International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Novel Deep Learning-Based Framework for Efficient Content-Based Medical Image Retrieval

Vasudeva R.[1], Dr. S. N. Chandrashekara[2]

Submitted: 27/12/2023 **Revised**: 03/02/2024 **Accepted**: 11/02/2024

Abstract: In clinical research and decision-making, medical image retrieval is essential. In this article, we provide a brand-new framework for retrieving content-based medical images that is based on deep learning. The framework is designed to retrieve relevant medical images from five distinct categories: 'Breast Cancer', 'Tuberculosis', 'Alzheimer's Disease', 'Brain Tumor', and 'COVID-19'. We employ the Gray Levels Co-occurrence Matrix (GLCM) & concentrate on six distinct aspects: "dissimilarity," "correlation," "homogeneity," "contrast," "ASM" (Angular Second Moment), & "energy" in order to extract significant information from the medical images. These features provide important insights into the texture and structure of the images, enabling effective discrimination between different medical conditions. For training the retrieval framework, we employ two models: Artificial Neural Network (ANN) and the Convolutional Long Short-Term Memory (CLSTM) hybrid deep learning model. The ANN model achieves an accuracy of 91% in classifying the medical images, while the CLSTM model outperforms it with an accuracy of 99.01%. Our test results show how well the suggested framework works at quickly retrieving pertinent medical photos. The integration of deep learning techniques enhances the accuracy of image classification and improves the retrieval performance. The framework has potential applications in medical research, diagnosis, and treatment planning by enabling quick and accurate access to relevant medical images for specific conditions.

Keywords *Medical image retrieval, deep learning, content-based retrieval, GLCM, ANN, CLSTM, breast cancer, tuberculosis, Alzheimer's disease, brain tumor, COVID-19.*

1. Introduction

The utilization of extensive image databases has witnessed a significant surge in recent years, the development of multimedia technologies. Image retrieval has become a crucial technique for image processing applications in this environment, with the widespread usage of Content-Based Images Retrieval (CBIR) in numerous domains. However, relying solely on a single image feature often yields unsatisfactory outcomes. Consequently, the integration of multiple image features has become a common strategy to achieve enhanced results. Nevertheless, the efficient and swift retrieval of relevant images from such databases presents a formidable challenge in previous iterations of this system [1]. CBIR has used an integrated feature extraction method that incorporates local energy, rotation-invariant universal local binary patterns (RULBP), & color auto-correlogram.

The effective retrieval of images from large databases that are pertinent to a given Query Image (QI) is made possible by Content-Based Images Retrieval (CBIR) systems. Current CBIR systems exhibit limitations by extracting only a limited set of features. This study presents a novel approach that involves extracting comprehensive and robust features from an image database, storing them as feature vectors in a repository. These encompass color signatures, shape attributes, and texture characteristics. The process involves extracting features from a specific QI and subsequently employing a novel similarity evaluation comparing QI attributes with those of databases photos utilizing a metaheuristic approach (genetic algorithm plus simulated annealing). By leveraging distance metrics, the system facilitates the search for related images when a QI is introduced from the database. This framework advances the core concept of CBIR, enabling enhanced retrieval performance [2].

In order to get photographs matching the user's query within the database, it is necessary to require significant computer resources due to the multimedia material provided by devices & image processing techniques. The traditional annotation-based technique for image retrieval lacks coherence due to its pixel-wise image matching, leading to notable disparities in patterns, storage, and angles.

In these situations, Content-Based Images Retrieval (CBIR) emerges becoming a more popular alternative. CBIR effectively gauges the resemblance between database and query images. This method entails gathering images resembling the query image from an extensive database and extracting more informative attributes from the query image [3]. Subsequently, it establishes relationships and matches these attributes with those of

[1*]*Research Scholar, Department of Computer Science and Engineering, C Byregowda Institute of Technology, Kolar, Karnataka, India. Email: vasudev.rama@gmail.com*

[2] *Professor and Head, Department of Computer Science and Engineering, C Byregowda Institute of Technology, Kolar, Karnataka, India. snc.boe.cse@gmail.com*

database images, ultimately retrieving and presenting images with similar features. This approach offers a more efficient means of image retrieval, aligning with the underlying principles of CBIR.

Surveillance videos find widespread application for security purposes globally. Within these videos, object detection has gained paramount importance, yet comprehensively analyzing such videos and extracting pertinent information is arduous [4]. In the realm of agriculture, managing potato diseases is pivotal due to potential crop yield losses. Timely recognition and classification of these diseases are essential to mitigate losses, though the process is time-intensive and reliant on human intervention. Therefore, a precise automated method is imperative to promptly detect and classify these diseases [5].

In the context of burgeoning video content, video summarization is pivotal. It seeks to distill the essential events from lengthy videos into concise highlights. Given the time-consuming nature of storing, browsing, and sharing videos, this task is increasingly challenging. In this work, SumVClip, a flexible method that creates generic video summary clips, is introduced. To create these video, key frames are chosen using deep learning classification employing methods such as AlexNet [6]. The proposed approach addresses the demand for efficient video summarization in the face of expanding video content.

2. Literature Review

To improve image representation, a variety of image retrieval techniques utilizing low-level descriptors of features have been developed. Both global & local feature representations can be generally used to categorize these techniques. Color [7], edge [8], texture [9], and GIST Centrist are examples of global features. Local feature representations include models like the bag of words [10] and descriptors such SWIFT [11], SURT [12], GLCM [1].

Tunga et al. unveiled a strategy to picture retrieval that makes use of machine learning algorithms, treating image categories as semantic concepts [13]. Their approach computes similarity solely based on images falling into the exact same class as the query image, rather than matching the query picture with every image in the database. This approach reduces the computational burden and improves retrieval efficiency. Another significant contribution is the mapping learning scheme proposed by Singh [14], which facilitates efficient retrieval in large-scale image applications. Singh's method maps while maintaining semantic similarity, convert highly dimensional information into binary codes. This approach enables effective image retrieval with reduced memory requirements and computational costs.

The content-based retrieval of images (CBIR) system was introduced by Kumar et al. [15] and integrates SIFT, ORB, K-Means grouping, & LPP dimensionality reduction techniques. By employing these techniques, their system achieves improved retrieval performance by effectively capturing and representing the underlying visual features of the images. Liu et al. a two-layer codebook features-based picture retrieval technique was presented [16]. Their method combines high-level data obtained from the GoogLeNet deep convolutional network with low-level characteristics produced by Dot-Diffused Block Truncation is Coding (DDBTC), such as texture and color. By integrating these complementary feature representations, their method enhances the discrimination and retrieval accuracy of the system.

A deep belief network (DBN) is used by Saritha et al. to offer a content-based image retrieval framework that effectively learns feature representations [17]. By combining characteristics such as color histogram, edge, edge directions, & other features, their method takes a multi-feature approach. These characteristics are retrieved and saved as compact signature files, making it possible to compare and retrieve related photos quickly using their signatures. The similarity metric takes into account the separations between various features, which have been normalized using suitable weights. Wan et al. presented a deep learning framework for CBIR tasks, specifically using convolutional neural networks (CNNs) [18]. Three schemes make up their strategy. The first method uses CNNs' fully connected layers to directly represent features. By treating relationships between instances belonging to a single class as important & those belonging to separate classes as irrelevant, the second scheme improves similarity learning. The third technique, which improves the model's performance in certain retrieval tasks, refines it by retraining a network that was previously trained on a new dataset.

Wang et al. conducted a comprehensive examination of feature representations taken from several CNN layers in their CBIR model [19]. Using the IMAGENET 2012 dataset and the pretrained AlexNet model, they assessed their study. They discovered that the characteristics taken out of completely connected layers, specifically fc4096a and fc4096b, performed exceptionally well. These layers, which directly follow the stack of convolutional and pooling layers, demonstrated superior generalization ability for datasets with unseen categories. The authors further trained the network on a new dataset to fine-tune its performance. Additionally, they found that cosine similarity outperformed Euclidean similarity in their experiments. In order to develop compact binary algorithms for effective picture retrieval for large-scale datasets, Liu et al. suggested a unique Deep Supervision

Hashing (DSH) method utilizing convolutional neural networks [20]. By employing a supervised learning framework, DSH effectively preserves the semantic similarity between images while reducing the storage requirements and computational complexity associated with large-scale image retrieval.

Researchers have developed various image retrieval methods [21] focusing on low-level feature descriptors. These methods employ different techniques, including machine learning algorithms, deep learning architectures, and feature fusion, to increase the retrieval efficiency and accuracy. The adoption of these methods aids in the development of content-based picture retrieval and has potential applications in fields such as medical research, diagnostics, and treatment planning.

The research gap lies in the development of a system for retrieving medical images using deep learning that addresses the unique challenges of medical image analysis, incorporates advanced feature fusion techniques, utilizes large-scale medical image datasets, and focuses on specific medical image categories. Such a framework would significantly contribute to clinical decision-making, research, and healthcare applications by improving the retrieval of medical images' precision and effectiveness.

3. Proposed Work

In this article, we provide a brand-new framework for retrieving content-based medical images that is based on deep learning. The framework is designed to retrieve relevant medical images from five distinct categories: 'Breast Cancer', 'Tuberculosis', 'Alzheimer's Disease', 'Brain Tumor', and 'COVID-19' of 4525 Medical images data used in this work. To extract meaningful features, The Gray Level Co-occurrence Matrix (GLCM) is used, and we concentrate on properties like "dissimilarity," "correlation," "homogeneity," "contrast," "ASM," and "energy." Convolutional Long Short-Term Memory (CLSTM) and Artificial Neural Network (ANN) are two models that were used are employed for training. Experimental results demonstrate the framework's effectiveness in retrieving relevant medical images and improving retrieval performance. The proposed system has potential applications in medical research, diagnosis, and treatment planning.

3.1 GLCM-Based Feature Extraction

The GLCM algorithm 1 is employed in this study to perform feature extraction operations on a specified dataset. This algorithm effectively extracts highly informative features, particularly texture features, which are crucial for image analysis and provide valuable regions of interest. GLCM is a vital approach for feature extraction and is widely used in a variety of texture analysis applications. For tasks such as retrieval and indexing, GLCM captures the visual contents of an image. To solve computational problems in many applications, the collected GLCM data is used. For a CBIR (Content-Based Image Retrieval) system, an effective feature extraction method that includes both text-based data like annotations and keywords as well as visual elements like shape, texture, and color is crucial. Features of the texture that are independent of the intensity or appearance of one color indicate the homogeneity attributes. Effective use of color characteristics allows for the recognition of important visual features. Color is a crucial visual component of CBIR and is processed in a straightforward and effective manner. The rows & columns in a matrix are determined by the GLCM image matrix, which also specifies the number of gray levels (G). two isolated pixels separated by a pixel distance (x, y), and having a same relative frequency The matrix element P(i, j | x, y) defines the intensities 'i' and 'j'. The second-order statistics probabilities for variations among gray levels 'i' and 'j' at a particular angle () and displacement distance (d) are represented by the matrix's element P(i, j | d,). The use of high-intensity values (G) denotes the storing of a significant amount of temporary data, denoting a G G matrix for each fusion of (x, y) or (d,). The size of the texture sample significantly impacts GLCM due to its high dimensionality, leading to a reduction in the count of gray levels. The relationship between a reference pixel (i) and a neighboring pixel (j) is represented using GLCM.

Algorithm 1: Steps in GLCM for Feature Extraction

1. Input as query image and dataset of medicalimages

2. Separate R, G, B images planes.

3. Steps 3 to 5 are repeated.

4. Compute GLCM Matrices using Eq. (1)

5. Using Eq (2) to (7) Statistical features are computed for every GLCM matrix

in four dissimilar angles.

6. The variance and mean values of all parameters are utilized to calculate the feature vector.

7. Mine the more texture instructive features from the images

The well-known statistical tool GLCM is used to extract second-order texture data from photographs. The rate of occurrence of a horizontally adjacent pair of pixels with values of i and j is represented by each element (i, j) in the GLCM. Typically, a pixel with value j shows horizontally next to a picture with a gray-level (gray-scale potency) measurement of i. To calculating texture feature calculations, the GLCM's contents are used to measure intensity fluctuations at particular pixel positions. Two factors are used to calculate the co-occurrence matrix: the relative orientation & the relative separation, expressed in pixel units, among the pixel pairings. Generally, these parameters are quantized in four directions (0°, 45°, 90°, and 135°), although alternative quantization directions are also possible. There are fourteen features in the GLCM, the most valuable of which are the angular second moment (ASM), correlation's informational measure, sums the entropy inverse differential moment, and correlation. In terms of

texture analysis, these traits are very promising.

The normalized probability density, denoted as P_δ(i, j), is defined for co-occurrence matrices as follows in Equation (1):

$$P_\delta(i,j) = \frac{\#\{|(x,y),(x+d,y+d)|\epsilon S|f(x,y)=i,f(x+d,y+d)=j\}}{\#S} \quad (1)$$

In the context of the defined parameters:

- The coordinates of the pixel are denoted by x and y, ranging from 0 to N−1.
- The gray levels are represented by i and j, ranging from 0 to L−1.
- S stands for the collection of pixel pairs in the image that have certain relationships.
- #S denotes how many elements there are in S.

$$E = \sum_{i,j=0}^{N-1} p_{i,j}^2 \quad (2)$$

In GLCM, the count of gray-level co-occurrence matrices is represented as p.

The textured surface of the input image is described using the entropy Equation (3), an arbitrary measure of randomness.

$$\text{Entropy} = -\sum\sum p(i,j)\log\, p(i,j) \quad (3)$$

By calculating the intensity difference between a pixel & its neighboring pixel over the course of the entire image, difference in GLCM detects local fluctuations. The contrast value for an unchanging image is 0 Equation (4).

$$\text{Contrast} = \sum\sum (i-j)^2 p(i,j) \quad (4)$$

The pixel at location (i, j) is represented as p (i, j).

The probability that two specific pixel pairings will occur together is measured by correlation Equation (5).

$$\text{Correlation} = \frac{\sum_{i=0}^{G-1}\sum_{j=0}^{G-1}(i-\mu_i)(j-\mu_j)p(i,j)}{\sigma_i\sigma_j} \quad (5)$$

The degree of element distribution homogeneity Equation (6) along the GLCM diagonal is quantified.

$$\text{Homogeneity} = \sum_{i,j} \frac{P(i,j)}{1+|i-j|} \quad (6)$$

$$\text{Dissimilarity} = \sum_{i,j=0}^{N-1} P_i, j|i-j| \quad (7)$$

The first element in Equation (7) denotes the vertical coordinate, and the second element denotes the horizontal coordinate.

- The likelihood that the initial pixel is given the intensity value i and the subsequent pixel is given the brightness value j, spaced by a specified distance, is denoted by the symbol P(i, j).

The Energy (E) value is employed to measure the extent of repetition of pixel pairs, serving as a measure of image uniformity. High similarity pixels are indicated by high energy values in the pixels. The computation looks like this Equation (2):

All of these traits have a strong ability to discriminate between different types of images. In this study, second-order texture GLCM is used to extract statistical texture features. Six second-order features are specifically computed: contrast, dissimilarity, homogeneity, correlation, entropy, & energy. Contrast value gauges local level differences, while energy value measures image smoothness. High contrast images yield higher values, whereas low contrast images yield lower values. Homogeneity, measured within the GLCM, indicates the closeness of element distribution and ranges from 0 to 1, with a value of 1 for diagonal GLCM. Entropy serves as a measure of randomness. Dissimilarity effectively captures the differences between further information about the kinds of cancer or classes they most likely belong to is provided by picture features. Therefore, more useful characteristics can be retrieved from the provided weed database using the GLCM approach and the energy, homogeneity, entropy, correlation, dissimilarity, & contrast texture features.

3.2. Artificial Neural Network (ANN)

A computational model called an Artificial Neural Network (ANN), commonly referred to as a neural network, and is one that is motivated by the composition and operation of the human brain. It is an effective machine learning method utilized for many different tasks, including recognizing patterns, regression analysis,

classification, & data production. Artificial neurons or nodes, which are interconnected processing units, make up an ANN. Fig 1 shows the Workflow model of ANN.

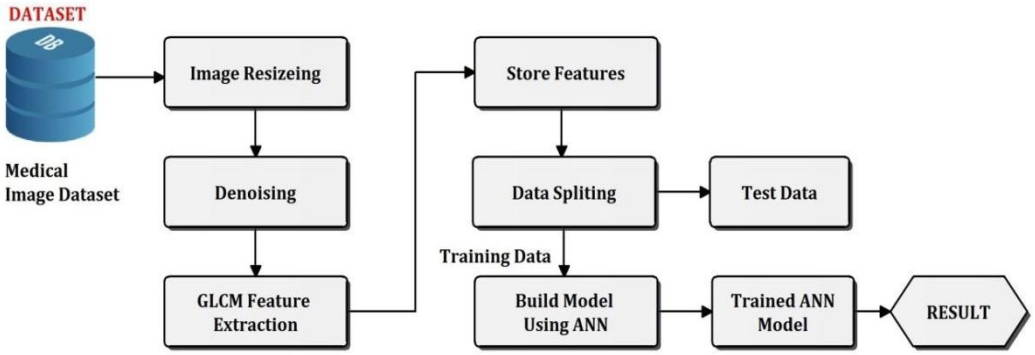


Fig 1: Workflow Model of ANN

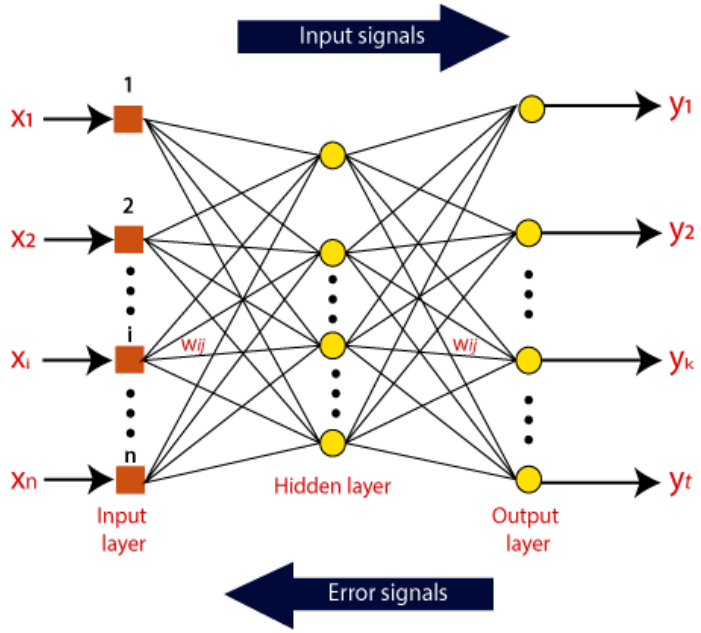


Fig 2: Artificial Neural Network Architecture

Fig 2 depicts the Layers used to organize these nodes, with a layer for input, one or more layers that are hidden, as well as an output layer being usual. Weights—learned throughout the training process—represent the links between the nodes. The basic building block of an artificial neuron is the perceptron, which takes a set of inputs, applies weights to each input, sums them up, and runs an activation function on the total. The activation function adds non-linearity to the model, allowing it to pick up on intricate patterns and connections. An ANN is trained using a two-step technique called back propagation and forward propagation. The input data is supplied into the network during forward propagation, and the activations are computed layer after layer until the result layer is reached. An error signal is then calculated by comparing the output to the desired output. Backpropagation is the procedure used to update the network's weights in order to reduce error. The weights are changed based on the changing slope of the measurement error with respect to those weights as the error signal travels backward through the network. This iterative process continues until the network learns to make accurate predictions or achieves a desired level of performance.

3.2. Convolutional Long Short-Term Memory (CLSTM)

Convolutional Long Short-Term Memory (CLSTM), a variation on the Long Short-Term Memory (LSTM) structure that adds Convolutional processes, goes by the name of CLSTM. Convolutional neural networks (CNNs) can extract spatial features from input, while LSTMs can capture long-term dependencies. These two capabilities are combined to create CLSTM. The Fig 3 Shows the workflow model of proposed CLSTM.

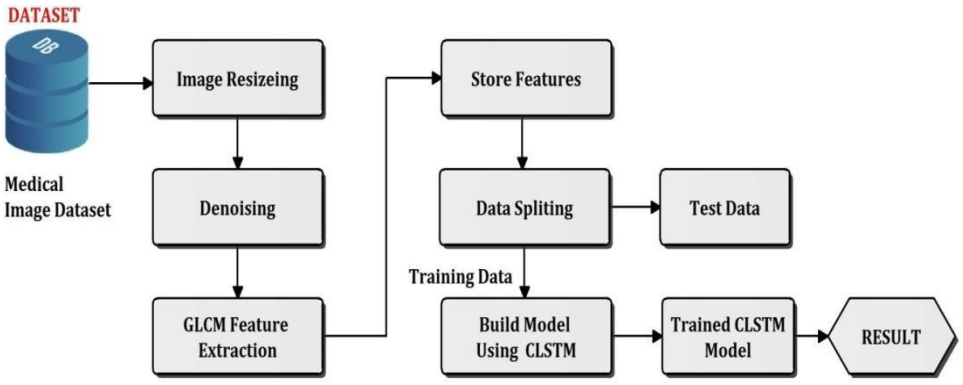
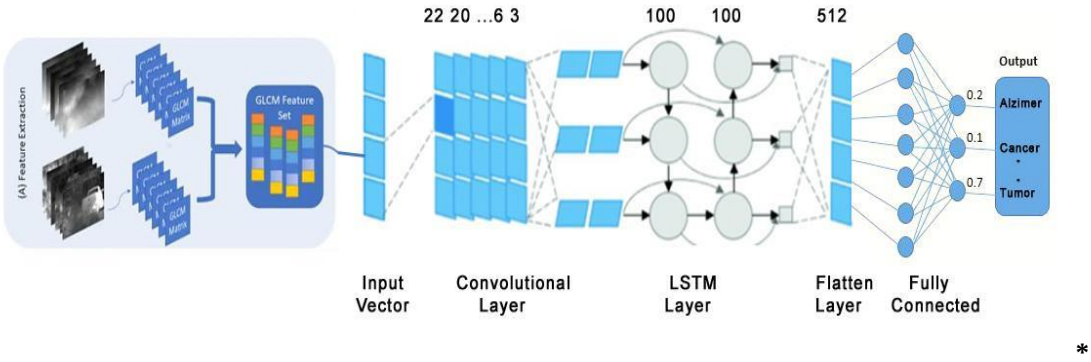


Fig 3: Workflow Model of Proposed CLSTM



*

Fig 4: Convolutional Long Short-Term Memory (CLSTM) Architecture

In a standard LSTM, the input is treated as a sequence, and the recurrent connections allow the network to capture dependencies between elements in the sequence. However, LSTMs do not explicitly consider the spatial structure of the data. CLSTM addresses this limitation by incorporating Convolutional operations into the LSTM framework.The main idea behind CLSTM is to replace the matrix multiplication operations in standard LSTMs with Convolutional operations. This allows the model to capture spatial patterns and dependencies in the input data. By using convolutions, the CLSTM can effectively process grid-like data such as images or spatio-temporal data.

The CLSTM unit consists of three main components as shown in Fig 4:

Convolutional Operations: The input sequence is convolved with filters, similar to a Convolutional layer in a CNN. This step enables the CLSTM to capture spatial features and patterns from the input data.

LSTM Memory Cell: The convolved features are then passed through an LSTM memory cell. The memory cell has a similar structure to a standard LSTMincludes the input gate, output gate, and forget gate. It retains the ability to capture long-term dependencies and process sequential information.

Output: The output of the CLSTM unit is obtained by passing the LSTM's output through a fully connected layer or another appropriate operation, depending on the specific task.

Algorithm 2: CLSTM Classification Process

1. Prepare the data for CLSTM model: *`X_train, X_test, y_train, y_test = split_data(F)`*
2. Define the CLSTM model architecture: *`CLSTM_model = define_CLSTM_model()`*
3. Train the CLSTM model: *`CLSTM_model.train(X_train, y_train)`*
4. Evaluate the model on the test data: *`accuracy = CLSTM_model.evaluate(X_test, y_test)`*
5. Get the predicted class: *`predicted_class = CLSTM_model.predict(F)`*
6. Output the predicted class: *`print("Predicted class:", predicted_class)`*

The Algorithm 2 depicts the different steps in proposed CLSTM classification process. The CLSTM classification

process begins with data preparation, where the extracted GLCM features are split into training and testing datasets, namely X_train, X_test, y_train, and y_test. Subsequently, a Convolutional Long Short-Term Memory (CLSTM) model is defined, tailored for medical image classification. The model is then trained using the training dataset (X_train, y_train) to learn relevant patterns and relationships. Its performance is assessed by evaluating accuracy on the test data (X_test, y_test). Once trained, the CLSTM model can predict the class of new medical images based on their extracted GLCM features, facilitating efficient medical image retrieval and classification.

4. Implementation & Result Analysis

In this implementation and result analysis, two models, Artificial Neural Network (ANN) and CLSTM, were utilized. The objective was to assess their performance and compare the results. The evaluation was based on various metrics such as precision, recall, F1-score, which were obtained from the classification report. The confusion matrix provided a detailed breakdown of the models' predictions and the actual classes. The Receiver Operating Characteristic (ROC) curve was employed to analyze the models' ability to classify instances accurately across different thresholds. By analyzing these metrics and visualizations, we gain valuable insights into the strengths and weaknesses of models.

ANN Architecture Components

We have developed an Artificial Neural Network (ANN) system, where we designed the ANN components as outlined in table 1, to accurately classify five different classes of medical images.

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 24)	600
dropout (Dropout)	(None, 24)	0
dense_1 (Dense)	(None, 24)	600
dropout_1 (Dropout)	(None, 24)	0
dense_2 (Dense)	(None, 24)	600
dense_3 (Dense)	(None, 24)	600
dense_4 (Dense)	(None, 5)	125
Total params: 2,525 Trainable params: 2,525 Non-trainable params: 0		

Table 1: ANN Architecture Components

The ANN consists of several layers, The first layer, named "dense," is a dense layer with 24 neurons. It produces an output shape of (None, 24), which means it generates an array of 24 elements for each input sample. This layer contains 600 trainable parameters representing the weights and biases associated with each neuron.

The second layer, "dropout," is used for regularization to prevent overfitting in the model. It retains the output shape of (None, 24) since dropout does not alter the shape of the data. There are no trainable parameters in the dropout layer as it does not have any weights or biases.

The third layer, "dense_1," is another dense layer with 24 neurons, producing an output shape of (None, 24). It also contains 600 trainable parameters.

The fourth layer, "dropout_1," is another dropout layer, and like before, it retains the output shape of (None, 24) and does not have any trainable parameters.

Next, there are two additional dense layers, "dense_2" and "dense_3," each with 24 neurons and 600 trainable

parameters. Both layers maintain the same output shape of (None, 24).

Finally, the last layer, "dense_4," is a dense layer with 5 neurons, representing the total amount of categories in the task of classifying medical images. It generates a shape of (None, 5) as the output indicating that the model generates a probability distribution over the five classes for each input sample. This layer contains 125 trainable parameters.

The complete ANN model has 2,525 trainable parameters in total. These variables are discovered during the training phase to enhance the accuracy of the model in precisely categorizing medical images. On the other hand, the non-trainable parameters are set to zero, and they represent certain layer configurations or fixed operations that do not change during training

Confusion Matrix- ANN

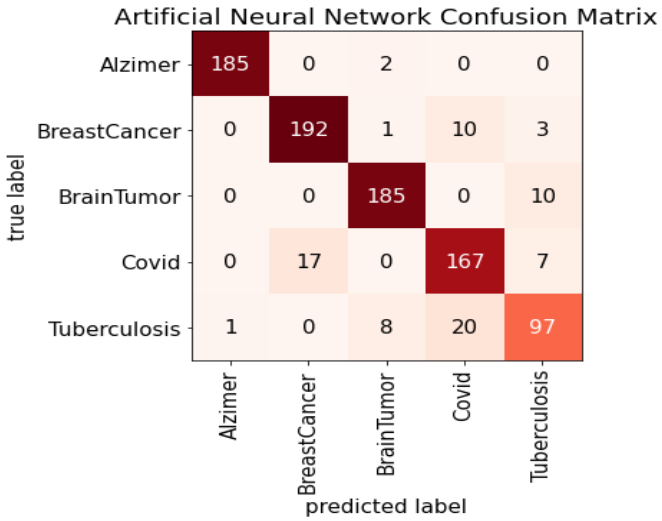


Fig 6: Confusion Matrix for ANN

The chart for the confounded ANN is shown in Fig 6. Understand how well this model operates from the confusion matrix. A total of 903 images are offered. And the accuracy of this model was 91%.

AUC-ROC curve for ANN

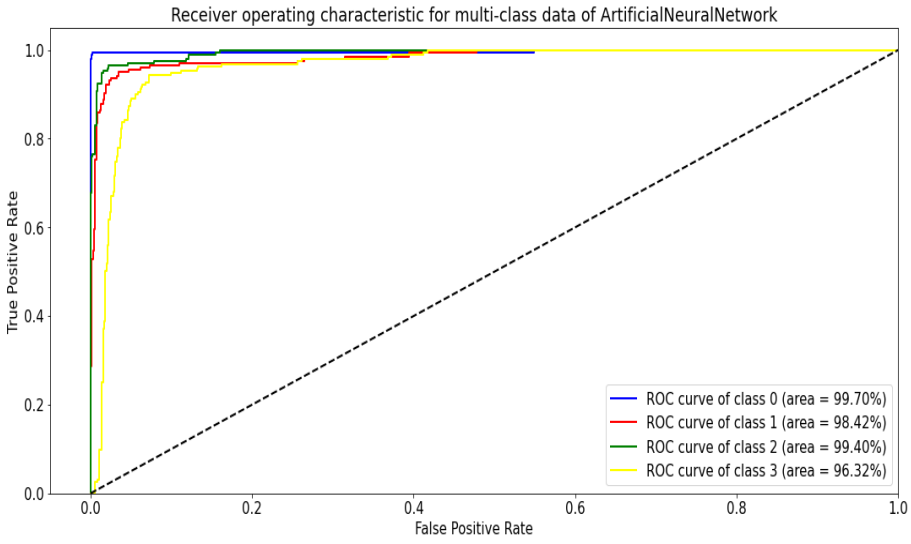


Fig 7: AUC-ROC curve for ANN

A common performance assessment technique for Artificial Neural Network (ANN) models is the AUC-ROC curve shown in Fig 7. AUC-ROC is the acronym for "Area Under the Curve - Receiver Operating Characteristic" curve. At various categorization criteria, the curve compares the True Positive Rate (TPR) and False Positive Rate (FPR). The model's capacity to distinguish among positive & negative examples is quantified by the AUC-ROC, which shows the way the model performs at various thresholds. AUC-ROC values closer to 1 shows great model efficacy in terms of accurately categorizing positive and negative examples,

whereas values closer to 0.5 imply random guessing.

CLSTM Architecture Components

The table 2 showcases the architecture of a CLSTM model created to classify five classes. By integrating convolutional and LSTM layers, the model efficiently processes input data, resulting in precise classification outcomes.

The model begins with two 1D convolutional layers, named "conv1d" and "conv1d_1," each having 128 filters. These layers process the input data, resulting in output shapes of (None, 22, 128) and (None, 20, 128), respectively. The "conv1d" layer has 512 trainable parameters, while the "conv1d_1" layer has 49,280 trainable parameters.

Next, a 1D max-pooling layer, "max_pooling1d," is applied to down-sample the data. It produces an output shape of (None, 10, 128) without introducing any trainable parameters.

Subsequently, two more 1D convolutional layers, "conv1d_2" and "conv1d_3," with 256 filters each, further process the data. The output shapes are (None, 8, 256) and (None, 6, 256), respectively. The "conv1d_2" layer has 98,560 trainable parameters, while the "conv1d_3" layer has 196,864 trainable parameters.

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 22, 128)	512
conv1d_1 (Conv1D)	(None, 20, 128)	49280
max_pooling1d (MaxPooling1D)	(None, 10, 128)	0
conv1d_2 (Conv1D)	(None, 8, 256)	98560
conv1d_3 (Conv1D)	(None, 6, 256)	196864
max_pooling1d_1 (MaxPooling1D)	(None, 3, 256)	0
dropout (Dropout)	(None, 3, 256)	0
1stm (LSTM)	(None, 3, 100)	142800
1stm_1 (LSTM)	(None, 100)	80400
flatten (Flatten)	(None, 100)	0
dense (Dense)	(None, 512)	51712
dense_1 (Dense)	(None, 5)	2565

Table 2: CLSTM Architecture Components

The output shape is (None, 3, 256) after a 1D max-pooling layer, "max_pooling1d_1," conducts additional down-sampling.

The subsequent layer, "dropout," is a regularization layer, and it does not introduce any trainable parameters.

Following the convolutional layers, two LSTM (Long Short-Term Memory) layers, "lstm" and "lstm_1," process the sequential data. The "lstm" layer has 100 units, producing an output shape of (None, 3, 100), and it has 142,800 trainable parameters. The "lstm_1" layer, also with 100 units, produces an output shape of (None, 100) and has 80,400 trainable parameters.

A flatten layer, "flatten," is used to convert the 3-dimensional tensor into a 2-dimensional tensor with shape (None, 100).

The model then incorporates two dense layers, "dense"

and "dense_1," for the final classification. The "dense" layer has 512 neurons and an output shape of (None, 512), with 51,712 trainable parameters. The last dense layer, "dense_1," has 5 neurons, representing the number of classes in the classification task. Its output shape is (None, 5), and it introduces 2,565 trainable parameters.

The neural network has a total of 622,693 trainable parameters, which are modified during the training procedure to enhance the performance of the model. The non-trainable parameters, which indicate particular layer configurations that don't change throughout training, are set to zero.

Confusion Matrix-CLSTM

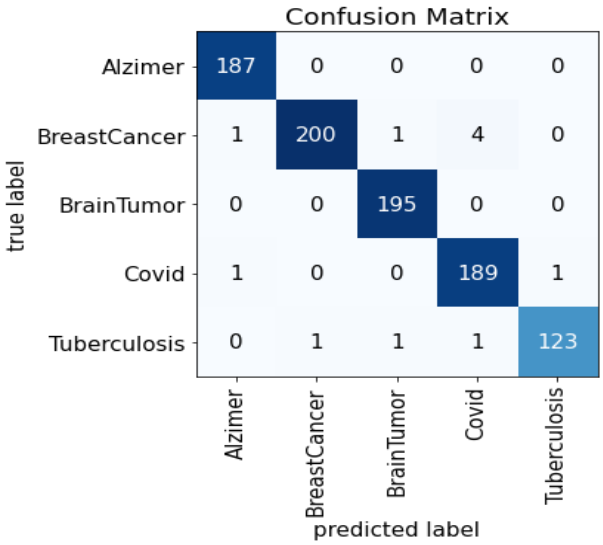


Fig 9: Confusion Matrix for CLSTM

The chart for the confounded CLSTM is shown in Fig 9. Understand how well this model operates from the confusion matrix. A total of 903 images are offered. And the accuracy of this model was 99.01%.

AUC-ROC curve for *CLSTM*

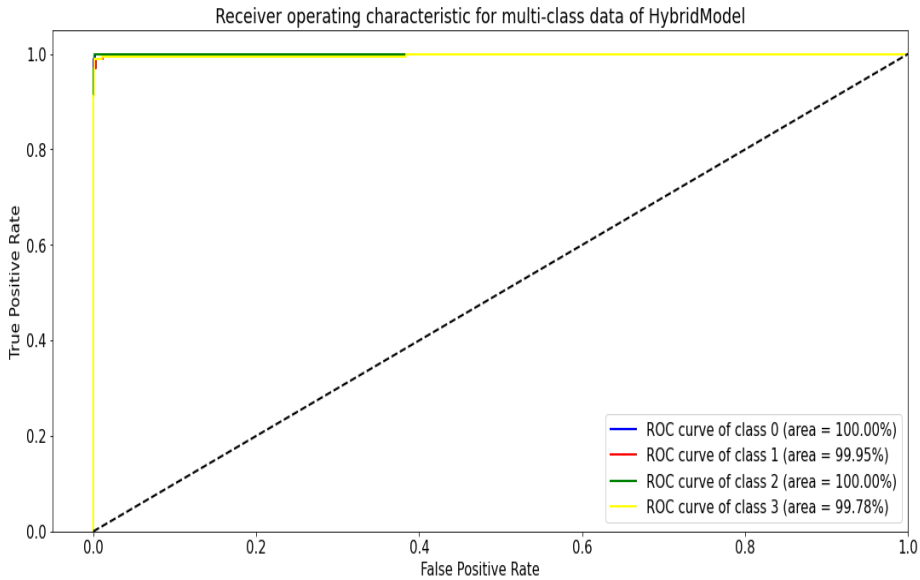


Fig 10: AUC-ROC curve for CLSTM

A performance measure is the AUC-ROC (Area under the Receiver's Operational Characteristic Curve) evaluation metric commonly used in CLSTMmachine learning models shown in Fig 10. CLSTMmodels combine multiple algorithms or techniques to improve predictive accuracy and address complex problems effectively.

In especially for binary classification tasks, the categorization performance of a CLSTM model is evaluated using the AUC-ROC curve. By displaying the True Positive Rate (TPR) versus the False Positive Rate (FPR) at various classification thresholds, it assesses the model's capacity to discern between positive and negative cases. The AUC-ROC score goes from 0 to 1, with 0.5

denoting a random classifier and 1 denoting a perfect classifier. The AUC-ROC score's proximity to 1 indicates how well the CLSTM model can distinguish among the two classes.

The AUC-ROC curve offers a comprehensive view of the CLSTMmodel's performance across various thresholds, making it particularly valuable when dealing with imbalanced datasets or complex classification tasks. It allows researchers and practitioners to make informed decisions about model selection and fine-tuning to achieve the best possible performance in CLSTM machine learning systems.

Table 3: Comparison of CLSTM with ANN

Class	**Precision**	**Recall**	**F1-Score**	**Accuracy**
ANN				
Alzimer	0.99	0.99	0.99	0.91
BreastCancer	0.92	0.93	0.93	
BrainTumor	0.94	0.95	0.95	
Covid	0.85	0.87	0.86	
Tuberculosis	0.83	0.77	0.80	
CLSTM				
Class	**Precision**	**Recall**	**F1-Score**	
Alzimer	0.99	1.00	0.99	0.9901
BreastCancer	1.00	0.97	0.98	
BrainTumor	0.99	1.00	0.99	
Covid	0.97	0.99	0.98	
Tuberculosis	0.99	0.98	0.98	

The Table 3 shows that the ANN (Artificial Neural Network) model achieved an accuracy of 91%, while the CLSTM model achieved an impressive accuracy of 99.01%. Additionally, the CLSTM model consistently outperformed the ANN model in terms of precision, recall, and F1-scores across various classification tasks. The CLSTM model's high accuracy of 99.01% suggests that it is highly effective in disease classification and demonstrates superior performance compared to the ANN model.

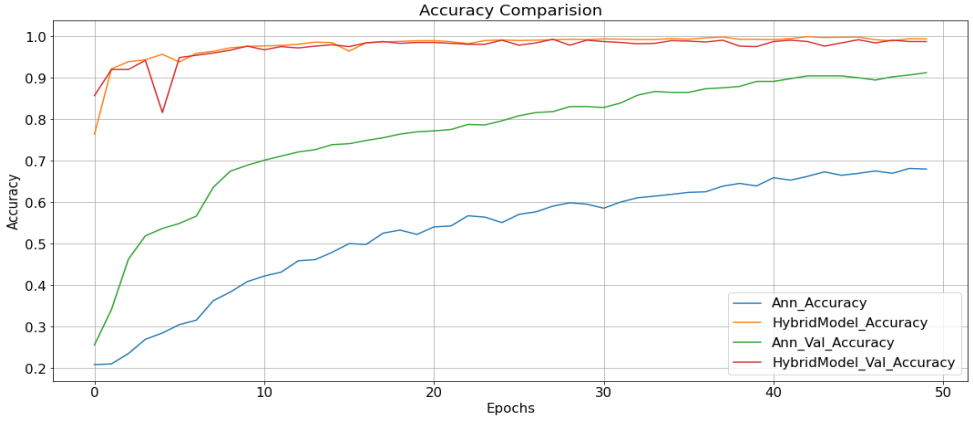


Fig 11: Accuracy Comparison Graph

Accuracy comparison between Artificial Neural Networks (ANN) and CLSTM models is a crucial aspect of evaluating their performance in various machine learning tasks. ANN, a class of deep learning algorithms, excels in learning complex patterns from data and handling large datasets. Its accuracy heavily depends on the network architecture, number of layers, and neurons, along with hyper parameters tuning. In tasks like speech

recognition, natural language processing, & picture recognition, ANN can reach high accuracy. On the other hand, CLSTM models combine the strengths of different algorithms or techniques to enhance overall performance. By integrating various models, they can handle diverse data types and address complex problems effectively. CLSTM models can outperform standalone algorithms in certain situations, especially when dealing with imbalanced datasets or handling multiple types of features.

The Fig 11 shows the accuracy comparison between ANN and CLSTM models varies based on the dataset and problem at hand. In some cases, ANN might yield higher accuracy due to its ability to learn intricate patterns. In contrast, CLSTM models can offer superior accuracy in scenarios where data diversity and feature engineering play critical roles. Ultimately, the choice between ANN and CLSTM models depends on the specific problem, data characteristics, and computational resources available. By systematically testing and validating both approaches, researchers and practitioners can determine which model type is more suitable for a particular task, ultimately achieving the desired level of accuracy.

5. Conclusion

We have provided a brand-new framework for effective content-based retrieval of medical images that is based on deep learning. The framework shows how well the Gray Level Co-occurrence Matrix (GLCM) & particular image attributes may be used to find pertinent medical images across a range of categories. Our experimental results show that the framework is capable, at the ANN (Artificial Neural Network) model categorizing medical images with an accuracy of 91%. The Convolutional Long Short-Term Memory (CLSTM) hybrid deep learning model outperforms the ANN model with an accuracy of 99.01%. These high accuracies highlight the potential of the proposed framework in accurately retrieving relevant medical images.

Future enhancements could include expanding the dataset to a broader range of medical conditions and incorporating more diverse medical image modalities would increase the system's versatility. Furthermore, integrating clinical metadata and patient-specific information into the retrieval process could enable personalized medical image retrieval, tailored to individual patient needs. This would enhance the clinical decision-making process and contribute to more precise diagnoses and treatment planning.

References

[1] Sivakumar, M. & Saravana Kumar, N.M. & N., Karthikeyan, An Efficient Deep Learning-based Content-based Image Retrieval Framework. Computer Systems Science and Engineering. 43. 683-700, 2022.

[2] Alsmadi, Mutasem K. "Content-based image retrieval using color, shape and texture descriptors and features." Arabian Journal for Science and Engineering, 3317-3330, 2020.

[3] Sikandar, Shahbaz, Rabbia Mahum, and AbdulMalik Alsalman. "A Novel Hybrid Approach for a Content-Based Image Retrieval Using Feature Fusion" Applied Sciences 13(7), 4581, 2023.

[4] Shilpa Marathe et al. Fusion of Colour, Texture and Shape features with Supervised Learning Model for Content Based Image Retreival, International Journal of Intelligent Systems and Applications in Engineering (IJISAE), ISSN: 2147-6799, 2023.

[5] Mahum, Rabbia, et al. "A novel framework for potato leaf disease detection using an efficient deep learning model." Human and Ecological Risk Assessment: An International Journal 29(2), 303-326, 2023.

[6] Mahum, Rabbia, et al. "A generic framework for generation of summarized video clips using transfer learning (SumVClip)." 2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC). IEEE, 2021.

[7] Hiremath, P.S. and Pujari, J., 2007, December. Content based image retrieval using color, texture and shape features. In Advanced Computing and Communications, 2007. ADCOM 2007. International Conference on (pp. 780-784). IEEE.

[8] Shereena, V.B. and David, J.M.,Content Based Image Retrieval: A Review. In Computer Science & Information Technology, Computer Science Conference Proceedings (CSCP) (pp. 65-77), 2014.

[9] Piras, L. and Giacinto, G., Information fusion in content based image retrieval: A comprehensive overview. Information Fusion, 37, pp.50-60, 2017.

[10] Saritha, R.R., Paul, V. and Kumar, P.G, Content based image retrieval using deep learning process. Cluster Computing, pp.1-14, 2018.

[11] Avni, U., Greenspan, H., Konen, E., Sharoon, M. and Goldberger, J. X-ray categorization and retrieval on the organ and pathology level, using patch-based visual words. IEEE Trans. Medical Imaging, 30(3), pp.733-746, 2011.

[12] Bay, H., Tuytelaars, T. and Van Gool, L , May. Surf: Speeded up robust features. In European conference on computer vision (pp. 404-417). Springer, Berlin, Heidelberg, 2006.

[13] Tunga, S., Jayadevappa, D. and Gururaj, C.. A

comparative study of content based image retrieval trends and approaches. International Journal of Image Processing (IJIP), 9(3), pp.127-155, 2015.

[14] Singh, A.V.. Content-based image retrieval using deep learning. Rochester Institute of Technology.Anshuman Vikram Singh, 2015.

[15] Kumar, M., Chhabra, P. and Garg, N.K.. An efficient content based image retrieval system using BayesNet and K-NN. Multimedia Tools and Applications, pp.1-14, 2018.

[16] Liu, P., Guo, J.M., Wu, C.Y. and Cai, D., Fusion of deep learning and compressed domain features for content-based image retrieval. IEEE Transactions on Image Processing, 26(12), pp.5706- 5717, 2017.

[17] Saritha, R.R., Paul, V. and Kumar, P.G.,. Content based image retrieval using deep learning process. Cluster Computing, pp.1-14, 2018.

[18] Wan, J., Wang, D., Hoi, S.C.H., Wu, P., Zhu, J., Zhang, Y. and Li, J., , November. Deep learning for content-based image retrieval: A comprehensive study. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 157-166). ACM, 2014.

[19] Wang, H., Cai, Y., Zhang, Y., Pan, H., Lv, W. and Han, H., , November. Deep learning for imageretrieval: What works and what doesn't. In Data Mining Workshop (ICDMW), 2015 IEEE International Conference on (pp. 1576-1583). IEEE, 2015.

[20] Liu, H., Wang, R., Shan, S. and Chen, X., Deep supervised hashing for fast image retrieval. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2064- 2072). 2016.

[21] Vasudeva.R , Dr Chandrashekara S.N, A Comprehensive Study on Image Retrieval Algorithms of Cloud Storage for Information Extraction in Health Care System. International Journal of Computing and Digital Systems, 12(1), 1315-1328, 2022.