# Computer Aided Tongue Diagnosis System using Color and Texture Feature Extraction-based Deep Learning CNN

**Sreerama Prasad Chelluboina[1*], Kunjum Nageswara Rao[2]**

[1]Research Scholar, Computer Science and Systems Engineering, AU College of Engineering (A), Andhra University, Visakhapatnam, Andhra Pradesh, India

[2]Professor, Computer Science and Systems Engineering, AU College of Engineering (A), Andhra University, Visakhapatnam, Andhra Pradesh, India

**Abstract**

Tongue diagnosis is an important way of monitoring human health status in Indian ayurvedic medicine (IAM), which helps to identify the different diseases of human through tongue image analysis. Several machine learning models are presented to classify the diseases through tongue image analysis. However, they are suffering with the low classification performance due to variations in tongue appearance such as color, shape, coating, and texture properties. Therefore, this article focuses on deep learning convolutional neural network (DLCNN) for disease predication through tongue image analysis, which is hereafter named as Tongue-Net. Initially, fast nonlocal mean (FNLM) filtering is applied on given tongue image for preprocessing operations such as noise removal, and quality enhancement. Next, color features from preprocessed tongue image are extracted using color statistics such as mean, skewness, and standard deviation. In addition, grey level cooccurrence matrix (GLCM) and local binary pattern (LBP) approaches are used extract the texture and shape features. Finally, DLCNN classifier is used to classify the different diseases from extracted features. The proposed Tongue-Net model is capable of predicting six distinct diseases including the healthy, appendicitis, bronchitis, gastritis, heart disease, and pancreatitis disease. The simulation results shows that proposed Tongue-Net classification model obtained 97.90% of accuracy, and 98.01% of F1-score.

**Keywords:** Indian ayurvedic medicine, tongue image analysis, disease prediction, color and texture features, deep learning, convolutional neural networks.

## 1. Introduction

In oriental medicine, such as IAM, traditional Chinese medicine, Japanese traditional herbal medicine, and traditional Korean medicine, the condition of a patient's internal organ can be effectively evaluated using a method that does not involve any sort of invasive procedure through the use of tongue diagnosis [1]. The procedure of diagnosis is dependent on the professional's opinion, which is derived from a visual examination that includes the color, material, coating, shape, and motion of the tongue. Traditional tongue diagnosis is more likely to detect the condition than it is to notice abnormalities in the look of the tongue and diseases that affect the tongue. For instance, a white, greasy coating on the tongue might indicate cold syndrome, while a yellow, thick coating on the tongue can indicate hot syndrome [2]. Both of these syndromes are linked to health issues such as infection, inflammation, stress, immunological diseases, or endocrine abnormalities. According to IAM while, the distinct parts of the tongue are shown in Figure 1 along with the internal organs that correlate to those parts of the tongue. Pathological changes in the heart and lungs are reflected in the tip of the tongue, whereas abnormalities in the liver and gallbladders are reflected on the sides of the tongue on both sides of the tongue. The center of the tongue reflects the pathological changes that have occurred in the spleen and stomach [3], while the root of the tongue reflects the pathological abnormalities that have occurred in the kidneys, intestines, and bladder region.

Traditional tongue diagnosis [4], on the other hand, is heavily dependent on the amount of clinical expertise a clinician has; as a consequence, various doctors are likely to arrive at different diagnostic conclusions for the same patient. Thankfully, computer-aided methods for tongue diagnosis are able to improve upon these shortcomings because to the utilization of computers and other approaches that are relevant to the field [5]. The objective, quantitative, and automated auxiliary analyses that have been introduced to contemporary tongue diagnosis as a result of developments in computer science and technology have assisted medical professionals in making more accurate diseases diagnoses. The modernization of the IAM tongue diagnostic has mostly resulted in the technique being broken down into three parts: segmentation of the tongue, extraction of tongue features, and analysis of the condition. In order to get an accurate image of the tongue from the initial image of the tongue, segmentation of the tongue makes use of a segmentation algorithm [6]. Because digitally acquired tongue images include not only the tongue region but also parts of the none-tongue regions such as lips, teeth, and so on, tongue segmentation from the whole image is a prerequisite for tongue characterization. Digitally acquired tongue images include both the tongue region and parts of the none-tongue regions. Nevertheless, tongue segmentation is a challenging process because of interference caused by shifting light and crowded backgrounds. After that, the characteristics of the tongue that are necessary for an IAM diagnosis are extracted. After the traits have been retrieved, they are utilized to categories and diagnose diseases. In recent years, approaches for tongue image feature extraction [7] have been subjected to considerable research. According to the findings of these investigations, the approaches of computer-assisted tongue diagnosis may be broken down into two distinct categories: single feature and multifeatured. There have been many different proposals for and applications of single feature extraction approaches [8] for tongue image analysis. These kinds of algorithms may extract important information based on a very basic descriptor, such as the color, texture, shape, or orientation of an object. The identification of the sort of tongue image that may assist IAM clinicians in reaching a further diagnostic conclusion is the work that is referred to as the tongue image-based diseases categorization. The segmentation of the tongue [9] has been addressed from a variety of angles and approaches in the past. Some of these methods are sensitive to changes in illumination or backgrounds that are clustered, some of these methods confuse the lips with the body of the tongue, and some of these methods require additional preprocessing, which makes the entire segmentation process more complicated.

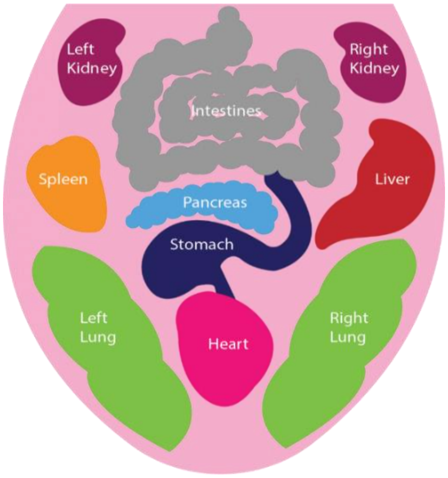


Figure 1: Reflex zones of tongue.

However, the majority of these methods are based on traditional image processing techniques [10]. In more recent times, automated tongue segmentation has been approached from the perspective of deep learning-based approaches. Although the performance of those approaches based on deep learning is superior to the performance of the most conventional methods for segmenting the tongue, those methods still have significant limitations. Image enhancement is an example of the additional preprocessing that is necessary, which contributes to the overall complexity of the segmentation process. In a similar way, brightness discrimination as a preprocessing step decreases a model's capacity for generalization based on deep learning [11]. Nevertheless, mask is able to locate the item; however, it is unable to differentiate between the different types of objects. As a consequence, the segmentation process is sluggish and less accurate than it might be since unrelated objects were processed unnecessarily. The following is a list of the primary goals of this work to achieve in order to address these issues:

- Implementation of Tongue-Net model to classify the healthy, appendicitis, bronchitis, gastritis, heart disease, and pancreatitis disease classes using DLCNN classifier.
- Initially, FNLM is applied to remove the noises from the tongue images, which also performs the color enhancements.
- Further, joint texture and color features are extracted from preprocessed image using color moment, LBP, and GLCM descriptors.
- Finally, DLCNN is used to classify the diseases using extracted features. The simulation results shows that the proposed Tongue-Net outperformed as compared to existing methods.

Rest of the paper is contributed as follows: Section 2 deals with the related work with their drawbacks. Section 3 deals with the detailed analysis of proposed Tongue-Net model. Section 4 deals with the results and discussions with comparative analysis. Section 5 deals with the conclusion and future scope.

## 2. Literature survey

This section is focused on survey of tongue disease detection and classification methods. Initially, machine learning models such as logical regression, random forest [12], support vector machine (SVM) and naive bayes are widely used to classify the early-stage oral tongue squamous cell cancer. Here, SVM [13] method resulted in better accuracy as compared to other machine learning models. Further, deep learning-based feedforward neural network (FFNN) [14] is also used to classify the OTSCC but resulted in the reduced accuracy such as 92.7%. In [15], authors developed the deep convolutional neural network (DCNN) model for classifying the benign and pre-cancerous diseases from tongue image. But this method suffering with the high computational complexity due to high training time. Further, DCNN also used for tongue health classification with ResNet34 [16] feature extraction. Anyhow, this method is capable of classifying healthy and unhealthy nature of tongue, but unable to classify the diseases from tongue images.

Recently, Tongue Diagnosis Analysis System (TDAS) [17] is proposed for detection of prediabetes and diabetes diseases from tongue images. The TDAS system contains ResNet-50 for extracting the color and textures features with SVM classification. However, this method is suffering with the high computational complexity. Further, deep learning based Radial Basis Function Neural Network (RBFNN) [18] based classifier is used for diabetes disease classification from tongue images. Here, ResNet50 us used to extract the color, shape, thickness, and tooth marked based features. Anyhow, this method suffers with the low classification accuracy with higher training time. In [19] authors proposed automated synergic deep learning-based tongue color image (ASDL-TCI) for tongue disease classification. Here, median filtering is used to perform the preprocessing operation with ASDL feature extraction. Finally, deep neural network (DNN) based classifier is used with enhanced black widow optimization (EBWO) optimization. But this method is unable to classify the multiple diseases from tongue images. In [20] authors used the machine learning models for classify the diabetes mellitus and symptoms of gastric disease from tongue images. Here, SVM recursive feature elimination (SVM-RFE) is used to extract the optimal color and texture features. Further, Fully-channel regional attention network [21] is developed for the disease identification and classification using multi-layer convolution networks. However, this method extracted the basic tongue features.

To overcome these problems, hybrid deep learning [22] models are developed by combining the individual CNN models with ensemble stacking. Anyhow, this method identifies the diseases using physique analysis of tongue, which resulted in poor performance. Further, Zero-Shot Learning for Constitution Recognition (ZSLCR) [23] is implemented with ResNet18 and ResNet34 models for tongue color analysis. This method utilizes the automatic weight updation properties, which resulted in the better performance but taking much time for training. In [24] authors implemented the diabetic risk prediction using hybrid machine learning classifier through tongue images. This work considered the ResNet50 for extracting the features from tongue images. However, there are huge loss of features during the training procedure, which resulted in reduced performance. In addition, the hybrid features [25] are extracted using transfer learning models, which analyzed the patterns of the tongue images and extracted hided features. Further, color-based disease analysis is performed using this method, which is not considered the feature selection operations. However, various works are carried out on single, two diseases diagnosis (classification) from tongue images, but multi-class disease classification from tongue images is still research problem. Further, there is no work has been published on multi class disease classification from tongue images.

## 3. Proposed methodology

Disease prediction from tongue images is difficult task as the tongue images contains different colors, structures. Therefore, this work is focused on implementation of hybrid machine learning models for disease prediction through tongue images. Figure 2 presents the proposed Tongue-Net model architecture. Initially, FNLM preprocessing method is performed for normalizing the sizes of the input image. The preprocessing operation also performs the noise removal operation and color illumination problems also overcome by using FNLM. Further, color features such as mean, skewness, standard deviation is extracted from preprocessed images using color moments. In addition, GLCM is also used to extract the texture features like energy, contrast, entropy, correlation, and homogeneity. Further, LBP is also used to extract the shape-based properties. Finally, DLCNN classifier is used to classify the different disease classes from extracted features. The proposed Tongue-Net model is capable of predicting the healthy, appendicitis, bronchitis, gastritis, heart disease, and pancreatitis disease classes.

### 3.1 Pre processing

The tongue images are capture in the different conditions, which contains the dissimilar properties like shape, size, rotation angle, resolution, number of pixels, foreground light, background intensities, color, contrast, brightens, hue-based color properties. The uneven nature of these properties makes the feature extraction more difficult, which causes reduction of performance. Further, the people may contain tongue infections, tongue cancers, and some other tongue diseases, which resulted in mismatch in disease prediction operation. Thus, image preprocessing operation is performed to overcome these problems, which normalizes the properties of all images equally. Further, image preprocessing also removes the basic background and foreground illumination problems of the tongue images. Usually, tongue images are damaged by several forms of artefacts, which lowers the overall performance. The FNLM successfully removes the numerous forms of sounds such as salt, pepper, gaussian, random, jitter, spackle, poison noises from tongue images. Further, FNLM further increases the contrast, brightness, contrast, color based statical qualities of tongue images.

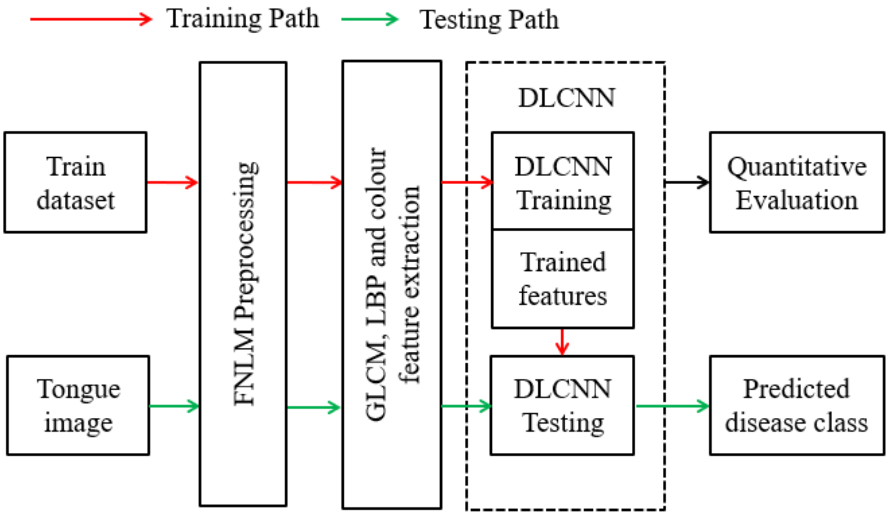


Figure 2. Proposed Tongue-Net architecture.

Consider the $f(n,m)$ is the noisy tongue image, which is created by adding the noise to original image. Equation (1) describes the noise model of tongue images.

$$f(n,m) = v(n,m) * \ u(n,m)\ +\ \gamma\ (n,m) \tag{1}$$

In this case, n and m stand for the row and column locations of the tongue image, which may also be thought of as spatial coordinates. In addition, $u(n,m)$ is the representation of the original image without any noise, $\gamma\ (n,m)$ is the representation of the artefacts that include a variety of noises, and $v(n,m)$ is the representation of the damaged image pixel characteristics. The FNLM operation is carried out on $f(n,m)$ by using the non-local-mean window, which is denoted by $w(n,m)$. Here, FNLM carries out the convolution operation between $f(n,m)$ and the overlapping bands of $w(n,m)$, which ultimately results in the generation of a noise-free output image denoted by u $o(n,m)$, respectively. In the following manner, Equation (2) depicts the process of FNLM:

$$u_o(k,l) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} w(k-n,l-m) * \ f(n,m) \tag{2}$$

The maximum allowed rows in this instance are N, while the maximum allowed columns are M. In addition, the most important part of the noise reduction process is the construction of the FNLM window $w(n,m)$, which uses linear exponential probability models to generate weight coefficients. This is where the majority of the work is done.

$$w(k-n,l-m) = \frac{1}{Z[n,m]} e^{-\frac{\left\| N_i[P_{n,m}] - N_j[P_{k,l}] \right\|_{2a}^{2}}{h}} \tag{3}$$

Here, $N_i$ , $N_j$ are the pixel patches that are located in the immediate neighborhood of the window. Further, $a$ represents the non-local gaussian kernel-based standard deviation. $Z[n,m]$ represents the zero mean of the window values. h represents the filtering parameter of the window. In addition, the weight ranges are chosen from those that fall within the stated range, such as 0 to $w(Ni,Nj)$ less than 1. In addition, the non-local kernel pixels keep the uniform property, such as $P_j \in N_i\ w\left(N_i\ , N_j\ \right) = 1$; this is the case because of the non-local nature of the kernel. In addition, $Z[n,m]$ may be obtained using the following formula:

$$Z[n,m] = \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} e^{-\frac{\| N[P_n,m] - N[P_k,l] \|^{2}{}_{2,a}}{h}} \tag{4}$$

In order to determine which pixel patches are comparable to one another, a calculation called the Euclidean distance between $N_i$ , $N_j$ patches are performed.

$$d = \left\| I(N_i) - I(N_j) \right\|_{2,a}^{2} \tag{5}$$

Then, the weight coefficients are revised using the aforementioned weight coefficients in the following manner:

$$w\left(N_i, N_j\right) = \frac{1}{Z(i)} e^{(-d)/h^2} \tag{6}$$

Finally, the denoising and enhancement procedure is completed by taking into account this updated $w\left(N_i, N_j\right)$.

$$u_o(k,l) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} w\left(N_i, N_j\right) * f(n,m) \tag{7}$$

### 3.2 Feature extraction

Features are the statistical properties of the images, which holds the color, shape, entropy and other forbidden characteristics of image. Further, the appropriate feature extraction will result in improvement of the classification performance. The proposed Tongue-Net model extracts the LBP based shape features, GLCM based texture features and mean, skewness, standard deviation-based color features.

### 3.2.1 GLCM feature extraction

Techniques for analyzing images include the general linear correspondence method and related texture feature computations. The GLCM is a tabulation that determines how frequently various combinations of grey levels co-occur in an image or image sub region as shown in Figure 3. Given an image that is composed of pixels, each of which has an intensity (a particular grey level). Then, the GLCM is applied to determine how often different combinations of grey levels co-occur. Calculations using texture features make use of the information contained within the GLCM to provide a measure of the change in image texture (also known as intensity) at the pixel of interest. Quantize the image data, so the value of each sample on the echogram is considered to be the intensity of the corresponding image pixel, and each sample is regarded as a single pixel on the echogram. After then, the intensities are further quantized into a certain number of discrete grey levels.

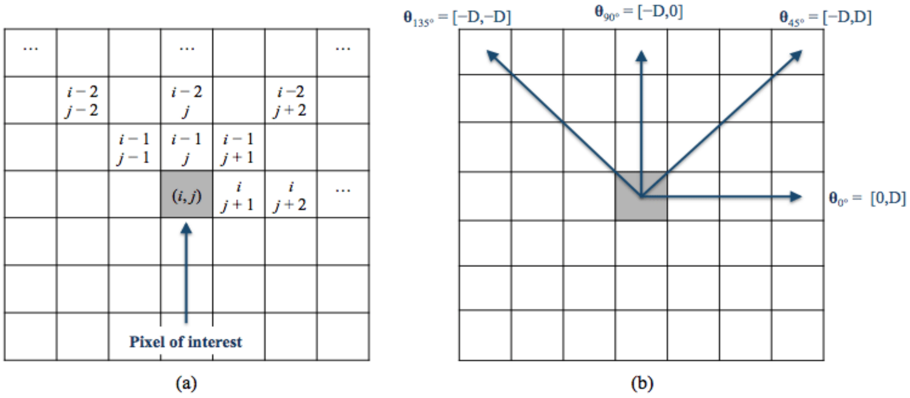


Figure 3. GLCM feature extraction, (a) spatial relationships of pixels; (b) the GLCM directions.

The GLCM dimensions will be $N$ squares on each side, where $N$ is the number of levels that was given under the quantization heading. Let us assume that $S$ is the sample that will be used for the computation. Let $W$ represent the collection of samples that surround sample s and that fit inside a window that is centered on sample s and has the size that was defined under the heading Window Size. Taking into account just the samples that are part of the set $W$ with elements $i, j$, where these two samples with intensities $i$ *and* $j$ occur in a certain spatial relationship (where $i$ and $j$ are intensities between 0 and $N-1$). The total number of instances in which the required spatial connection may be found in $W$ will be equal to the sum of all of the components $i, j$ that make up the GLCM. Create a symmetrical GLCM and continue the operation for all pixels. Create a duplicate of the GLCM that has been transposed with different phase angle. The GLCM will compute the surrounding pixels in $0^0$, $45^0$,$90^0$,$135^0$, and $180^0$ angles. This results in a symmetric matrix in which the connection $i$ to $j$ cannot be differentiated from the relationship $j$ to $i$ due to the symmetry of the matrix (for any two intensities $i$ and $j$). The components of the GLCM may now be interpreted as probabilities of discovering the connection $i, j$ (or $j, i$ in $W$. The value of this computed feature is substituted for the sample $S$ in the virtual variable that was produced as a consequence. The texture features extracted using GLCM are energy, entropy, contrast, homogeneity, and correlation.

$$Energy = \sum_{i,j=0}^{N-1} (P_{ij})^2 \quad (8)$$

$$Entropy = \sum_{i,j=0}^{N-1} -\ln(P_{ij}) P_{ij} \quad (9)$$

$$Contrast = \sum_{i,j=0}^{N-1} \left(P_{ij}(i-j)\right)^2 \quad (10)$$

$$Homogeneity = \sum_{i,j=0}^{N-1} \frac{P_{ij}}{1+(i-j)^2} \quad (11)$$

$$Correlation = \sum_{i,j=0}^{N-1} P_{ij} * \frac{(i-\mu)\,(j-\mu)}{\sigma^2} \quad (12)$$

Here, $P_{ij}$ is the elements of the symmetrical GLCM features. Here, $\mu$ is GLCM mean, it is a thought of an estimate of the intensity of each pixel that participated in the associations. This also approximates, but is not equivalent to, the mean of all the pixels in the data window $W$, and it is reliant upon the choice of spatial connection such as direction. Further, $\sigma^2$ is the variation in intensity across all reference pixels used in the relationships that were used to calculate the GLCM.

$$\mu = \sum_{i,j=0}^{N-1} i * P_{ij} \quad (13)$$

$$\sigma^2 = \sum_{i,j=0}^{N-1} P_{ij}(i-\mu)^2 \quad (14)$$

### 3.2.2. Color features

The color features hold the color intensities of tongue image, which are extracted using color moments. Moments of color are measurements that define the color distribution in an image in the same way as central moments uniquely describe a probability distribution. Color moments are similar to central moments in that they characterise the distribution of colors in an image. Color moments are most often used for the purpose of color indexing, where they serve as features in image retrieval applications and are used to evaluate the degree to which two images are comparable on the basis of color. In most cases, one image is compared to a database of digital images that have already had their characteristics calculated in order to discover and get an image that is similar to the one being searched for. Following each comparison of two diseases, a similarity score is generated; the lower this value is, the greater the likelihood that the two images have a high degree of resemblance.

Color moments are the three most prominent points in an image's color distribution that are used. The mean ($\mu_c$), standard deviation ($\sigma_c$), and skewness are their respective features. The value of a color that is considered to represent the image's pixels average is referred to as the mean. The square root of the variance of the distribution is the analysis for calculating the standard deviation. A measure of the degree to which the distribution is not symmetrical is referred to as the skewness of the distribution. The value of a color that is considered to represent the image's mean is referred to as the mean. At least three different values may be used to define a color. In an image, moments are computed for each of these channels individually. Therefore, an image is defined by 9 moments, with 3 moments corresponding to each of its three-color channels. After that, the three-color moments may be characterized as follows:

$$\mu_c = \sum_{i,j=0}^{N-1} \frac{1}{N} * P_{ij} \quad (15)$$

$$\sigma_c = \sqrt{\frac{1}{N}\left(\sum_{i,j=0}^{N-1} (P_{ij} - \mu_c)^2\right)} \quad (16)$$

$$S_i = \sqrt{\frac{1}{N} * \left(\sum_{i,j=0}^{N-1} (P_{ij} - \mu_c)^3\right)} \quad (17)$$

### 3.2.3. LBP feature extraction

The LBP operator is an efficient method for describing textures, and it possesses remarkable properties such as rotation invariance and grayscale invariance. At the same time, it eliminates the issue of illumination changes to a certain extent, making it one of the most advantageous operators available. As can be seen in Figure 4, the threshold value for the first LBP operator is the pixel value of the center point in the region of a window that is 3 by 3, and the grey value of each of the neighboring eight-pixel points is compared one at a time. This is done in a window that is 3 by 3. If the points that surround the center point are less significant than the center point, then it becomes 0; otherwise, it becomes 1

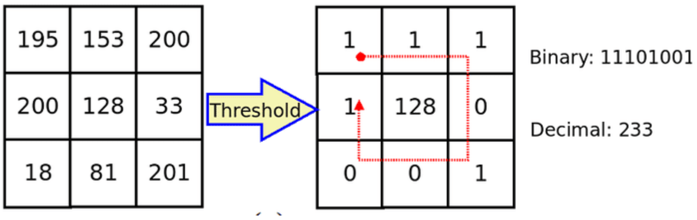


Figure 4. LBP operation example.

. The LBP operator will then transform the values of 0 and 1 that were acquired from each pixel point into the appropriate decimal number, which is the pixel value of the current center point. This process is repeated until the pixel value of the current center point has been achieved. To be more specific, the fundamental LBP operator may be described as follows:

$$LBP(P,R) = \sum_{Q=0}^{P-1} S(g_q - g_c) * 2^p \tag{18}$$

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{19}$$

Here, $P$ is the number of sampling points, $R$ is the sampling radius, $g_C$ is the grey value of the central pixel in the local area, and $g_q$ is the grey value of the qth sampling point in the neighborhood of the central pixel. Here, $P$ is the number of sampling points, $R$ is the sampling radius, and $g_C$ is the grey value of the central pixel. A coding mode consisting of a combination of 0 and 1 may be generated by an LBP operator. The quantity of calculations that need to be done will continue to grow as the number of sample points is increased, which will also bring about a rise in the amount of information that is redundant. Further, an equivalent model is proposed to reduce the dimension of the model types of the LBP operator in order to solve such a problem and improve the efficiency of the operator. As a result of this proposal, the number of models was decreased from the initial $2p$ to $p\ (p\ -\ 1)\ +\ 3$, which is a significant improvement. It is possible to specify the LBP mode value of the comparable mode in the following way:

$$LBP1(P,R) = \begin{cases} \sum_{Q=0}^{P-1} S(g_q - g_c) * 2^p\ , & U\big(LBP\ (P,R)\big) < 2 \\ 0,\ otherwise \end{cases} \tag{20}$$

In this context, the number of transitions between 0 and 1 in the basic LBP mode is denoted by the expression $U(LBP\ (P,R))$. It is necessary to divide the LBP feature image into m local blocks, extract the histogram of each local block, and then connect these histograms in turn to form the statistical histogram of LBP features, which is referred to as LBPH. Finally, it is necessary to use the machine learning method to train the LBP feature vector for image detection. This will allow the LBP features to be converted into information that can be used.

### 3.3 DLCNN classification

The use of DL methods is an effective way to resolve a large number of the classification and optimization issues that arise in computer vision applications due to the presence of restrictions. The DLCNN is one of the best solutions for the classification process, which takes the local features from higher inputs and combines them into more complex features at lower levels. This technique is one of the finest options for the classification process. In addition, the functionality of the DLCNN is enhanced by synchronizing the weights and kernel sizes of the network with the local connections. The DLCNN-based Tongue-Net model can be found shown in Figure 5. As conducting multi-class classification, stacked ensemble learning is utilized to stack the Tongue-Net architecture. This provides in improved results when compared to those achieved by utilizing a single model. The categorization process is carried out by combining all of the layers into one and proceeding as follows:

**Convolution layer:** This layer is a crucial operational block in DLCNN. Its purpose is to create the local features by performing the convolution operation between the extracted feature and the weight matrix. In this case, the characteristics of the weight matrix are determined by the size of the kernel and the activation function. The mathematical procedure known as convolution layer is described in the following way:

$$F(i,j) = (I * K)(i,j) = \sum_m^M \sum_n^N I(i+m, j+n) K(m,n) \tag{21}$$

In this particular instance, the input image or matrix is designated by $I$, the 2D filter is denoted by $K$ with $i,j$ as the filter size, and the output of the 2D feature map is marked by the letter F. The $F$ value is produced as a result of a convolution operation that is carried out between $I$ and $K$. The output that is produced by the convolution layer is then sent via a Rectified Linear Unit (ReLU) based activation function, which is what establishes the non-linear connection between the different characteristics. The ReLU starts with the assumption that the threshold value is zero, and then it compares that value to the input. If the input feature is more than zero, this would lead to the output being greater than zero; otherwise, the input would be the output. The following is an indication of the mathematical analysis that was done on the ReLU activation function:

$$f(x) = \max(0, x) \tag{22}$$

**MaxPooling Layer:** In the DLCNN environment, this layer is utilized for down-sampling, which is used to lower the input spatial size and also reduce the network parameters by a factor of two. Down-sampling is used for the following purposes: The MaxPooling layer extracts the data by searching the input feature range maximum and picking the best feature properties, in contrast to the AvgPool and L2-Norm pooling layers, which were losing the feature characteristics.

**Batch Normalization layer:** It is difficult to train DLCNN models that have tens of layers because these networks are often sensitive to the initial random weights that are used and the design of the learning algorithm. When the weights are adjusted after each mini batch, the distribution of the inputs to the layers deep in the network could shift, which might be one of the reasons why this challenge is so tough to solve. Because of this, the learning system could end up chasing a moving object indefinitely. The term "internal covariate shift" is the formal name given to this alteration in the distribution of inputs to layers in the network. Standardizing the inputs to a layer for each mini batch is the goal of the batch normalization method, which is a methodology for training very deep neural networks. This has the effect of making the learning process more consistent and significantly cutting down on the number of training epochs that are necessary to train DLCNN models.

**Flatten Layer:** In order to create a column-wise feature map from the input pooled three-dimensional feature map, a flatten layer must first be applied. This design has a number of pooling layers, each of which consists of a number of pooled feature maps that are arranged in a sequence. As a result, they are arranged in this layer in the form of a single long column, consecutively, one after the other. This layer's primary function is to combine all CXR characteristics into a single vector for easier analysis.

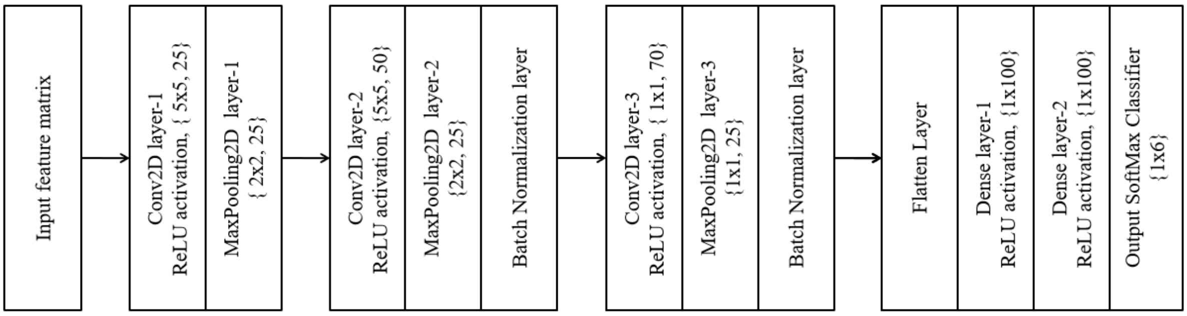

Input feature matrix
Conv2D layer-1 ReLU activation, { 5x5, 25}
MaxPooling2D layer-1 { 2x2, 25}
Conv2D layer-2 ReLU activation, {5x5, 50}
MaxPooling2D layer-2 {2x2, 25}
Batch Normalization layer
Conv2D layer-3 ReLU activation, { 1x1, 70}
MaxPooling2D layer-3 {1x1, 25}
Batch Normalization layer
Flatten Layer
Dense layer-1 ReLU activation, {1x100}
Dense layer-2 ReLU activation, {1x100}
Output SoftMax Classifier {1x6}

Figure 5. DLCNN model of Tongue-Net.

**Dense Layer:** The dense layer is an output layer that is used to produce all of the available interconnections between the neurons of the preceding layer and the neurons of the next layer. Therefore, each and every neuron is involved in the categorization process in some way. The matrix vector multiplication operation is carried out by performing it between the row vector neurons of the layer that came before it and the column vector neurons of the layer.

**Classification process-SoftMax Classifier:** To create a multi-class DLCNN model, each of the layers that are part of the proposed architecture for Tongue-Net are layered on top of one another. The DLCNN model incorporates a SoftMax classifier as a means of simplifying the process and cutting down on the amount of time spent in training. This classifier is also capable of categorizing the various tongue illnesses. The filter sizes are progressively increased during the training phase as a result of this effort. The classifier is made up of a bias vector, weight matrix, and activation function, and its purpose is to carry out the classification operation. The following is a definition of the mathematical connection that exists between these properties:

$$Output = ReLU(dot(input, kernel) + bias) \tag{23}$$

Here, $Output$ holds the classes of different tongue diseases.

## 4. Results and discussion

This section gives the detailed analysis of simulation results and performance comparison with conventional approaches. Further, the proposed Tongue-Net method and existing classifiers used the same dataset for evaluating the performance.

### 4.1 Dataset

The dataset contains the healthy, appendicitis, bronchitis, gastritis, heart disease, and pancreatitis disease classes. There is no standard dataset for disease prediction using tongue images. The dataset is collected by various sources and followed the rules of IAM for disease indexing. Further, data augmentation is used to increase the number of samples during training, which can improve the performance of proposed Tongue-Net method. The data augmentation performs image morphological operations, rotation, scaling, resizing, compression, flipping and masking operations. Table 1 presents the number of images presented in the dataset with respect to each disease class.

### 4.2 Performance evaluation

This section compares the performance evaluation of proposed method. Figure 6 presents the confusion matrices of SVM [13], Random Forest [12], and proposed Tongue-Net. The conventional methods resulting in the smaller number of true positives, and increased in false negative, whereas the proposed Tongue-Net resulted in higher true positive values and less false predictions. Table 2 presents the performance comparison of proposed method with conventional machine learning approaches like SVM [13], Random Forest [12], TDAS [17] and SVM-RFE [20]. Further, the proposed Tongue-Net resulted in the superior performance for all metrics as compared to conventional approaches.

Table 1. Dataset description.

| Class | healthy | appendicitis | Bronchitis | gastritis | heart | pancreatitis |
|---|---|---|---|---|---|---|
| Number of images | 105 | 126 | 105 | 126 | 146 | 105 |
| Total | | | 713 | | | |

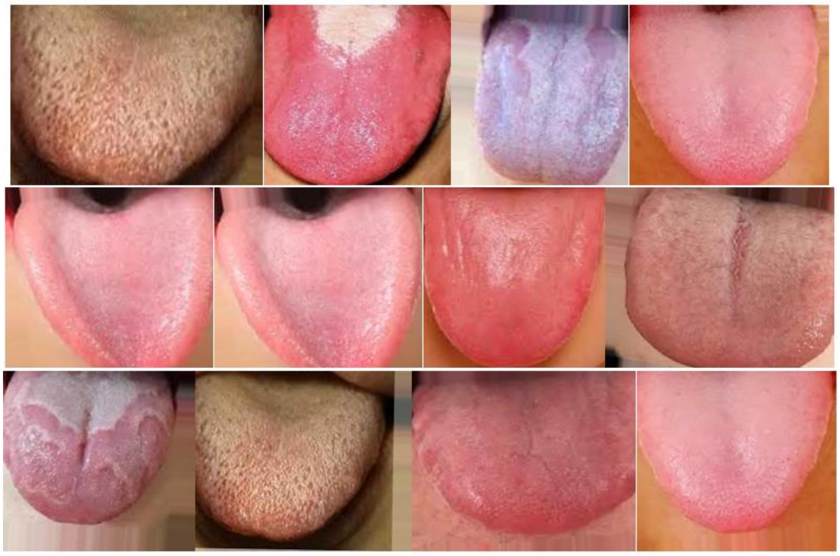
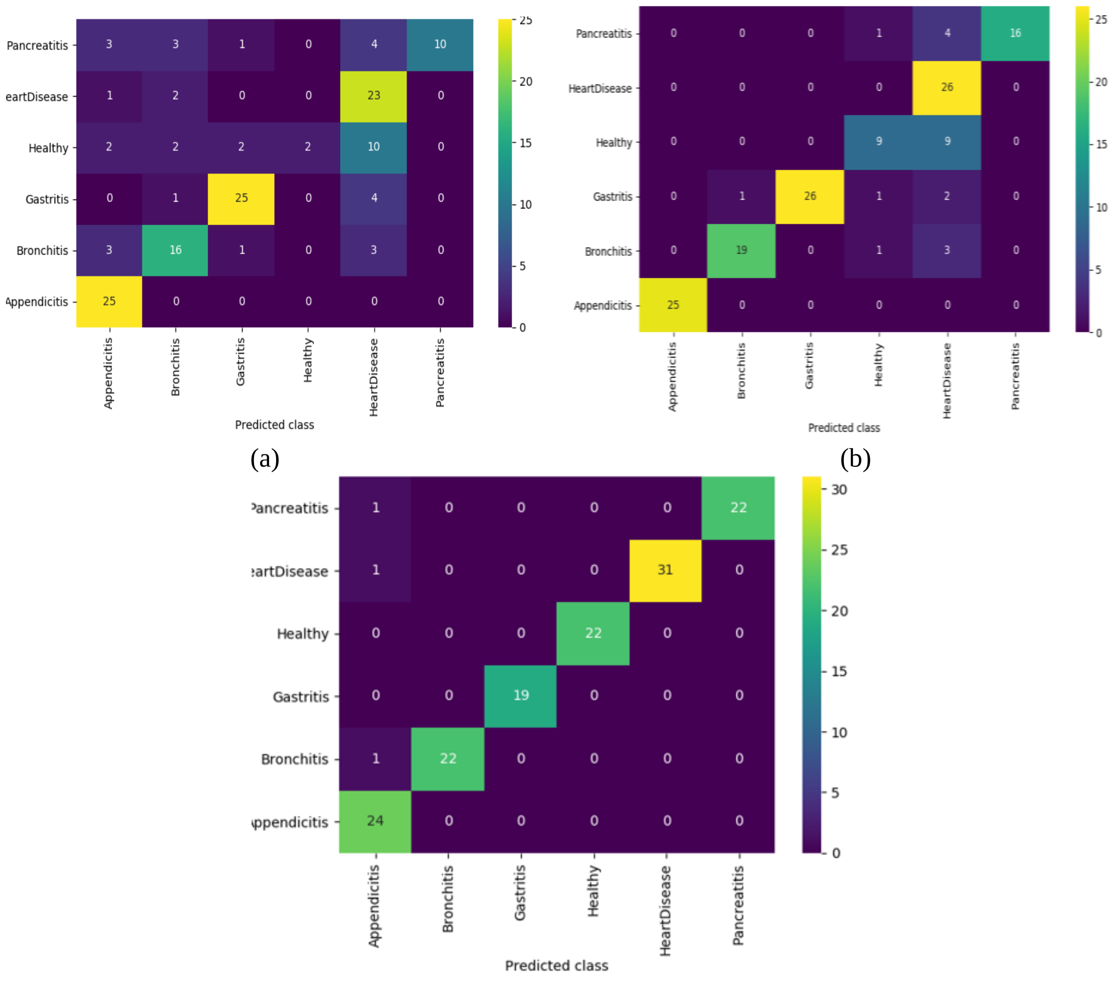


Figure 6. Sample tongue image dataset labeled with various diseases.

Table 2. Obtained classification quality metrics using existing and proposed Tongue-Net

| Method | Accuracy | Precision | Recall | F-score | Sensitivity | Specificity |
|---|---|---|---|---|---|---|
| SVM [13] | 70.62 | 79.77 | 66.68 | 64.63 | 100 | 84.21 |
| Random Forest [12] | 84.61 | 88.18 | 82.57 | 83.66 | 100 | 100 |
| TDAS [17] | 86.27 | 89.30 | 84.38 | 84.28 | 100 | 100 |

| | | | | | | |
|---|---|---|---|---|---|---|
| SVM-RFE [20] | 90.392 | 90.19 | 86.34 | 86.33 | 100 | 100 |
| Proposed Tongue-Net | 97.90 | 98.14 | 98.02 | 98.01 | 100 | 95.65 |



(a) (b)

(c)

Figure 7. Confusion matrices, (a) SVM [13], (b) Random Forest [12], (c) Proposed Tongue-Net.

## 5. Conclusion

This work implemented the Tongue-Net model according to the IAM principles, which classified the healthy, appendicitis, bronchitis, gastritis, heart, and pancreatitis disease classes using DLCNN classifier. Initially, FNLM was applied on tongue images to remove the different types noises like salt-pepper, gaussian, random noises, which also performed the color enhancement operation. Further, joint texture and color features are extracted from preprocessed image using color moments and GLCM, LBP descriptors. Here, texture features extracted using GLCM are energy, entropy, contrast, homogeneity, correlation, and LBP extracts the shape related features. In addition, the mean, standard deviation, and skewness are their respective color features. Finally, DLCNN is used to classify the diseases using extracted features. The simulation results shows that the proposed Tongue-Net outperformed as compared to existing methods. Further, this work can be extended with transfer learning models for better accuracy.

## References

[1]. Shi, Dan, et al. "An annotated dataset of tongue images supporting geriatric disease diagnosis." Data in brief 32 (2020): 106153.

[2]. Xu, Qiang, et al. "Multi-task joint learning model for segmenting and classifying tongue images using a deep neural network." *IEEE journal of biomedical and health informatics* 24.9 (2020): 2481-2489.

[3]. Gholami, Elham, and Seyed Reza Kamel Tabbakh. "Increasing the accuracy in the diagnosis of stomach cancer based on color and lint features of tongue." *Biomedical Signal Processing and Control* 69 (2021): 102782.

[4]. Wen, G., Wang, K., Li, H., Huang, Y., & Zhang, S. (2021). Recommending prescription via tongue image to assist clinician. *Multimedia Tools and Applications*, *80*(9), 14283-14304.

[5]. Braz, D. C., Neto, M. P., Shimizu, F. M., Sá, A. C., Lima, R. S., Gobbi, A. L., & Oliveira Jr, O. N. (2022). Using machine learning and an electronic tongue for discriminating saliva samples from oral cavity cancer patients and healthy individuals. *Talanta*, *243*, 123327.

[6]. Dulam, S., Ramesh, V., & Malathi, G. (2020). Tongue image analysis for COVID-19 diagnosis and disease detection. International Journal of Advanced Trends in Computer Science and Engineering, 7924-7928.

[7]. Ma, Jiajiong, et al. "Complexity perception classification method for tongue constitution recognition." *Artificial intelligence in medicine* 96 (2019): 123-133.

[8]. Xu, Qiang, et al. "Multi-task joint learning model for segmenting and classifying tongue images using a deep neural network." *IEEE journal of biomedical and health informatics* 24.9 (2020): 2481-2489.

[9]. Lin, Huiping, et al. "Automatic detection of oral cancer in smartphone-based images using deep learning for early diagnosis." *Journal of Biomedical Optics* 26.8 (2021): 086007.

[10]. Jiang, Tao, Xiao-jing Guo, Li-ping Tu, Zhou Lu, Ji Cui, Xu-xiang Ma, Xiao-juan Hu et al. "Application of computer tongue image analysis technology in the diagnosis of NAFLD." *Computers in Biology and Medicine* 135 (2021): 104622.

[11]. Heo, J., Lim, J. H., Lee, H. R., Jang, J. Y., Shin, Y. S., Kim, D., ... & Kim, C. H. (2022). Deep learning model for tongue cancer diagnosis using endoscopic images. *Scientific reports*, *12*(1), 1-10.

[12]. Alabi, Rasheed Omobolaji, et al. "Comparison of nomogram with machine learning techniques for prediction of overall survival in patients with tongue cancer." *International Journal of Medical Informatics* 145 (2021): 104313.

[13]. Shan, Jie, et al. "Machine learning predicts lymph node metastasis in early-stage oral tongue squamous cell carcinoma." *Journal of Oral and Maxillofacial Surgery* 78.12 (2020): 2208-2218.

[14]. Alabi, Rasheed Omobolaji, et al. "Machine learning application for prediction of locoregional recurrences in early oral tongue cancer: a Web-based prognostic tool." *Virchows Archiv* 475.4 (2019): 489-497.

[15]. Shamim, Mohammed Zubair M., et al. "Automated detection of oral pre-cancerous tongue images using deep learning for early diagnosis of oral cavity cancer." *The Computer Journal* 65.1 (2022): 91-104.

[16]. Wang, X., Liu, J., Wu, C., Liu, J., Li, Q., Chen, Y., Wang, X., Chen, X., Pang, X., Chang, B. and Lin, J., 2020. Artificial intelligence in tongue diagnosis: Using deep convolutional neural network for recognizing unhealthy tongue with tooth-mark. Computational and structural biotechnology journal, 18, pp.973-980.

[17]. Li, J., Yuan, P., Hu, X., Huang, J., Cui, L., Cui, J., ... & Xu, J. (2021). A tongue features fusion approach to predicting prediabetes and diabetes with machine learning. *Journal of biomedical informatics*, *115*, 103693.

[18]. Balasubramaniyan, Saritha, Vijay Jeyakumar, and Deepa Subramaniam Nachimuthu. "Panoramic tongue imaging and deep convolutional machine learning model for diabetes diagnosis in humans." *Scientific Reports* 12.1 (2022):

[19]. Mansour, Romany F., Maha M. Althobaiti, and Amal Adnan Ashour. "Internet of Things and Synergic Deep Learning Based Biomedical Tongue Color Image Analysis for Disease Diagnosis and Classification." *IEEE Access* 9 (2021): 94769-94779.

[20]. Fan, Shangyong, et al. "Machine learning algorithms in classifying TCM tongue features in diabetes mellitus and symptoms of gastric disease." *European Journal of Integrative Medicine* 43 (2021): 101288.

[21]. Hu, Yang, et al. "Fully-channel regional attention network for disease-location recognition with tongue images." *Artificial Intelligence in Medicine* 118 (2021): 102110.

[22]. Li, H., Wen, G., & Zeng, H. (2019). Natural tongue physique identification using hybrid deep learning methods. *Multimedia Tools and Applications*, *78*(6), 6847-6868.

[23]. Wen, G., Ma, J., Hu, Y., Li, H., & Jiang, L. (2020). Grouping attributes zero-shot learning for tongue constitution recognition. *Artificial Intelligence in Medicine*, *109*, 101951.

[24]. Li, Jun, Qingguang Chen, Xiaojuan Hu, Pei Yuan, Longtao Cui, Liping Tu, Ji Cui et al. "Establishment of noninvasive diabetes risk prediction model based on tongue features and machine learning techniques." *International Journal of Medical Informatics* 149 (2021): 104429.

[25]. Balu, S., & Jeyakumar, V. (2021). A study on feature extraction and classification for tongue disease diagnosis. In *Intelligence in Big Data Technologies—Beyond the Hype* (pp. 341-351). Springer, Singapore.