF), Linear Regression (LR) and Support Vector Machine (SVM). From the result analysis, it can be observed that proposed Model-4 has performed better in comparison with the rest of seven proposed models with the reported literature. An improvement of 4% to 5% is seen in performance evaluation of model in comparison with reported literature.

Keywords: machine learning; LBP; GLCM; intensity; noise removal; feature extraction.

Reference to this paper should be made as follows: Srivastava, R. and Kumar, P. (2024) 'Performance comparison of various machine learning classifiers using fusion of LBP, intensity and GLCM feature extraction techniques for thyroid nodules classification', *Int. J. Grid and Utility Computing*, Vol. 15, No. 1, pp.84–96.

Biographical notes: Rajshree Srivastava is pursuing PhD degree in Computer Science and Engineering from Jaypee University of Information Technology, Waknaghat, District Solan, Himachal Pradesh, India. Her research interest includes machine learning, deep learning, healthcare informatics, etc.

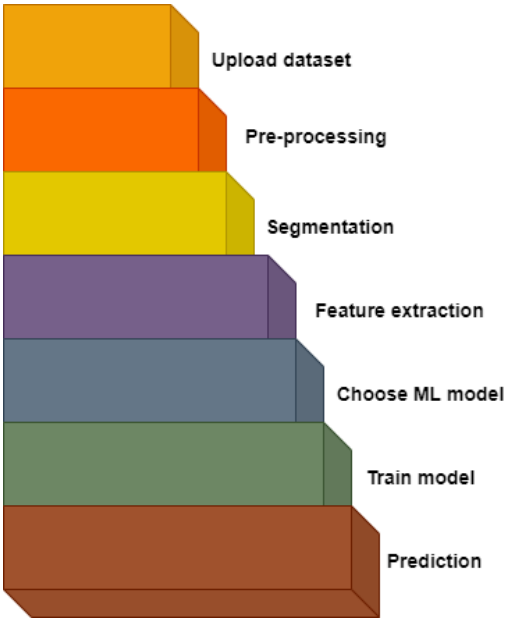
Pardeep Kumar is currently working in the Department of Computer Science & Engineering at Jaypee University of Information Technology (JUIT), Waknaghat and he has 15 plus years of extensive experience in academics. He is also serving as Senior Member of Association for Computing Machinery (ACM), Life Member of International Association of Engineers (IAENG) and IAENG Society of Computer Science and Society of Data Mining. He has published around 50 papers in peer reviewed journals and conferences of national and international repute with publishers like IEEE, ACM, Springer, Elsevier, etc. His area of interest includes machine learning, deep learning, healthcare informatics, etc.

1 Introduction

In recent years, the thyroid nodules are the most common nodular tumour in the adult population. The early diagnosis of this tumour is essential. There are many imaging modalities for screening thyroid nodules, but Ultrasonography (USG) is widely used as it is cost-effective and real-time (Meiburger et al., 2018). A thyroid nodule can be defined as a lump of nodes present in the thyroid region of the neck. These thyroid nodules can be benign or malignant. In most cases, these nodules when undergone through the USG found to be benign. Some of the characteristics of benign nodules are regular shape, while malignant nodules have irregular shapes, hypo-echogenicity, etc. The person with a high risk of malignant nodules is recommended for surgery, while medicines and regular follow-up are suggested for benign nodules. The traditional diagnostic method based on doctor's expert knowledge has one of the limitations. Sometimes a double screening scheme system is recommended in the hospitals by employing an additional expert, which results in time-consuming and extra expenditure (Jung et al., 2014). In medical research, a Computer-Aided Diagnosis (CAD) system is developed to diagnose disease, which has proved successful in many applications like lung cancer, brain tumour, breast

cancer, etc. (Cheng and Liu, 2018). Thyroid nodule formation mostly occurs when there is an excess production of thyroxine hormone in the body (Srivastava and Kumar, 2021a). Thyroid imaging reporting and data system (TI-RADS) has assigned some scores based on the characteristics of the thyroid USG images (Pedraza et al., 2015). The traditional methods mainly focus on selecting better hand-crafted features (Nguyen et al., 2019). Figure 1 shows the working of ML techniques. The steps followed by ML techniques for classification are: (i) uploading of data sets, (ii) pre-processing steps like removing missing data, RGB to grey scale conversion, resizing, (iii) segmentation, (iv) feature extraction, (v) apply or use ML algorithms like ANN, SVM, DT, RF, etc., (vi) train the model and (vii) prediction.

Figure 1 Working of ML techniques (see online version for colours)



Source: https://www.mygreatlearning.com/blog/what-is-machine-learning/

Traditional ML classifiers like SVM, DT, neural network (Abiodun et al., 2018), etc., along with feature extraction techniques give good results with the raw data. ML has progressed a lot from the last two decades. It has become a choice for developing practical software in the field of medical, robot control, computer vision, etc. ML in disease prediction has gained a significant attention due to the wide adaption of Computer-Based Technology (CBT) into health sector and availability of large healthcare data set to the practitioner and researchers (Uddin et al., 2019). It contains a set of methods that allows machines to learn meaningful patterns from data directly with minimal human interactions. The scope of this research work is primarily on the performance analysis of disease prediction using different ML classifiers (Culler et al., 1998).

The structure of the paper is organised as: Section 2 covers the related work, Section 3 focuses on the proposed methodology, Section 4 discusses the experimental work and the analysis of the results and Section 5 covers the conclusion.

1.1 Research contribution

1) Graph cut based segmentation along with fusion of GLCM, LBP and intensity-based feature extraction techniques is used for the detection of thyroid nodules. Further, various ML classifiers like SVM, ANN, DT, RF, KNN, XGBoost, Naïve Bayes and LR is explored for the classification.

2) The performance of all eight models is evaluated on 5-fold and 10-cross-validation on public and collected data sets. An improvement of 4 to 5% is seen in comparison with the reported literature.

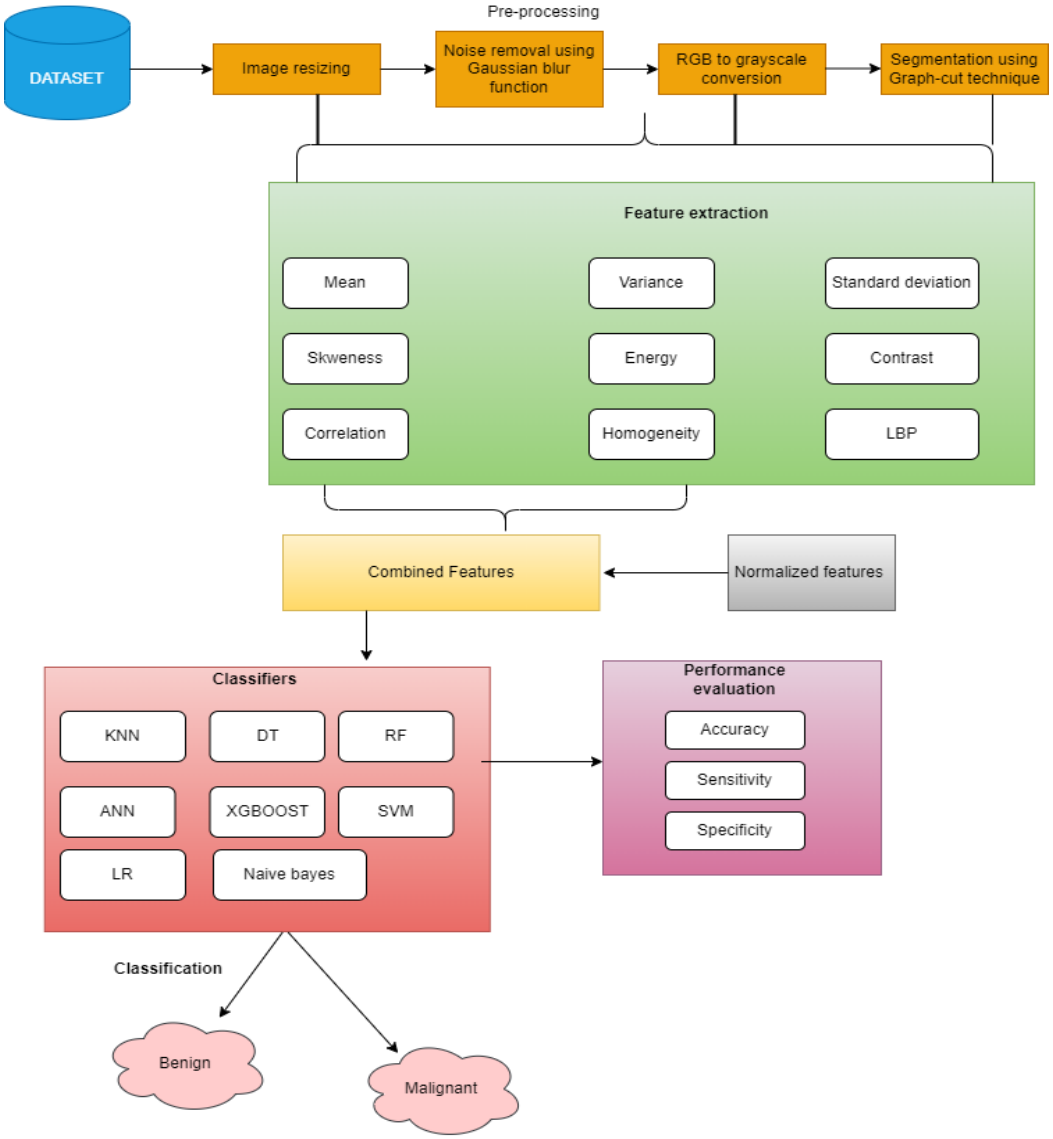
2 Related work

In last few decades, many ML algorithms have been used to identify and classify thyroid nodules. (Song et. al., 2015) used GLCM for feature extraction and various ML techniques like logistic regression, DT, RF, XGBoost, SVM and ANN for classification. Among all these classifiers logistic regression achieved the highest result with 78.5% accuracy. Nugroho et al. (2016) proposed a CAD system that used a fusion (GLCM, histogram + Grey Level Run Length Matrix (GLRLM)) methods. The model has achieved an accuracy of 89.74%, sensitivity of 88.89% and specificity of 91.67% using MLP as the classifier. The model was evaluated on less sample size, i.e., 39 images. Jiang et al. (2017) proposed an intelligent-based model using various ML techniques. The model has achieved an accuracy of 84%. Dandan et al. (2018) proposed a model using wavelet multi-sub-bands Co-occurrence matrix (WMCM) for feature extraction and SVM for the classification. The model has achieved 87% accuracy using 180 USG images. Colakoglu et al. (2019) proposed a CAD system. It was found that RF achieves highest result with 86.8% accuracy, 85.2% sensitivity and 87.95% specificity. The model was evaluated on 235 USG images. (Tüzüner and Ataç, 2020) addressed the classification of thyroid tumor images to TIRADS categories via texture analysis methods. The model used fusion (GLCM, GLRM and Law's Texture Energy Measures) method for feature extraction and SVM and RF for the classification. The model has achieved 85% accuracy. Xie et al. (2020) proposed a novel hybrid model using DL and handcrafted features LBP for feature extraction and achieved an accuracy of 85%. The model was evaluated on 623 USG images collected from Shanghai Tenth People's hospital. Sun et al. (2020) proposed hybrid method with 86.5% accuracy using fusion of DL-based technique and statistical features.

3 Proposed methodology

For the performance comparison of various ML classifiers, there are four steps involved in this research work. First, data collection, second pre-processing steps, third segmentation and feature extraction classifiers and fourth classification using various machine learning classifiers. Figure 2 shows the proposed model.

Figure 2 Proposed model (see online version for colours)



3.1 Phase I: data collection

In this work, two data sets are considered: 1. Public thyroid Ultrasound Image Database open data set for thyroid USG images (Pedraza et al., 2015). 2. Collected data set form Kriti Scanning Diagnostic Cancer duly approved by NABH (https://www.nabh.co/frmViewCGHSRecommend.aspx?Type=Diagnostic%20Centre&cityID=94). The total number of public thyroid USG images available is 295, out of which 107 and 188 were benign and malignant images. In case of collected data set, total number of images was 654 out of which 428 and 256 were benign and 256 malignant images (Srivastava and Kumar, 2021b).

3.2 Phase II: pre-processing phase

Some of the pre-processing steps involved are image resizing, noise removal and RGB to greyscale conversion. Initially sample size of images was 560 × 360 pixels, it's been resized to 256 × 256 pixels. The major difference between the greyscale and coloured images is the bit size. Greyscale image uses 8-bit for each pixel that is the combination of eight binary numbers. Thus, the range for pixel is from 0 to 255. While in the case of RGB scale image, it uses 24-bit that is each pixel is represented by three bytes (RGB) and supports 256 × 256 × 256 possible combined colours. Hence, RGB increases the complexity of the problem. That's why greyscale image is used in this work. Conversion of RGB to grey scale is performed using equation (1):

$$I = rgb2grey(RGB) \tag{1}$$

Noise is an inherent property of medical USG imaging. It generally tends to reduce the image resolution and contrast, reducing the diagnostic value of the imaging modality. Thus, it is necessary to remove noise present in the USG images. Noise removal from ultrasound images is performed using Guassian blur function. Guassian blur function is computed using equation (2):

$$h = fspecial(type) \tag{2}$$

where fspecial: returns h as a correlation kernel, h: creates a 2D filter h of the specified type.

3.3 Phase III: segmentation and feature extraction techniques

3.3.1 Image segmentation

It is defined as an analysis to partition an image into various segments. It involves converting an image into a collection of regions of pixels. Segmentation helps us to focus on the important segments rather than the entire input image (https://www.nabh.co/frmViewCGHSRecommend.aspx?Type=Diagnostic%20Centre&cityID=94). In this work, Graph Cut (GC) method-based segmentation technique is used to segment the thyroid nodules effectively in MATLAB 2019B. Let a graph denoted by G and computed using equation (3):

$$G = (V, E, W) \tag{3}$$

where V: vertex, E: edges, W: weights assigned to edges.

In the above equation, V is been divided into two parts V_o and V_a. The V_o corresponds to pixels of an image and V_a covers the terminal nodes. Similarly, E is also divided into two parts:

1) E_n: connecting the neighbouring pixels;

2) E_t: connecting the terminal nodes with neighbourhood nodes

The cut C is subset of the E and V having two disjoints sets if C is removed from G. The cost of the C (minimum cut with smallest cost) is total of the W and E of C (https://in.mathworks.com/discovery/image-segmentation.html). The optimal cost cut is computed using equation (4):

$$|C| = \sum_{s \in C} \omega_E \tag{4}$$

The energy cost $E(f)$ function is minimised using equation (5):

$$E(f) = \lambda * E_R(f) + E_B(f) \tag{5}$$

where E_R: measures how well the vertices V_o fits the labels f, E_B: boundary constraint for segmentation.

The E_R is defined using equation (6):

$$E_R(f) = \sum_{p \in V_0} D_p f_{(p)} \tag{6}$$

where $D_p f_{(p)}$: penalty for label f_p and is computed using equations (7) and (8):

$$D_P(1) = -\ln \Pr_p(1) = -\ln\left(\frac{n_{p,1}}{n_1}\right) \tag{7}$$

$$D_P(0) = -\ln \Pr_p(1) = -\ln\left(\frac{n_{p,0}}{n_0}\right) \tag{8}$$

where $\Pr_p(1), \Pr_p(0)$: possibility of node p being labelled as object (obj.) and background (B),

The boundary term E_b measures the extent to which f is not piecewise smooth. E_B plays a significant role as if E_B makes F smooth everywhere the result of segmentation will be poor (Price et al., 2010). E_B is computed using equation (9):

$$E_B(f) = \sum_{\{p,q\} \varepsilon N} V_{p,q}(f_p, f_q) * \delta(f_p, f_q) \tag{9}$$

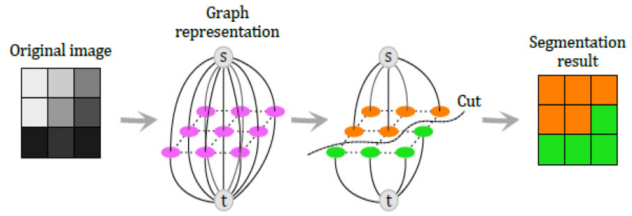
$$\delta(f_p, f_q) = \begin{cases} 1, & f_p = f_q \\ 0, & f_p \neq f_q \end{cases} \tag{10}$$

where N: neighbouring element, $V_{p,q}$: the internal relationship of nodes p and q and is computed using equation (11):

$$V_{p,q} = \exp - \left(-\frac{I_p = I_q}{2\sigma^2}\right) * \left(-\frac{1}{d_{p,q}}\right) \tag{11}$$

Hence, when the intensities of two neighbouring nodes, i.e., p and q are different, the minimum energy is said to be obtained which indicates that the segmentation is successful. Figure 3 shows the graph cut segmentation technique.

Figure 3 Graph cut segmentation technique (see online version for colours)



Source: Gauriau et al. (2015).

Algorithm 1: Graph cut image segmentation

Step 1:	Define graph
Step 2:	Set weights to foreground and background
Step 3:	Set weights for edges between pixels.
Step 4:	Apply mask
Step 5:	Apply min-cut algorithm
Step 6:	Go to step 2 using current labels to compute foreground and background.

3.3.2 Feature extraction techniques

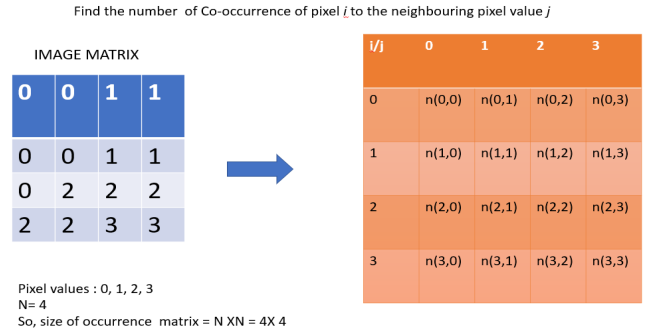
Feature extraction can be described as the processes of making a source/raw data into some numerical value features. This helps to identify two or more images (Peng et al., 2013). In this work, the handcrafted features are extracted from the images using GLCM, LBP and intensity-based features extraction techniques. Normalisation avoids raw data and various problems of data sets by creating new values and maintaining general distribution as well as a ratio in data. Further, it also improves the performance and accuracy of machine learning models using various techniques and algorithms. Min-max normalisation is used in this research work to normalise the data sets. It works by assigning 0 for every minimum value and the maximum value gets transformed into 1. The other values are transformed into a decimal between 0 and 1. All these extracted features are normalised using equation (12):

$$\text{normalised } z_i = \frac{\left(x_i - \min\left(x\right)\right)}{\max\left(x\right) - \min\left(x\right)} \tag{12}$$

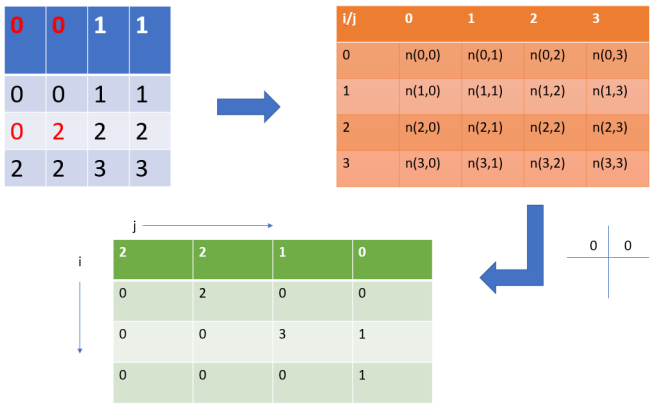
where $\min(x)$: minimum value in the sample; x_i: i-th normalised value in data set; $\max(x)$: maximum value in the sample

GLCM and intensity-based feature extraction techniques: It reflects the grey level of the image in the adjacent direction, etc., and also reflect the position distribution characteristics between the pixel with the same grey level. The distribution of GLCM matrix elements with respect to diagonal reflect the thickness of the image texture (Sekhar et al., 2021). Regions having rich coarse textures have a greater number of non-zero elements in GLCM and densely distributed near the main diagonal. For regions having fine textures, the non-zero elements in GLCM are distributed (Xia et al., 2020). The working of GLCM is explained in Figures 4(a) and 4(b).

Figure 4 Working of GLCM (see online version for colours)



(a)



(b)

Intensity is defined as the average pixel values, variance, asymmetry or standard deviation of the whole input image. The features extracted are mean, variance, standard deviation, skewness, contrast, correlation, energy and homogeneity. The extracted features description are as follows:

Mean:

$$\text{Mean} = \sum_{i,j=1}^{n} P\left(i,j\right) \tag{13}$$

where: P: co-occurrence matrix

Variance:

$$\text{Variance} = \sum_{i,j=1}^{n} \left(i,j\right)^2 P\left(i,j\right) \tag{14}$$

Standard Deviation:

$$\text{Standard Deviation} = \sqrt{\sum_{i,j=1}^{n} P\left(i,j\right)\left(i-\mu\right)^2} \tag{15}$$

where μ : mean

Skewness:

$$\text{Skewness} = \sum_{i,j=1}^{n} \left(i-j\right)^3 P\left(i,j\right) \tag{16}$$

Energy:

$$\textit{Energy} = \sum_{i,j=1}^{n} P\left(i,j\right)^2 \tag{17}$$

Contrast:

$$\text{Contrast} = \sum_{i,j=1}^{n} P\left(i,j\right)\left(i-j\right)^2 \tag{18}$$

Correlation:

$$\text{Correlation} = \sum_{i,j=1}^{n} P\left(i,j\right)\frac{\left(i-\mu\right)\left(j-\varpi\right)^2}{\sigma} \tag{19}$$

where σ : standard deviation

Homogeneity:

$$\text{Homogeneity} = \sum_{i,j=1}^{n} \frac{P\left(i,j\right)}{1+\left(i-j\right)^2} \tag{20}$$

Algorithm 2: Feature extraction using GLCM

Step 1: Define matrix (MN), quantisation level (QL) and displacement (D).

Step 2: glcm= zeros ([NumQuantlevels NumQuantLevels]);

Step 3: **for** i=1: size (MN, 1)

  for j=1: size (MN, 2)–1

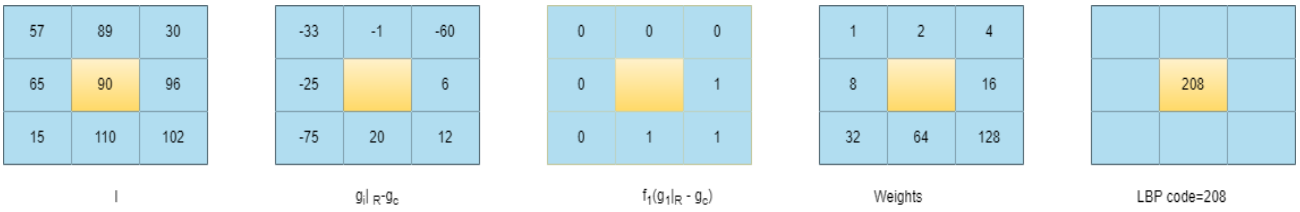
    glcm (MN (I, j), MN (I, j+1) =glcm (MN (I, j), MN (I, j+1)) +1,

  end

end

LBP: It was introduced to classify the textures from greyscale image and is widely used in image processing (Garg and Dhiman, 2021). It is one of the effective texture pattern descriptors. It works in block size of 3 × 3 where the centre pixel is used as a threshold (T) for the neighbouring pixel. The LBP code of a centre pixel is generated by encoding the computed threshold value (T_v) into a decimal value (Ojala et al., 2002). Let the centre of the pixel image be I, its value is calculated by comparing its grey value with its neighbours as given in Figure 5.

Figure 5 Working of LBP (see online version for colours)



Source: Murala et al. (2012).

It is computed using equations (21) and (22):

$$LBP_{P,R} = \sum_{i=1}^{P} 2^{(i-1)} xf_1\left(gi\middle|R - gc\right) \tag{21}$$

$$f1(x)\begin{cases}1 & x > 0\\ 0 & \text{else}\end{cases} \tag{22}$$

where gc: grey value of the pixel, P: number of neighbours at a distance R from gc, $gi|R$: grey value of neighbour at a radius R from gc.

Algorithm 3: Feature extraction using LBP

Step 1: Load data set (ImgDThy) having size of width (Wd) x height (Hg)

Step 2: **for** i=1 to Wd-2 **do**

Step 3: **for** j=1 to Hg-2 **do**

Step 4: block=ImgDThy(i: i+2, j: j+2); /*cal. Values

Step 5: **end**

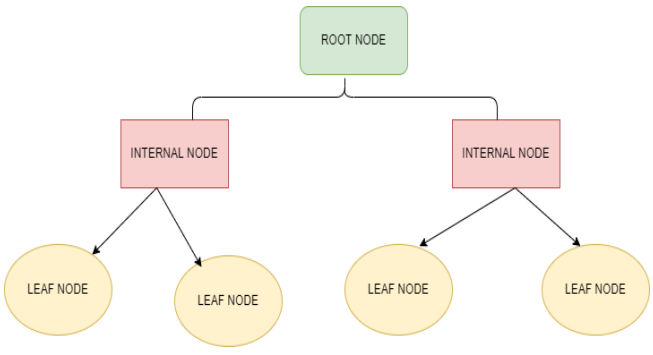
Step 6: **end**

3.4 Phase IV: classification

3.4.1 DT

It is used for both the classification and regression tasks (Gautam and Raman, 2020). It comprises several Leaf Nodes (LN), Internal Nodes (IN) and a single root node. Every LN possesses a class label and is connected to the root node via internal nodes (Quinlan, 1993). Figure 6 shows an architecture of DT.

Figure 6 Architecture of DT (see online version for colours)



Source: Azad et al. (2021).

Algorithm 4: Classification using DT

S: Sample, Feat: features, *R*: root, *L*: leaf, *C*: child

Step 1: ***If*** stopping_criteria(S, Feat)= Yes **then**

L=createNode()
L Label=classify(*y*)
return *L*

Step 2: *R*= createNode()

Step 3: R.test_condition= findBestSplit(S,Feat)

Step 4: Vertex= {v\v possible outcomecfroot.test_condition}

Step 5: **for** each value *v* ϵ Vertex:

Sv={s|R.test_condition(s)= *v* and *s* ϵ *S*};
C= TreeGrowth (sv, Feat);
Add *R* →*c* as *v*

Step 6: **return** prediction

3.4.2 RF

It is a parallel ensemble technique which fits several DT in parallel on different data-set sub samples (Zaki et al., 2014). One of the advantages of RF is that it solves over-fitting problem and increases the accuracy prediction of the model (Breiman, 2001). It is useful in classification and regression problem and can handle categorical and continuous data. Figure 7 shows the architecture of RF.

Figure 7 Architecture of RF (see online version for colours)

Source: https://corporatefinanceinstitute.com/resources/knowledge/other/random-forest/

Algorithm 5: Classification using RF

Input: DTI: decision tree inducer, *N*: no. of iterations, Tset: training set, μ: sample size, NA: no. of attribute

Output: M_i: t= 1……...*N*

Step 1: **for** each *N* **do**

Step 2: $Tset_i$← Sample μ instances from Tset with replacement

Step 3: Build classifier R_n using DTI (NA) on $Tset_i$

Step 4: *i*++

Step 5: **end**

3.4.3 Naïve Bayes

It is based on Bayes' theorem with the assumption of independence between each pair of features. It works well on the small data set sample size and fits fell on binary and multi-class feature data set (Le Cessie and Van Houwelingen, 1992).

Algorithm 6: Classification using Naïve Bayes

Feat: features, *P*: prediction, μ: mean, σ: standard deviation, *p*: probability, PA: predictor attribute, TA: target attribute

Step 1: Load extracted feat

Step 2: Select TA and PA

Step 3: **for** each PA

Calculate μ, σ

Step 4: **for** each class in TA

for each PA

Calculate P

Step 5: **for** each class in TA

Step 6: Calculate likelihood

Step 7: **return** *P*

3.4.4 SVM

It is one of the widely used algorithm in ML for classification and regression task. There are different types of kernels of SVM like Radial Basis Function (RBF), linear, etc which works on different mathematical functions (Han et al., 2011). However, in this work RBF performs best among the rest of the kernels.

Algorithm 7: Classification using SVM

Input: feat: features

Output: *P*: prediction

Step 1: Load feat

Step 2: Select the optimal value of cost and kernel for SVM

Step 3: **while** (stopping criteria is not satisfied) **do**

Step 4: Train SVM on test and training data point

Step 5: **end while**

Step 6: **return** P

3.4.5 K-nearest neighbour

KNN is widely known as lazy learning algorithm as it does not focus on constructing a general internal model (Keerthi et al., 2001). It uses Euclidean distance to classify new data points (Aha et al., 1991).

Algorithm 8: Classification using KNN

Input: feat: features, T_{train}: training data, l: label, T_{test}: test data

Output: *P*: prediction

Step 1: Load feat

Step 2: Classify (T_{train}, l, T_{test})

Step 3: **for** *i*=1 to *m* **do**

Compute distance *d* (T_{traini}, T_{train})

Step 4: **end**

Step 5: Compute set *I* having min set of *k* distance *d* (T_{traini}, T_{train})

Step 6: **return** majority l for $\{L_i; i \in I\}$

Step 7: **return** *P*

3.4.6 Linear regression

For the classification and regression purposes LR is widely recommended, but it has shown a good result in the field of classification problem (Pedregosa et al., 2011). It uses logistic function to estimate the probabilities (P) defined by sigmoid function:

$$g(z) = \frac{1}{1+\exp(-z)} \tag{23}$$

Algorithm 9: Classification using LR

Input: leR: learning_rate, gd: gradient_descent. Wt: weights

Output: prediction: *P*

i. Calculate gd;
return Wt+ leR*gd

ii. repeat step 1:
Wt= gd;
Until convergence

iii. *g*(*Z*)= dot product of predictor var and new Wt

iv. *P*= sigmoid function *g*(*z*)

v. **return** *P*

3.4.7 Extreme gradient boosting

It is an ensemble learning algorithm that generates model typically based on DT (Le Cessie and Van Houwelingen, 1992). The term gradient minimises the loss function similar to Neural Networks (NN) that uses gradient descent to optimise the weights (Han et al., 2011). It can handle large sample size of the data set (Harmon et al., 2020).

Algorithm 10: Classification using XGBoost

Input: feat: features, *p*: parameters

Output: *P*: prediction

i. Load feat

ii. Define *p* []

iii. Fit model

iv. Get predicted class *c*' []

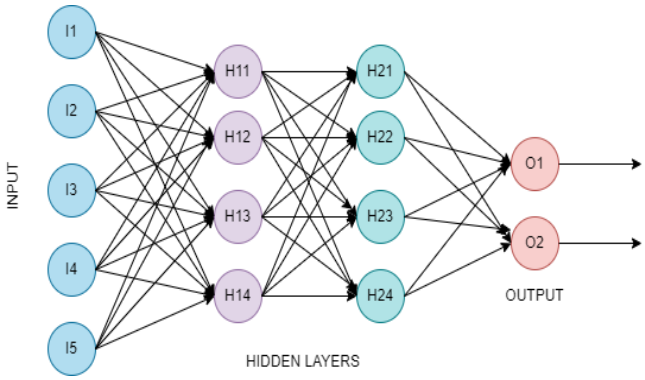
v. Evaluate (*c* [] =? *c*' [])

vi. **return** *P*

3.4.8 Artificial neural network

It consists of 3 layers (i) input, (ii) hidden and (iii) output layer (Demuth et al., 2014). Features extracted from the images are feed, as an input to the model, while hidden layers have 2 or more hidden layers. The result is obtained in the output layer. The basic architecture of ANN is shown in Figure 8.

Figure 8 Architecture of ANN (see online version for colours)



Source: Srivastava and Kumar (2022).

Algorithm 11: Classification using ANN

Input: feat: features

Output: *P*: prediction

Step 1: Load feat

Step 2: **for** i_n layer=1……..*n* layer

for i_n=1 to neuron(layer$_i$)

Step 3: **Calculate activation wt**

Sum[i_n]=bias [layer$_i$ i_n]

Step 4: **for** j_n=1……n neuron [layer$_i$ i_n – 1]

Sum[i_n]= wt[layer][i_n][j_n]* output[layer$_i$][i_n]

next j_n

Step 5: **Calculate output**

Output [layer][i_n]=activation fun (sum[i_n])

Step 6: **return** *P*

Algorithm 12: Detection and classification of thyroid nodules on medical ultrasound images

Input: ImgD: Image data set

Output: prediction

Classified image (Cimg): classified benign and malignant thyroid images

Step 1: Start

Step 2: Upload a data set: ImgD

Step 3: Resize the image.

Step 4: Convert the RGB to grey scale using equation (1).

Step 5: Remove noise using Guassian blur function using equation (2).

Step 6: Labelled the data set

Step 7: Segment the thyroid nodules using graph cut method using equation (3)

Step 8: Extract features using equations (13) to (21).

Step 9: Normalise the extracted features using equation (12).

Step 10: Set the training and testing ratio.

Step 11: Perform classification using various ML classifiers: Cimg

Step 12: Prediction

Step 13: Stop

4 Experimental and result analysis

This section is divided into two parts experiment-I and experiment-II. In both the experiments, steps are same only classifiers are changed and tested on 5-fold and 10-fold Cross-Validation (CV). So, for better representation, classification performed with KNN is named as model-1, DT as model-2, RF as model-3, ANN as model-4, XGBoost as model-5, Naïve Bayes as model-6, SVM as model-7 and LR as model-8. Both experiments are executed on MATLAB 2019B, processor i5 8th generation, 16 GB RAM, 256 SSD.

Experiment-I: In this experimental part, results obtained on public data set is discussed. For the better representation, public data set is renamed as data set-1. Eight different types of ML classifiers like KNN, DT, SVM, RF, XGBoost, Naïve Bayes, ANN, LR are explored. Once the pre-processing, segmentation is done, features are extracted, normalised and is feed as input to the model. 80% training and 20% testing is set for the data set. Table 1 shows the performance of various proposed (1-7) models on ML classifiers based on 5-fold and 10-CV on dataset-1. From the table, it is found that model performs better using 10-CV in comparison with 5-fold CV. Further, an improvement of 1 to 2% is seen in the performance of the classifiers. Table 2 shows the comparison of the proposed Model-4 with

reported literature on dataset-1. Figure 9 shows the performance comparison of the proposed (1-7) models on various ML classifiers based on accuracy, sensitivity and specificity on dataset-1. It can be observed that ANN classifier has achieved 92.39% accuracy, sensitivity of 93.54% and specificity of 91.30% which is highest among the rest of the classifiers. The rest of sequence of the classifiers are SVM with 91.32%, LR with 90.31%, RF with 89.89%, DT with 89.01%, XGBoost with 87.77%, Naïve Bayes with 86.15% and KNN with 83.58% accuracies. Figure 10 shows the comparison of the proposed Model-4 with the reported literature based on accuracy on dataset-1. An improvement of 4% is observed in comparison with reported literature, i.e., 92.59%. Figure 11 shows the comparison of the proposed Model-4 with reported literature-based sensitivity and specificity on dataset-1. Here also, proposed model-4 performs better in comparison with rest of the models with sensitivity of 93.54% and specificity of 91.30%.

Table 1 Performance comparison of ML classifiers based on 5-fold and 10 CV on dataset-1

Models	*Cross-validation*	*Accuracy*	*Sensitivity*	*Specificity*
Model-1	5-fold	82.12	83.17	81.81
	10-fold	83.58	84.76	82.22
Model-2	5-fold	88.35	89	87.64
	10-fold	89.01	90.72	88
Model-3	5-fold	88.54	89	88.04
	10-fold	89.89	90.81	89.88
Model-4	5-fold	91.90	92.70	90.90
	10-fold	92.59	93.54	91.30
Model-5	5-fold	86.15	87.25	85.86
	10-fold	87.77	89	86.25
Model-6	5-fold	85.27	86.40	84.94
	10-fold	86.15	87.25	85.86
Model-7	5-fold	90.75	91.75	89.47
	10-fold	91.32	92.70	90.78
Model-8	5-fold	89.01	92.70	91.25
	10-fold	90.39	91.75	89.87

Table 2 Comparison of the proposed model-4 with reported literature on dataset-1

Ref ids	*Accuracy (%)*	*Sensitivity (%)*	*Specificity (%)*
Song et al. (2015)	78.5		
Nugroho et al. (2016)	89.74	88.89	91.67
Jiang et al. (2017)	84		
Dandan et al. (2018)	87		
Colakoglu et al. (2019)	86.8	85.2	87.95
Tüzüner and Ataç (2020)	85		
Xie et al. (2020)	85		
Sun et al. (2020)	86.5		
Proposed Model-4 on dataset-1	92.59	93.54	91.30

Experiment-II: In this experimental part, results obtained on collected data set is discussed. For the better representation, collected data set is renamed as dataset-2. Eight different types of ML classifiers like KNN, DT, SVM, RF, XGBoost, Naïve Bayes, ANN, LR is explored. Once the pre-processing, segmentation is done, features are extracted, normalised and is feed as input to the model. 80% training and 20% testing is set for the data set. Table 3 shows the performance of various proposed (1-7) models on ML classifiers based on 5-fold and 10-CV on dataset-2. From the table, it is found that model (1-7) performs better using 10-CV in comparison with 5-fold CV. Further, an improvement of 1 to 2% is seen in the performance of the classifiers. Table 4 shows the comparison of the proposed model-4 with reported literature. Figure 12 shows the performance comparison of various proposed (1-7) models on ML classifiers based on accuracy, sensitivity and specificity on dataset-2. It can be observed that ANN classifier has achieved 93.42% accuracy, sensitivity of 94.62% and specificity of 91.52% which is highest among the rest of the classifiers. The rest of sequence of the classifiers are SVM with 92.40%, LR with 92%, RF with 91.52%, DT with 89.47%, XGBoost with 88%, Naïve Bayes with 87% and KNN with 85.79% accuracies. Figure 13 shows the comparison of the proposed model-4 with the reported literature based on accuracy. An improvement of 5% is observed in comparison with reported literature, i.e., 93.42%. Figure 14 shows the comparison of the proposed model-4 with reported literature-based sensitivity and specificity. Here also, proposed model-4 performs better in comparison with rest of the models with 94.62% sensitivity and 91.52% specificity.

Figure 9 Performance comparison of the proposed (1-7) models on various ML classifiers based on accuracy, sensitivity and specificity on dataset-1 (see online version for colours)

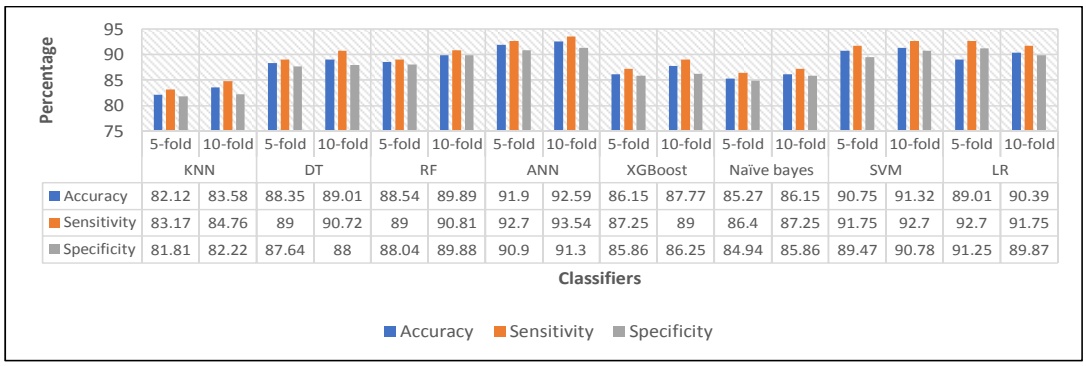


Figure 10 The comparison of the proposed model-4 with the reported literature based on accuracy on dataset-1 (see online version for colours)

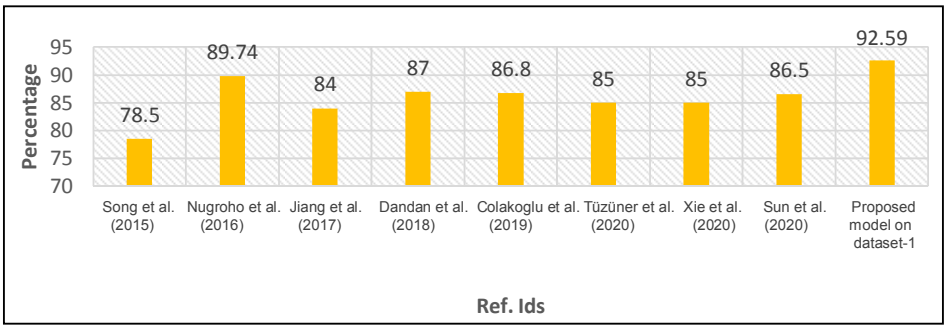


Figure 11 The comparison of the proposed model-4 with reported literature-based sensitivity and specificity dataset-1 (see online version for colours)

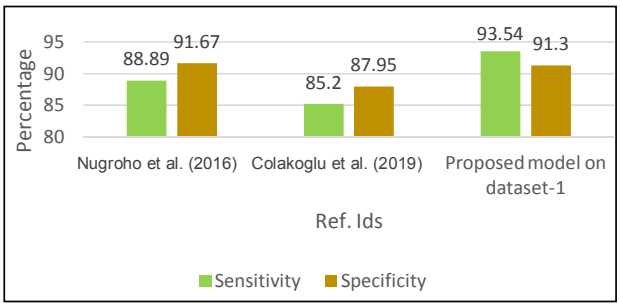


Table 3 Performance comparison of ML classifiers based on 5-fold and 10 CV on dataset-2

Models	*Cross-validation*	*Accuracy*	*Specificity*	*Sensitivity*
Model-1	5-fold	84.37	82.95	85.57
	10-fold	85.79	82.92	86.40
Model-2	5-fold	88.50	86.66	89.89
	10-fold	89.47	87.83	90.72
Model-3	5-fold	90.85	89.74	91.75
	10-fold	91.52	91.25	92.70
Model-4	5-fold	92.40	90.62	93.40
	10-fold	93.42	91.52	94.62
Model-5	5-fold	87.86	85.13	89
	10-fold	88	86.66	89.89
Model-6	5-fold	86.13	85.85	86.40
	10-fold	87	86.73	87.25
Model-7	5-fold	91.27	90.66	92.70
	10-fold	92.40	90.90	93.47
Model-8	5-fold	91.32	90.78	92.70
	10-fold	92	89.65	93.47

Table 4 Comparison of the proposed model-4 with reported literature on dataset-2

Ref ids	*Accuracy (%)*	*Sensitivity (%)*	*Specificity (%)*
Song et al. (2015)	78.5		
Nugroho et al. (2016)	89.74	88.89	91.67
Jiang et al. (2017)	84		
Dandan et al. (2018)	87		
Colakoglu et al. (2019)	86.8	85.2	87.95

Table 4 Comparison of the proposed model-4 with reported literature on dataset-2 (continued)

Ref ids	*Accuracy (%)*	*Sensitivity (%)*	*Specificity (%)*
Tüzüner and Ataç (2020)	85		
Xie et al. (2020)	85		
Sun et al. (2020)	86.5		
Proposed model-4 on dataset-2	93.42	94.62	91.52

Figure 12 Performance comparison of various proposed (1-7) models on ML classifiers based on accuracy, sensitivity and specificity on dataset-2 (see online version for colours)

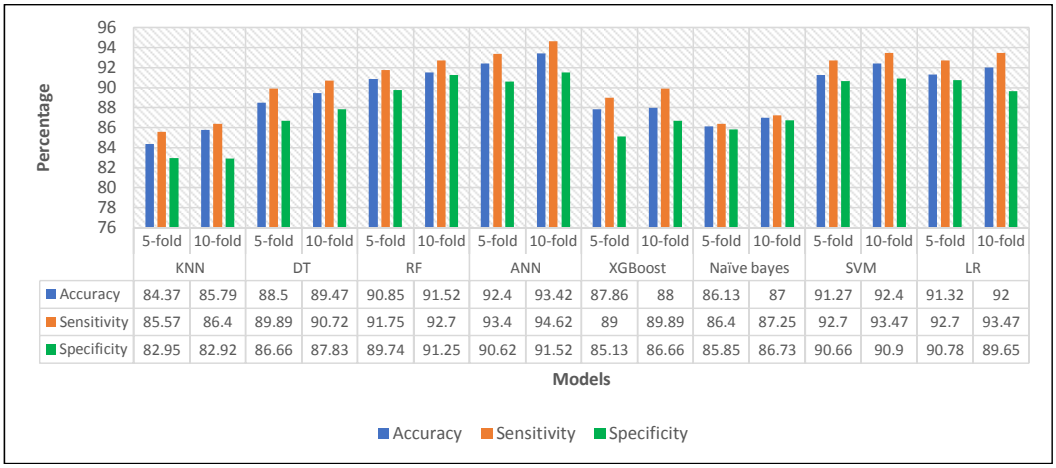


Figure 13 The comparison of the proposed model-4 with the reported literature based on accuracy on dataset-2 (see online version for colours)

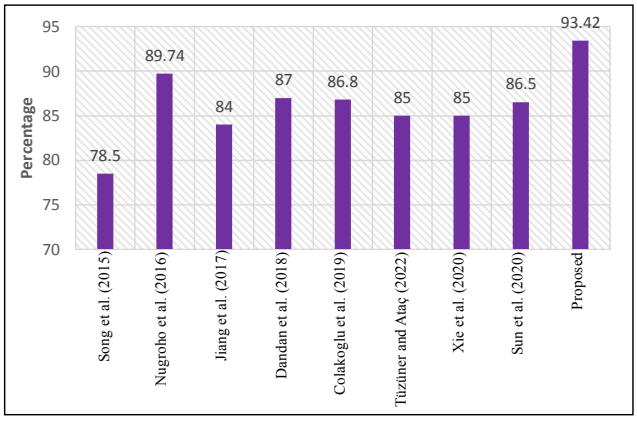
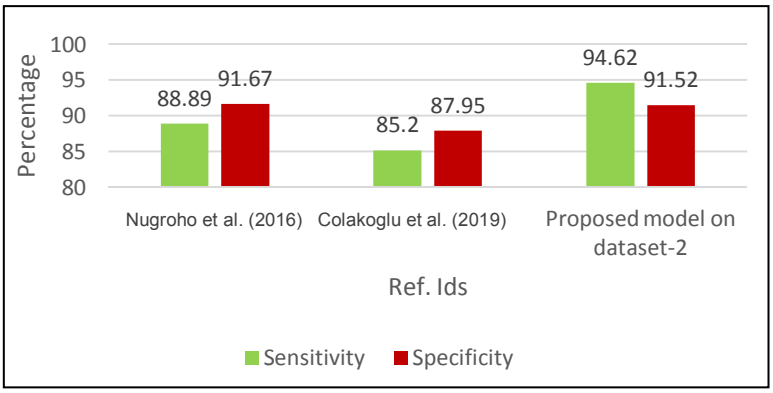


Figure 14 The comparison of the proposed model-4 with reported literature-based sensitivity and specificity on dataset-2 (see online version for colours)



5 Conclusion

In this research paper, fusion of LBP, Intensity and LBP based feature extraction techniques is explored with various ML classifiers. For better detection of thyroid nodules, graph-cut technique is applied on medical USG images. The experiment result shows that using 10-fold CV models yields an improvement of 1% to 2% in comparison with 5-fold CV. The proposed model, i.e., classification with ANN, has performed better than the other seven models (i.e., classification with SVM, LR, RF, DT, KNN, XGBoost, Naive Bayes) on dataset-1 and 2. An improvement of 4 to 5% is seen in performance evaluation in comparison with reported literature. It can be concluded that our proposed model could give a more accurate prediction to the clinicians for detection and classification of thyroid nodules. It can also be used for the study purpose of thyroid nodule identification by the researchers, practitioners, healthcare professionals. Future research work can be conducted on the large public data sets such as the lung or brain diseases. This will help to properly assess the real advantage of the proposed methods in the clinical realm, and the real improvement in performance over different deep learning architectures.

References

Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Mohamed, N.A. and Arshad, H. (2018) 'State-of-the-art in artificial neural network applications: a survey', *Heliyon*, Vol. 4, No. 11. Doi: 10.1016/j.heliyon.2018.e00938.

Aha, D.W., Kibler, D. and Albert, M.K. (1991) 'Instance-based learning algorithms', *Machine Learning*, Vol. 6, No. 1, pp.37–66.

Azad, C., Bhushan, B., Sharma, R., Shankar, A., Singh, K.K. and Khamparia, A. (2021) 'Prediction model using SMOTE, genetic algorithm and decision tree (PMSGD) for classification of diabetes mellitus', *Multimedia Systems*, pp.1–19.

Breiman, L. (2001) 'Random forests', *Machine Learning*, Vol. 45, No. 1, pp.5–32.

Cheng, C.H. and Liu, W.X. (2018) 'Identifying degenerative brain disease using rough set classifier based on wavelet packet method', *Journal of Clinical Medicine*, Vol. 7, No. 6. Doi: 10.3390/jcm7060124.

Colakoglu, B., Alis, D. and Yergin, M. (2019) 'Diagnostic value of machine learning-based quantitative texture analysis in differentiating benign and malignant thyroid nodules', *Journal of Oncology*. Doi: 10.1155/2019/6328329.

Culler, S.D., Parchman, M.L. and Przybylski, M. (1998) 'Factors related to potentially preventable hospitalizations among the elderly', *Medical Care*, pp.804–817.

Dandan, L., Yakui, Z., Linyao, D., Xianli, Z. and Yi, S. (2018) 'Texture analysis and classification of diffuse thyroid diseases based on ultrasound images', *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, IEEE, pp.1–6.

Demuth, H.B., Beale, M.H., De Jess, O. and Hagan, M.T. (2014) *Neural Network Design*, Martin Hagan, Cambridge, MA, USA.

Garg, M. and Dhiman, G. (2021) 'A novel content-based image retrieval approach for classification using GLCM features and texture fused LBP variants', *Neural Computing and Applications*, Vol. 33, pp.1311–1328.

Gauriau, R. (2015) *Shape-based Approaches for Fast Multi-Organ Localization and Segmentation in 3D Medical Images*, Doctoral Dissertation, Telecom ParisTech.

Gautam, A. and Raman, B. (2020) 'Local gradient of gradient pattern: a robust image descriptor for the classification of brain strokes from computed tomography images', *Pattern Analysis and Applications*, Vol. 23, No. 2, pp.797–817.

Han, J., Pei, J. and Kamber, M. (2011) *Data Mining: Concepts and Techniques*, Elsevier.

Harmon, S.A., Sanford, T.H., Xu, S., Turkbey, E.B., Roth, H., Xu, Z. and Turkbey, B. (2020) 'Artificial intelligence for the detection of COVID-19 pneumonia on chest CT using multinational datasets', *Nature Communications*, Vol. 11, No. 1, pp.1–7.

Jiang, Y., Deng, Z., Chen, J., Wu, H., Choi, K.S. and Wang, S. (2017) 'Intelligent diagnostic methods for thyroid nodules', *Journal of Medical Imaging and Health Informatics*, Vol. 7, No. 8, pp.1772–1779.

Jung, N.Y., Kang, B.J., Kim, H.S., Cha, E.S., Lee, J.H., Park, C.S. and Choi, J.J. (2014) 'Who could benefit the most from using a computer-aided detection system in full-field digital mammography?', *World Journal of Surgical Oncology*, Vol. 12, No. 1, pp.1–9.

Keerthi, S.S., Shevade, S.K., Bhattacharyya, C. and Murthy, K.R.K. (2001) 'Improvements to Platt's SMO algorithm for SVM classifier design', *Neural Computation*, Vol. 13, No. 3, pp.637–649.

Le Cessie, S. and Van Houwelingen, J.C. (1992) 'Ridge estimators in logistic regression', *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, Vol. 41, No. 1, pp.191–201.

Meiburger, K.M., Acharya, U.R. and Molinari, F. (2018) 'Automated localization and segmentation techniques for B-mode ultrasound images: a review', *Computers in Biology and Medicine*, Vol. 92, pp.210–235.

Murala, S., Maheshwari, R.P. and Balasubramanian, R. (2012) 'Local tetra patterns: a new feature descriptor for content-based image retrieval', *IEEE Transactions on Image Processing*, Vol. 21, No. 5, pp.2874–2886.

Nguyen, D.T., Pham, T.D., Batchuluun, G., Yoon, H.S. and Park, K.R. (2019) 'Artificial intelligence-based thyroid nodule classification using information from spatial and frequency domains', *Journal of Clinical Medicine*, Vol. 8, No. 11. Doi: 10.3390/jcm8111976.

Nugroho, H.A., Rahmawaty, M., Triyani, Y. and Ardiyanto, I. (2016) 'Texture analysis for classification of thyroid ultrasound images', *International Electronics Symposium (IES)*, IEEE, pp.476–480.

Ojala, T., Pietikainen, M. and Maenpaa, T. (2002) 'Multiresolution gray-scale and rotation invariant texture classification with local binary patterns', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 7, pp.971–987.

Pedraza, L., Vargas, C., Narváez, F., Durán, O., Muñoz, E. and Romero, E. (2015) 'An open access thyroid ultrasound image database', *Proceedings of the 10th International Symposium on Medical Information Processing and Analysis*, International Society for Optics and Photonics. Doi: 10.1117/12.2073532.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O. and Duchesnay, E. (2011) 'Scikit-learn: machine learning in Python', *The Journal of Machine Learning Research*, Vol. 12, pp.2825–2830.

Peng, B., Zhang, L. and Zhang, D. (2013) 'A survey of graph theoretical approaches to image segmentation', *Pattern Recognition*, Vol. 46, No. 3, pp.1020–1038.

Price, B.L., Morse, B. and Cohen, S. (2010) 'Geodesic graph cut for interactive image segmentation', *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, pp.3161–3168.

Quinlan, J.R. (1993) 'C4.5: programs for machine learning', *Machine Learning*, Vol. 16, No. 3, pp.235–240.

Sekhar, K.S.R., Babu, T.R., Prathibha, G., Vijay, K. and Ming, L.C. (2021) 'Dermoscopic image classification using CNN with handcrafted features', *Journal of King Saud University-Science*, Vol. 33, No. 6. Doi: 10.1016/J.JKSUS.2021.101550.

Song, G., Xue, F. and Zhang, C. (2015) 'A model using texture features to differentiate the nature of thyroid nodules on sonography', *Journal of Ultrasound in Medicine*, Vol. 34, pp.1753–1760.

Srivastava, R. and Kumar, P. (2021a) 'BL_SMOTE ensemble method for prediction of thyroid disease on imbalanced classification problem', *Proceedings of 2nd International Conference on Computing, Communications, and Cyber-Security*, Springer, Singapore, pp.731–741.

Srivastava, R. and Kumar, P. (2021b) 'A hybrid model for the identification and classification of thyroid nodules in medical ultrasound images', *International Journal of Modelling, Identification and Control*, Vol. 41, Nos. 1/2, pp.32–42.

Srivastava, R. and Kumar, P. (2022) 'A CNN-SVM hybrid model for the classification of thyroid nodules in medical ultrasound images', *International Journal of Grid and Utility Computing*, Vol. 13, No. 6, pp.624–639.

Sun, H., Yu, F. and Xu, H. (2020) 'Discriminating the nature of thyroid nodules using the hybrid method', *Mathematical Problems in Engineering*. Doi: 10.1155/2020/6147037.

Tüzüner, A.B. and Ataç, G.K. (2020) 'Classification of ultrasonographic thyroid tumor images to TIRADS categories via texture analysis methods', *Medical Technologies Congress (TIPTEKNO)*, IEEE, pp.1–4.

Uddin, S., Khan, A., Hossain, M.E. and Moni, M.A. (2019) 'Comparing different supervised machine learning algorithms for disease prediction', *BMC Medical Informatics and Decision Making*, Vol. 19, No. 1, pp.1–16.

Xia, K., Gu, X. and Zhang, Y. (2020) 'Oriented grouping-constrained spectral clustering for medical imaging segmentation', *Multimedia Systems*, Vol. 26, No. 1, pp.27–36.

Xie, J., Guo, L., Zhao, C., Li, X., Luo, Y. and Jianwei, L. (2020) 'A hybrid deep learning and handcrafted features based approach for thyroid nodule classification in ultrasound images', *Journal of Physics: Conference Series*, IOP Publishing, China. Doi: 10.1088/1742-6596/1693/1/012160.

Zaki, M.J., Meira, W. and Meira, W. (2014) *Data Mining and Analysis: Fundamental Concepts and Algorithms*, Cambridge University Press.