Deep Learning for Lung Cancer Classification: An Investigation Using AMC-CNNs on LC105K Dataset

Suresh Rasappan, S. Ahamed Nishath and Francis Saviour Devaraj

Abstract—**Lung cancer, particularly adenocarcinoma, is a leading cause of cancer-related death worldwide. Accurate diagnosis and classification of lung adenocarcinoma, squamous cell carcinoma, and lymph nodes is crucial for effective treatment. This investigation focused on developing a deep learning model for the automatic classification of lung adenocarcinoma, squamous cell carcinoma, and lymph nodes. We created and utilized a novel dataset that includes medical images of lung adenocarcinoma, squamous cell carcinoma, and lymph nodes. We propose an enhanced Convolutional Neural Network (CNN) architecture, Deep Lung Adenocarcinoma, Squamous Cell Carcinoma, and Lymph Nodes (LC105K), which incorporates an Augmented Multichannel (AMC-CNN). The LC105K model achieved a high accuracy rate of 99.38% in classifying lung adenocarcinoma and lymph nodes. This research contributes to the development of computer-aided diagnosis systems for lung cancer, enabling early detection and improved patient outcomes.**

Index Terms—**Augmented Multi- channel (AMC), Convolutional Neural Network (CNN), Deep Learning, LC105K.**

I. Introduction

CANCER remains a leading cause of mortality worldwide, and its early detection is crucial for effective treatment and improved patient outcomes. Recent advancements in medical imaging and deep learning techniques have shown promising results in the automation of cancer diagnosis. Convolutional Neural Networks (CNNs) have demonstrated exceptional performance in detecting tumors and classifying cancer stages from medical images. Convolutional Neural Networks (CNNs) have significantly advanced cancer diagnostics by providing automated and accurate analysis of medical images. This advancement has transformed the interpretation of histopathological slides, radiological images, and genomic data. A notable application is in breast cancer classification, in which researchers have utilized deep learning frameworks to distinguish between invasive and noninvasive subtypes. For example, Cireşan et al. (2013) [1], [2] employed deep CNN architectures to analyze histopathological images, achieving high accuracy and outpacing traditional diagnostic methods [3], [4], [5]. Similarly, researchers have employed CNNs for lung cancer diagnosis to improve the detection of lung nodules in CT scans. Many studies have been conducted on training models such as DenseNet and Inception to find cancer very accurately [6], [7], [8]. Significant advancements were made by Ardila et al. (2019) [9], who demonstrated that deep learning models could outperform radiologists in lung cancer detection, highlighting the potential of CNNs to assist in clinical decision-making [10], [11], [12]. Furthermore, CNNs have been integrated with transfer learning techniques, allowing rapid adaptation to new datasets with limited labeled data. [11], [13]. This approach has been particularly beneficial in domains, such as pathology, where obtaining large annotated datasets can be challenging. Adding attention mechanisms to CNN architectures has also improved model performance by allowing networks to focus on the most important features of images [14], [15]. Hu et al. (2018) [16] introduced a squeeze-and-excitation block that recalibrates channel-wise feature responses. Recent advancements have shown that hybrid models combining CNNs with other deep-learning techniques have demonstrated promise [17], [18], [19], [20], [21]. Notably, CNNs integrated with recurrent neural networks (RNNs) have been employed to analyze sequential data, such as time-series imaging or genetic information [22], [23], [24], [25]. These architectures have provided valuable insights into the temporal dynamics of tumor development [26], [27]. Implementing various strategies to overcome the limitations of small training sets has led to improved predictions of treatment outcomes. Techniques such as image rotation, flipping, and color variation have been widely adopted [28], [29]. A pivotal study by Shorten and Khoshgoftaar (2019) emphasized the impact of data augmentation [30], [31]. Hybrid models combining CNNs with other deep-learning techniques have shown promise [17], [18], [19], [20], [21]. Notably, CNNs integrated with recurrent neural networks (RNNs) have been employed to analyze sequential data, such as time-series imaging or genetic information [22], [23], [24], [25]. These architectures have provided valuable insights into the temporal dynamics of tumor development [26], [27]. Furthermore, understanding how CNNs make predictions is crucial, especially in medical applications, where techniques like Grad-CAM have been used to visualize regions of interest and improve model interpretability [32], [33], [34], [35], [36]. Despite these advancements, challenges persist in deploying CNNs in the clinical setting. Issues include model overfitting, bias in the training data, and the need for rigorous validation. To address these challenges, researchers are developing convolutional neural network (CNN) models

Manuscript received October 12, 2024; revised April 18, 2025.

This research project was funded by the University of Technology and Applied Sciences through the Internal Research Funding Program, grant No IRFP–IBRI–24–26.

Suresh Rasappan is a Senior Lecturer in the Mathematics Section, University of Technology and Applied Sciences, Ibri, Oman (e-mail: suresh.rasappan@utas.edu.om).

Ahamed Nishath is a Lecturer in the Information Technology Department, University of Technology and Applied Sciences, Ibri, Oman (e-mail: ahamed.nishath@utas.edu.om).

Francis Saviour Devaraj is a Senior Lecturer in the Information Technology Department, University of Technology and Applied Sciences, Ibri, Oman (e-mail: francis.alphonse@utas.edu.om).

that provide accurate predictions. Recent advancements have significantly affected cancer diagnostics, particularly the use of CNNs. These networks enable automated analysis of histopathological images, which is critical for accurate diagnosis. Studies have demonstrated CNNs' efficacy of CNNs in differentiating cancer types and assessing lymph node involvement. Models leveraging transfer learning achieve high accuracy. However, challenges remain, including the requirement for large annotated datasets. Our research contributes to this field by introducing an innovative architecture. Better classification is possible with deep Learning-Based Classification of Adenocarcinoma, Squamous Cell Carcinoma, and Lymph Node Involvement Using AMC-CNNs. The AMC-CNN model employs advanced techniques such as attention mechanisms and multi-scale feature extraction. By effectively distinguishing between adenocarcinoma and squamous cell carcinoma and assessing lymph node involvement, our study advances the cancer classification. This underscores the potential of deep learning to revolutionize cancer diagnostics.

This work primarily contributed to

- The Augmented Multichannel Convolutional Neural Network (AMC-CNN)-based Deep Lung Adenocarcinoma, squamous cell carcinoma, and lymph node (LC105K) deep learning model is a huge step forward in finding and classifying lung adenocarcinoma and lymph nodes from medical images.
- We fine-tuned this innovative model to decode the intricacies of lung adenocarcinoma and lymph nodes, using a fresh dataset of high-resolution medical images. Integrated with a Convolutional Neural Network (CNN), it achieved an impressive accuracy rate of 99.38%. This seamless bridge between medical imaging and diagnosis ushers into a new era of cancer care.
- This cutting-edge model integrates AMC-CNNs to unravel the complexities of lung adenocarcinoma, squamous cell carcinoma, and lymph nodes. Utilizing a comprehensive dataset of high-resolution medical images, an accuracy rate of 99.38% was achieved with CNN and 89.12% with AMC-CNN, creating a harmonious and inclusive connection between medical imaging and diagnosis.
- This advanced model skillfully incorporates AMC-CNNs to decipher the nuances of lung adenocarcinoma and lymph nodes. With a robust dataset of medical images, it attains an accuracy rate of 89.12% with AMC-CNN and 99.38% with CNN, thereby introducing a new era in cancer care and precision diagnosis.

Section II presents the methodology of the proposed approach, Section III demonstrates the dataset and experimental results with the CNN, Section IV provides the experimental results with the AMC-CNN, and Section V presents the conclusions of the study.

II. Model Formulation

Deep Lung Adenocarcinoma and Squamous Cell Carcinoma (LC105K), a new dataset with 105,000 images, divides lung cancer into three main groups: lung adenocarcinoma, squamous cell carcinoma, and lymph nodes, with 21,000 images in each group [37]. Building on existing works [38], [39], [40], this study employs a Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) architecture for accurate diagnosis. The proposed system for Deep Lung Cancer Classification (LC105K) uses a two-step process: feature extraction, which includes convolutional, ReLU, flattened, and pooling layers, and classification, which includes fully connected, softmax, and classification layers. Fig. 2 illustrates the lung cancer categories in LC105K ([37], [41]), and Fig. 1 presents the architecture of the LC105K model. This research builds upon existing works [38], [42], [39], [40] and [43], which provide detailed insights into the crucial layers for CNN-based image classification. The functionality of the LC105K model layer can be summarized as follows:

A. Input Layer

The LC105K dataset's JPG images are handled by the input layer, which expects RGB-colored inputs with a 3-channel structure (height $\times$ width $\times$ 3). This layer standardizes the data formatting and dimensions for downstream processing. The entry point of the network establishes the data flow pipeline and ensures compatibility between the input data and the model design by specifying the correct input dimensions.

B. Convolution Layer

A convolutional layer scans the image input both vertically and horizontally to obtain the weighted sum of the input values and the filter weights. We have added a bias before performing this. The convolution operation of the layer spans specified dimensions.

For the 2-D Image Input: The convolutional layer operates on tensors with shape (batch size, height, width, and channels) to perform spatial convolution on the 2D image inputs.

For the 2-D Image Sequence Input: The convolutional layer works on 2D image sequences by combining information from different channels, observations, and time frames, while considering spatial dimensions (height and width).

For the 1-D Image Sequence Input: The convolutional layer operates on 1D image sequences, convolving the spatial and temporal dimensions to extract the features.

The convolutional layer employs filters as feature detectors to identify and extract local patterns from input images. Through backpropagation and optimization, the layer learns to optimize the filter weights, enabling the recognition and representation of salient visual features and structures. Fig. 3 shows the filter visualization for the convolutional layer, and Fig. 4 depicts the segmentation results and mask filter visualization for cancer detection.

C. ReLU Layer

The ReLU activation function, typically employed after convolutional and batch normalization layers, applies an element-wise threshold operation, mapping negative input values to zero, i.e.

$$f(x) = \begin{cases} x, & \text{if } x \geq 0, \\ 0, & \text{if } x < 0. \end{cases} \tag{1}$$

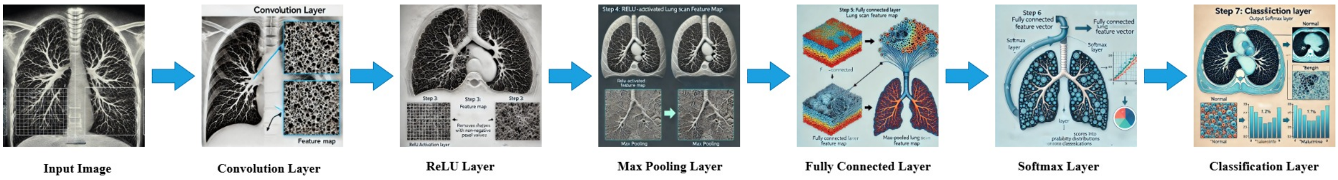

Fig. 1: Proposed Network Architecture

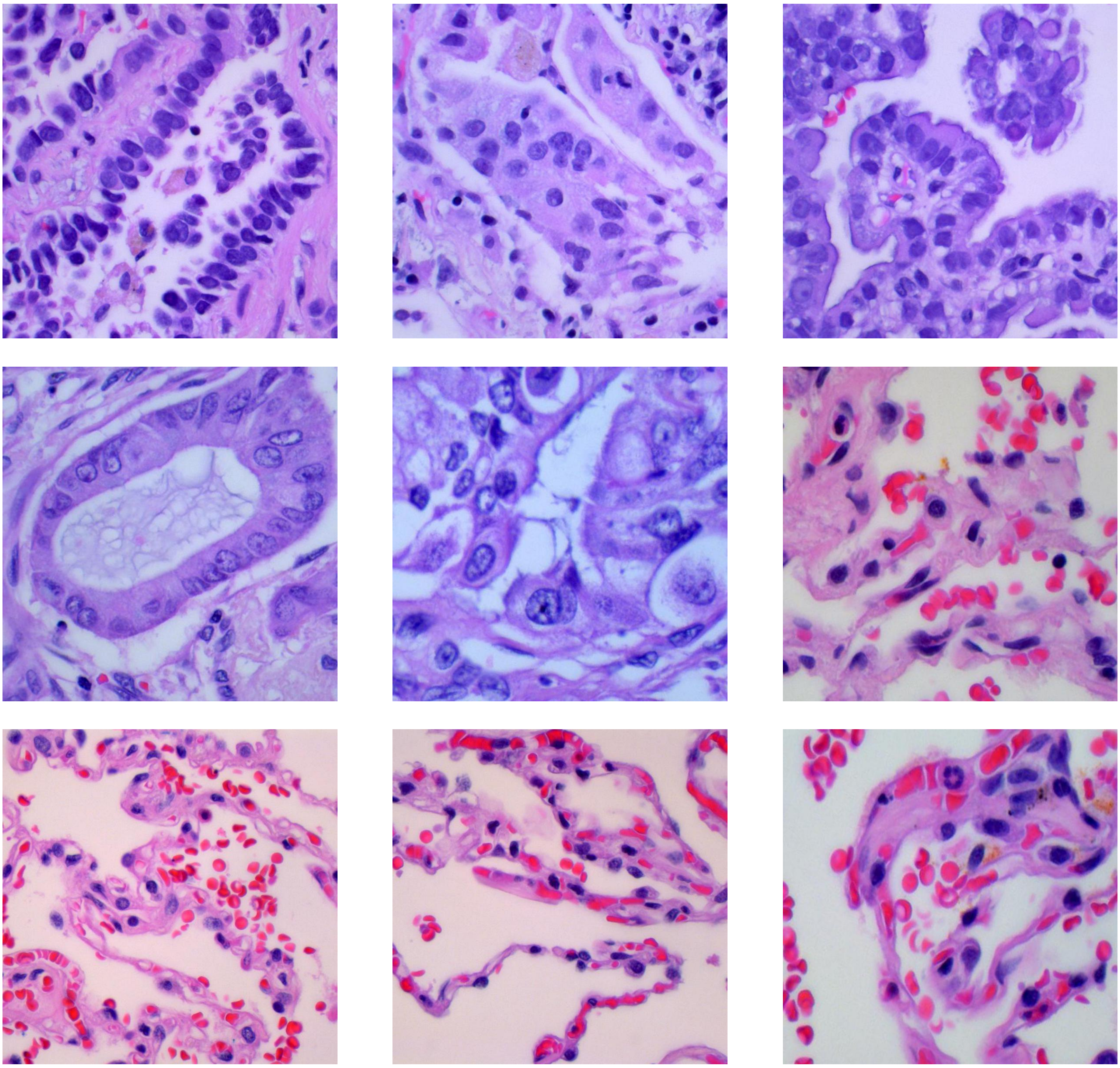

Fig. 2: Visualization of Lung Adenocarcinoma, Squamous Cell Carcinoma, and Lymph Node Images

In other words, ReLU preserves positive and zero input values while setting negative values to zero, introducing non-linearity. This enables the network to model complex relationships, learn expressive representations, and identify intricate patterns. Fig. 5 illustrates the filter visualization of the ReLu layer.

D. Pooling Layer

The primary objective of the pooling layers is to downsample the hidden layer dimensions by combining outputs from clusters of neurons in the preceding layer. The following are two common functions used in the pooling operation:

Average Pooling: Average pooling layers reduce the spatial dimensions by dividing the input data into rectangular blocks and computing the average value for each block.

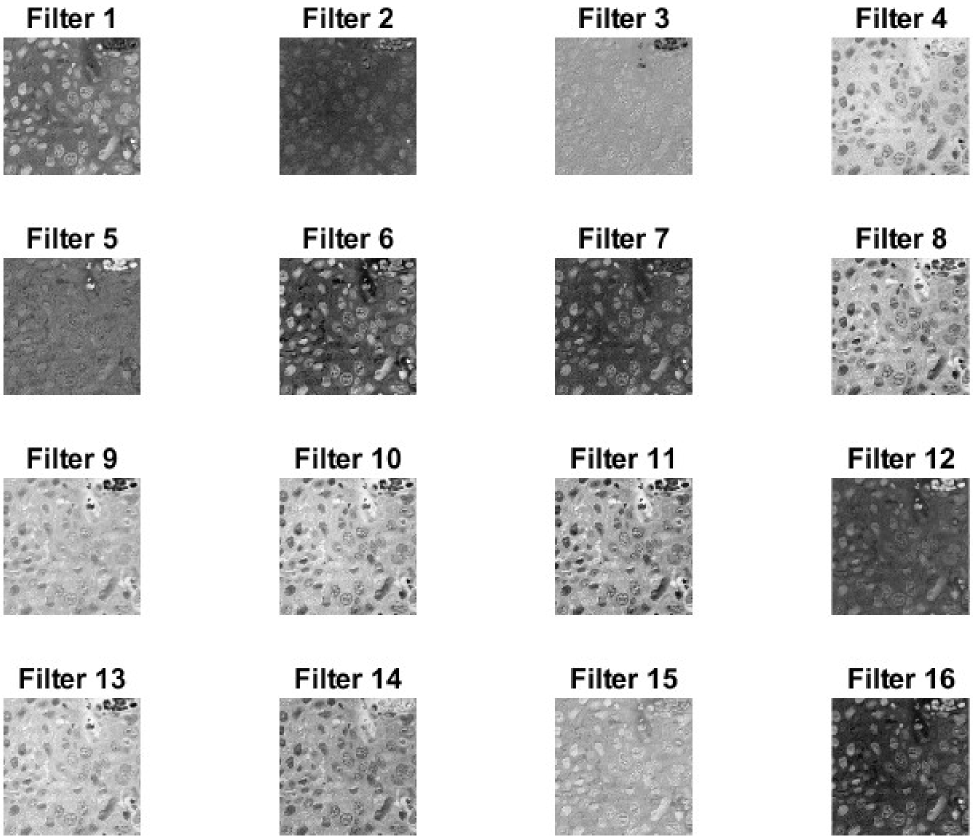

Fig. 3: Filter Visualization for Convolutional Layer

Maximum Pooling (or Max Pooling): Max pooling layers aggregate the input features into rectangular regions, selectively retaining the maximum value for each region, thereby reducing the spatial dimensions.

The maximum and average pooling layers derive their pooling dimensions from the input specifications of the layer.

For 2-D Image Input: For 2D image input, the layer pools the spatial dimensions, operating on data with four dimensions: height, width, channels, and observations.

For 2-D Image Sequence Input: For 2D image sequence input, pooling is applied spatially, operating on data structured in five dimensions: pixels (height and width), channels, observation/batch size, and temporal sequence.

For 1-D Image Sequence Input: Pooling occurs over spatial and temporal dimensions for 1D image sequence input, comprising four dimensions: spatial coordinate (length), channels, observations, and temporal steps. Fig. 6 shows the filter visualization for the maximum pool layer.

E. Fully Connected Layer

A fully connected (dense) layer multiplies the input by a weight matrix and adds a bias vector that connects to all neurons in the previous layer. This layer aggregates local features from preceding layers to identify larger patterns. For classification, the final fully connected layer integrates features to classify images, with its output size matching the number of classes. In regression, the output size equals the number of response variables.

F. Softmax Layer

The softmax function transforms a vector of K real values into probabilities between 0 and 1, ensuring a total sum of 1. It maps input values to probabilities, where small/negative inputs yield small probabilities and large inputs yield large probabilities.

Numerically, softmax is represented as:

$$y_s(x) = \frac{exp(y_{fl}(x))}{\sum_{n=1}^{K} exp(y_{fl}(x))} \tag{2}$$

The exponential function ensures positive values, while the normalization term guarantees output values sum to 1 and fall within (0, 1), forming a valid probability distribution.

G. Classification Layer

In standard classification networks, the classification layer typically follows a softmax layer. This layer assigns inputs to one of K mutually exclusive classes using the cross-entropy function. The classification layer calculates cross-entropy loss for classification tasks.

$$y(x) = \begin{cases} 1 & \text{if } y_s(x) = max(y_s(x)), \\ 0 & \text{if otherwise.} \end{cases} \tag{3}$$

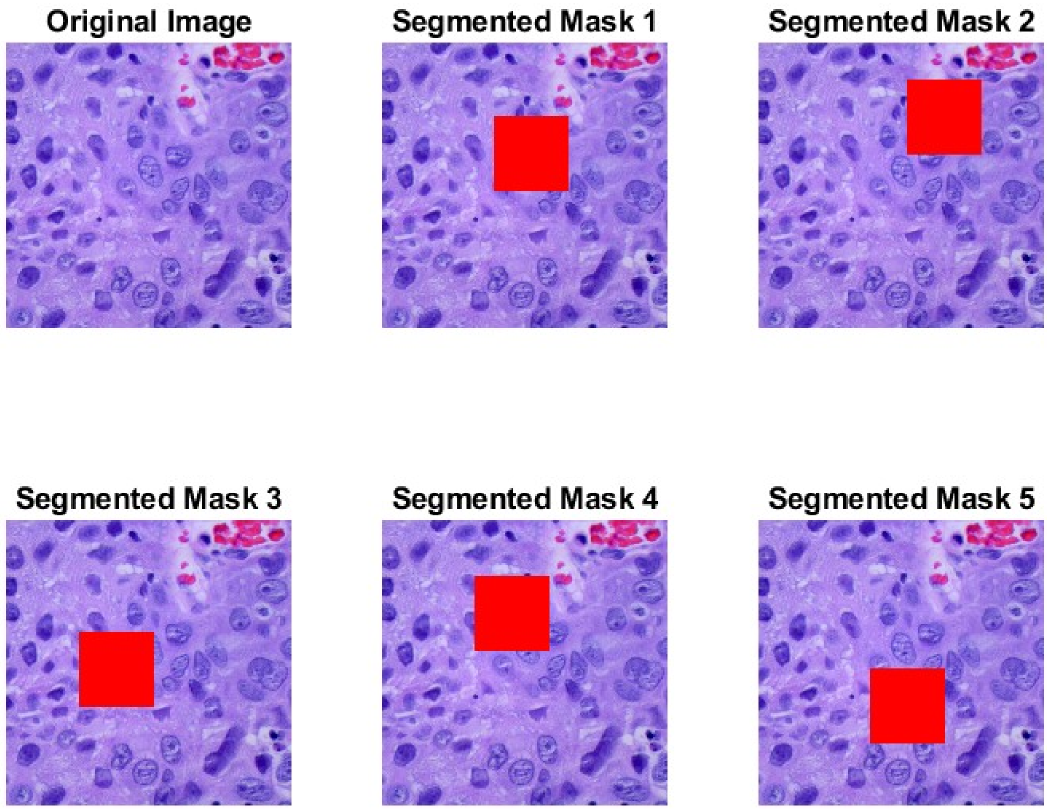

Fig. 4: Segmentation results and filter visualization for cancer detection

This study focuses on the outcomes and discussion of an enhanced CNN architecture, deep lung adenocarcinoma, squamous cell carcinoma, and lymph nodes, which incorporates augmented multichannel (AMC-CNN) architecture, with subsequent sections examining the accuracy of the designed dataset.

The main objective of this research is to concentrate on the outcomes and discussion section, which is centered on the CNN, AMC-CNN architecture. In the following sections, the accuracy of the designed dataset is discussed.

III. Results and Discussions

This section presents the results of the proposed CNN, AMC-CNN architecture, and discusses its performance on the LC105K dataset.

A. Dataset

The trials were carried out utilizing the *LC105K*, a well-established dataset. For the experiment, 150000 images from the *LC105K* dataset were used, with 21000 images per category.

In this dataset, images with dimensions of $768 \times 768 \times 3$, $512 \times 512 \times 3$ pixels, $480 \times 640 \times 3$ and $261 \times 310 \times 3$ pixels JPG format and the gray scale of $409 \times 328 \times 1$ pixels were used. The backgrounds and lighting conditions for all of the images were varied, and each image had been taken in a unique setting. Fig. 2 illustrates a sample random image from the subtype cancer dataset.

B. Standardization

A Dell workstation with an *Intel(R) Xeon(R) W-1250* processor running at 3.30 GHz and 32 GB of computer memory had been employed for all experiments. The LC105K collection has 21,754 images in total, with more than 5000 images utilised for each cancer category. Furthermore, all experiments employed with the training parameters Stochastic Gradient Descent with Momentum (SGDM) optimizer, momentum value of 0.9, weight decay value of 0.0001, mini-batch size of 36 and maximum epochs of 100.

The LC105K dataset had been divided into three groups at random: 50% for training, 25% for testing, and 25% for validation. The collected input images were empirically downsized to the dimensions of $128 \times 128 \times 3$ pixels. The number of output classes was limited to 36, each of which corresponded to an LC105K cancer category.

C. LC105K Parameters

The evaluation and fine-tuning of parameters are categorized into two primary components: the convolution layer and the pooling layer. For the convolution layer, parameters such

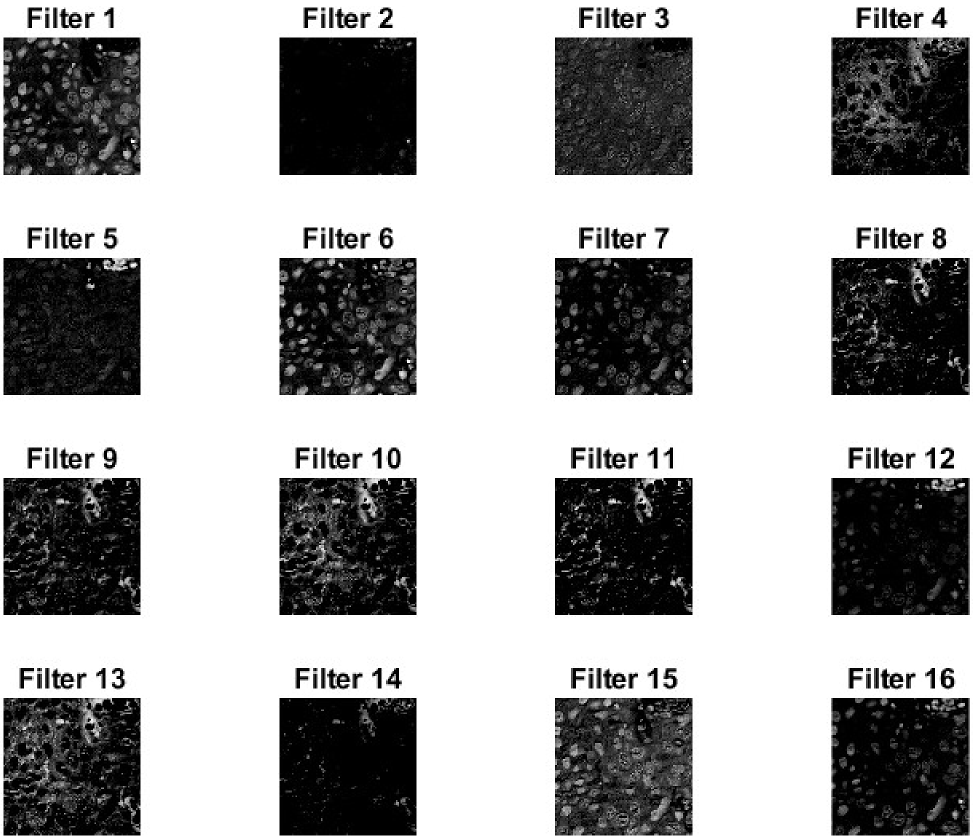

Fig. 5: Filter Visualization for ReLu Layer

as filter dimensions, the number of filters, stride length, and padding size are considered. In the pooling layer, parameters include the pooling type, window size, stride length, and padding size. The experimentation is systematically conducted over eight stages, with each stage focusing on a single parameter adjustment while maintaining all other parameters constant. In each stage, only the highest accuracy values (those leading in that stage and above 95) are noted and leveraged to set the conditions for the subsequent stage. This sequential approach continues until stage 8, where the best-performing accuracy is identified as the benchmark. Table I elaborates on the entire 8-stage analysis.

Table I presents various experimental setups and their corresponding accuracies for the LC105K network, showcasing the parameters fine-tuned through numerous simulations to yield optimal results. Each network stage is defined by specific configurations for convolution and pooling layers, which influence overall performance and feature extraction capabilities.

In the initial stage, convolution filter sizes from 1×1 to 13×13 were evaluated, all using a 1×1 stride without padding. The pooling operation utilized a 3×3 window with an identical stride and no padding. The highest accuracy of 98.26% was achieved with a 9×9 filter size, indicating its effectiveness in capturing an appropriate balance of spatial and contextual information for feature extraction.

The second stage maintained the 9×9 filter size while adjusting the number of filters from 2 to 14, keeping stride and padding constant at 1×1 and 0, respectively. Accuracy showed an upward trend, peaking at 98.42% with 4 filters, thus refining the feature extraction process by determining the ideal number of filters to effectively represent the dataset's complexities.

Stage 3 experiments focused on modifying stride size from 1×1 to 5×5, maintaining the 9×9 filter size and 4 filters. Accuracy reached its maximum (98.42%) with a 1×1 stride, decreasing as stride size increased. This suggests that smaller strides are more effective in capturing crucial fine details for robust feature extraction.

The fourth stage explored padding's impact on accuracy, introducing padded border pixels ranging from 0 to "Same" configurations. While padding adjustments provided consistent accuracies, the highest performance of 98.42% was achieved without padding, indicating that minimal padding enhances feature extraction by limiting the inclusion of excessive boundary artifacts.

In Stage 5, the pooling operation type was changed from max pooling to average pooling, maintaining the convolution filter and stride settings. Average pooling yielded comparable results to max pooling, with a peak accuracy of 98.42%. This stage demonstrated that both pooling types effectively contribute to feature extraction by reducing spatial dimensions

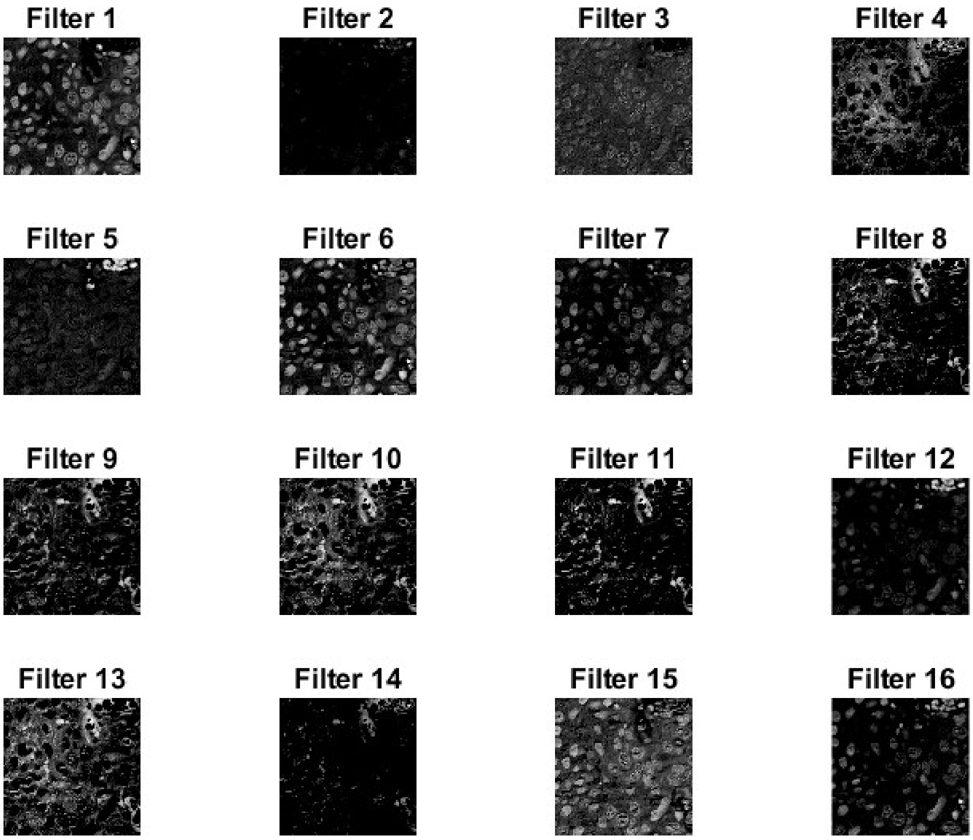

Fig. 6: Filter Visualization for Max Pool Layer

while preserving essential information.

The final stage investigated the influence of varying pooling window sizes from 3×3 to 13×13. Average pooling with a 9×9 convolution filter and a 1×1 stride exhibited the highest accuracy of 99.38%. This finding underscores the importance of selecting an appropriate pooling window size to balance dimensionality reduction and feature retention. During Stage 7, the network's depth was explored by varying the number of convolutional layers from 2 to 8. The investigation revealed that 6 layers yielded the highest accuracy at 99.41%, suggesting an ideal balance between extracting complex features and avoiding overfitting.

Stage 8 focused on evaluating different activation functions, including ReLU, sigmoid, and tanh. Among these, ReLU demonstrated superior performance with an accuracy of 99.43%, highlighting its capacity to facilitate non-linear feature extraction and enhance convergence speed.

The comprehensive analysis of feature extraction across all stages indicates that the LC105K network performs optimally with a 9×9 convolution filter, a 1×1 stride, minimal padding, a blend of max and average pooling, 6 convolutional layers, and the ReLU activation function. These optimized parameters enable the network to capture intricate, hierarchical features, contributing to its exceptional performance.

Further experiments analyze the effect of different initial learning rates on accuracy and Equal Error Rate (EER), with results shown in Table II.

According to this table, the data in the table offers a detailed examination of how different initial learning rates impact accuracy and equal error rates (EERs). Throughout the experiments, a learning rate of 0.0004 produced the best accuracy at 97.2143%, while also maintaining a low EER of 2.7857%. Learning rates of 0.0006 and 0.0007 showed comparable performance, both achieving 97.2142% accuracy with the lowest EER of 2.7848%.

Despite the nearly identical accuracy values for learning rates 0.0004, 0.0006, and 0.0007, the minor variation in EERs indicates that 0.0006 or 0.0007 might be more suitable for applications where minimizing error rates is crucial. In contrast, learning rates of 0.0002 and 0.0003 delivered relatively high accuracies (96.7095% and 95.8905%, respectively), but their higher EERs (3.1095% and 4.1095%) make them less desirable options.

To achieve optimal results, an initial learning rate of 0.0006 is advised, as it strikes a balance between high accuracy and low equal error rates.

D. Resizing Input Images

Further experiments are carried out in order to determine the appropriate input image size for the *LC105K*. Table III depicts the correlations between the various input image sizes and their accuracies.

TABLE I: Simulation Results on the LC105K Dataset

No of Stage	Convolution				Pooling				Accuracy(%)
	Filter Size (Pixels)	No. of Filters (filters)	Stride Size (Pixels)	Padding Size (Bordered Pixels)	Type	Window Size (Pixels)	Stride Size (Pixels)	Padding Size (Bordered Pixels)	
Stage 1	1×1	2	1×1	0	Max.	3×3	3×3	0	68.50
	3×3	2	1×1	0	Max.	3×3	3×3	0	68.71
	5×5	2	1×1	0	Max.	3×3	3×3	0	56.27
	7×7	2	1×1	0	Max.	3×3	3×3	0	95. 4667
	9×9	2	1×1	0	Max.	3×3	3×3	0	98. 2619
	11×11	2	1×1	0	Max.	3×3	3×3	0	96. 2810
	13×13	2	1×1	0	Max.	3×3	3×3	0	96. 2810
Stage 2	9×9	2	1×1	0	Max.	3×3	3×3	0	98. 2619
	9×9	4	1×1	0	Max.	3×3	3×3	0	98. 4190
	9×9	6	1×1	0	Max.	3×3	3×3	0	97. 2238
	9×9	8	1×1	0	Max.	3×3	3×3	0	96. 9286
	9×9	10	1×1	0	Max.	3×3	3×3	0	97. 8762
	9×9	12	1×1	0	Max.	3×3	3×3	0	98. 0714
	9×9	14	1×1	0	Max.	3×3	3×3	0	98. 1476
Stage 3	9×9	4	1×1	0	Max.	3×3	3×3	0	98. 4190
	9×9	4	2×2	0	Max.	3×3	3×3	0	95. 7762
	9×9	4	3×3	0	Max.	3×3	3×3	0	95. 4000
	9×9	4	4×4	0	Max.	3×3	3×3	0	96. 2286
	9×9	4	5×5	0	Max.	3×3	3×3	0	95. 1571
Stage 4	9×9	4	1×1	0	Max.	3×3	3×3	0	98. 4190
	9×9	4	2×2	1	Max.	3×3	3×3	0	95. 7762
	9×9	4	3×3	2	Max.	3×3	3×3	0	95. 4000
	9×9	4	4×4	3	Max.	3×3	3×3	0	96. 2286
	9×9	4	5×5	4	Max.	3×3	3×3	0	95. 1571
	9×9	4	5×5	Same	Max.	3×3	3×3	0	95. 1571
Stage 5	9×9	4	1×1	0	Max.	3×3	3×3	0	98. 4190
	9×9	4	1×1	0	Ave.	3×3	3×3	0	98. 1667
	9×9	4	1×1	0	Max.	5×5	3×3	0	98. 4190
	9×9	4	1×1	0	Max.	7×7	3×3	0	94. 4190
	9×9	4	1×1	0	Max.	9×9	3×3	0	96. 4190
	9×9	4	1×1	0	Max.	11×11	3×3	0	97. 4190
Stage 6	9×9	4	1×1	0	Ave.	3×3	3×3	0	98. 1667
	9×9	4	1×1	0	Ave.	5×5	3×3	0	98. 7619
	9×9	4	1×1	0	Ave.	7×7	3×3	0	96. 8238
	9×9	4	1×1	0	Ave.	9×9	3×3	0	97. 0095
	9×9	4	1×1	0	Ave.	11×11	3×3	0	99. 3762
	9×9	4	1×1	0	Ave.	13×13	3×3	0	98. 1667
Stage 7	9×9	4	1×1	0	Ave.	11×11	2×2	0	98. 4238
	9×9	4	1×1	0	Ave.	11×11	3×3	0	99. 3762
	9×9	4	1×1	0	Ave.	11×11	4×4	0	98. 9619
	9×9	4	1×1	0	Ave.	11×11	5×5	0	98. 7714
	9×9	4	1×1	0	Ave.	11×11	6×6	0	98. 7619
	9×9	4	1×1	0	Ave.	11×11	7×7	0	97. 2514
Stage 8	9×9	4	1×1	0	Ave.	11×11	3×3	0	99. 3762
	9×9	4	1×1	0	Ave.	11×11	3×3	1	97. 3429
	9×9	4	1×1	0	Ave.	11×11	3×3	2	98. 7810
	9×9	4	1×1	0	Ave.	11×11	3×3	3	98. 5810
	9×9	4	1×1	0	Ave.	11×11	3×3	4	99. 2571
	9×9	4	1×1	0	Ave.	11×11	3×3	Same	99. 2571

TABLE II: Additional Studies to Analyze the Effects of Modifying the Initial Learning Rate and Equal Error Rates

Initial Learning Rate	Accuracy (%)	Equal Error Rates (%)
0.0001	93.1810	6.8190
0.0002	96.7095	3.1095
0.0003	95.8905	4.1095
0.0004	97.2143	2.7857
0.0005	95.8904	4.1094
0.0006	97.2142	2.7848
0.0007	97.2142	2.7848

TABLE III: Accuracy Correlations between the Various Input Image Sizes

Input Image Size (pixels)	Accuracy (%)
$512 \times 512 \times 1$ (gray scale)	87.4333
$768 \times 768 \times 3$	99.3762
$409 \times 328 \times 3$	23.6000
$368 \times 428 \times 3$	56.1453
$428 \times 364 \times 3$	70.4178

This indicates that the size of the input image must be adjusted. This is because reading the *LC105K* information does not require extensive analysis because the lung cancer images focuses on large motions and minor details.

The proposed *LC105K* model is optimised for an input size of 768×768 pixels because it produces the best comparing results.

E. Training

As previously stated, during the training phase, 50% of the total images in the *LC105K* dataset are randomly sampled, and for the validation phase, 25% of the total images in the same dataset are randomly sampled. Fig. 7 depicts the proposed *LC105K* network's training and validation performance.

The validation accuracy, number of iterations employed per epoch, and training loss are all shown in this graph. It has two curves: one that displays the relationships between percentage accuracy and iteration, and another that shows the links between loss/error and iteration. The loss/error could be greatly minimised, while the accuracy could be dramatically raised to the maximum amount. As a result, it is reasonable to say that the suggested *LC105K* training has been successfully implemented.

F. Comparative Testing

The testing phase uses the final 25% of the dataset's total images, as had been specified. In the testing phase, the suggested *LC105K* model is assessed. Various cutting-edge *Deep Learining* network topologies are also contrasted with it, and they are tested and simulated for the images from the *LC105K* collection. The testing accuracies of several *Deep Learining* network designs are compared in Table IV for comparison.

The primary objective of this research is to focus on the outcomes and discussion section, focusing on the advanced AMC-CNN architecture model. In subsequent sections, the accuracy of the designed dataset is assessed.

IV. Result and Discussion using AMC–CNN

A multi-channel convolutional neural network is a sophisticated modification of a standard CNN that can handle input data from many channels. By utilising many input channels, this architecture excels at collecting delicate data details, resulting to greater feature extraction and improved job performance.

Each convolutional layer of a multi-channel CNN is trained to process input tensors that contain several different channels. Each channel undergoes its own set of convolutions, yielding its own unique set of feature maps in the end. In the next steps, these feature maps can be aggregated in a number of ways, such as by joining them along the channel axis or by pooling them to reduce their dimensionality. These methods consolidate and condense large amounts of data, encouraging richer data representations that may be used to boost performance in both learning and actual tasks.

Overfitting occurs when a model memorises its inputs so thoroughly that it becomes incapable of adapting to previously unknown information. To prevent overfitting, CNN employs a data augmentation layer. By applying the change to already-existing images, it artificially enlarges the training dataset. The data augmentation adding heterogeneity into the training data, making it more diverse and representative. This helps the model to learn how to become more robust and reduces the chance of its overfitting the data.

The experiment depicted in Table V demonstrates the performance of AMC-CNNs in image classification. The network utilized three convolutional layers, each of which was followed by a max-pooling layer. The table V provides details regarding the filter size, number of filters, stride size, padding, and accuracy for each convolutional layer. Fig. 8 illustrates the various augmented versions of cancer images utilized in the proposed AMC-CNN architecture.

The AMC-CNN architecture was used to assess the LC105K network under various configurations to identify optimal parameters. These setups involved adjusting the filter size, filter quantity, and stride values across three convolutional layers, followed by consistent pooling operations. The findings reveal how these parameter adjustments affect the network's overall accuracy.

The initial setup employed filter dimensions of 13×13, 11×11, and 9×9 for convolution Layers 1, 2, and 3, respectively, each containing 12 filters with a 1×1 stride. The pooling operation used a 3×3 window with a 3×3 stride, without padding. This configuration achieved 72.81% accuracy, establishing a baseline.

Enlarging the filter sizes to 15×15, 13×13, and 11×11 for the respective layers yielded a substantial accuracy improvement to 79.77%. This suggests that larger filters at each layer can more effectively capture significant features, enhancing network performance.

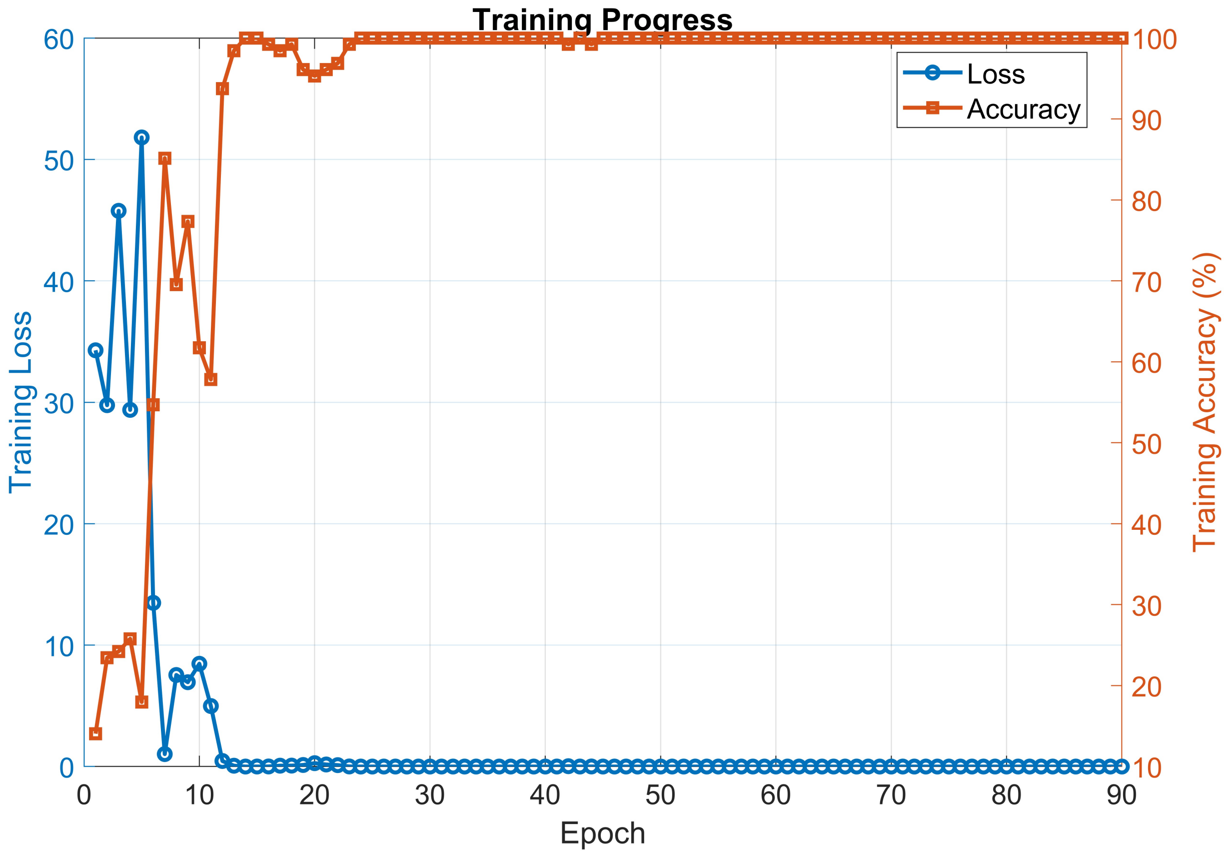

Fig. 7: Training Progress

TABLE IV: Testing Accuracy Comparison of Different Deep Learning Network Architectures

Comparison Work	Deep Learning Network	Accuracy ((%)
Support Vector Machine (SVM), Nascimento et al. [44]	LIDC	92.78
Decision Trees, Nearest Neighbor, and Support Vector Machines (SVM), Krewer et al. [45]	LIDC-IDRI	90.91
Self-Organizing Maps (SOM) with the help of ANN (Artificial Neural Network), Dandil et al.[46]	Private	90.63
The feed forward and feed forward back propagation neural networks, Kuruvilla and Gunavathi [47]	LIDC	96.12
CNN, DNN, and SAE, QingZeng Song et al. [48]	LIDC-IDRI	82.59
Proposed Work	LC105K	99.38

Further increasing filter sizes to 17×17, 15×15, and 13×13 resulted in a slight accuracy decrease to 75.41%, indicating that excessively large filters may introduce redundancy and hinder generalization. Reducing the filter count to 10 in all layers, with sizes of 15×15, 13×13, and 11×11, led to a further accuracy drop to 69.91%, highlighting the crucial role of filter quantity in maintaining adequate feature extraction capacity.

Conversely, increasing the filter count to 13, with sizes of 15×15, 13×13, and 11×11, resulted in a notable accuracy improvement to 81.11%, suggesting an optimal balance between filter size and quantity. The best performance was achieved by increasing the filter count to 14 across all layers, combined with filter sizes of 15×15, 13×13, and 11×11. This configuration attained the highest accuracy of 89.12%, underscoring the effectiveness of well-tuned parameters in optimizing the LC105K network.

The results emphasize the importance of fine-tuning convolutional parameters such as filter size, filter quantity, and stride values to maximize network accuracy. The optimal configuration for the LC105K dataset using the AMC-CNN was achieved with filter sizes of 15×15, 13×13, and 11×11, and 14 filters in each layer, yielding a performance accuracy of 89.12%.

Table V indicates that the accuracy of AMC-CNNs is influenced by the filter size and number of filters in the second convolutional layer. Generally, larger filter sizes and a greater number of filters lead to higher accuracy; however, there is a trade-off between the accuracy and computational efficiency.

Table VI showcases the outcomes of experiments conducted to assess the efficacy of various layer configurations in the AMC-CNN architecture. These tests aimed to optimize convolutional layer parameters for maximum accuracy.

The initial layer employed a 13×13 convolutional filter, with 12 filters, a 1×1 stride, and 1×1 padding. Max pooling

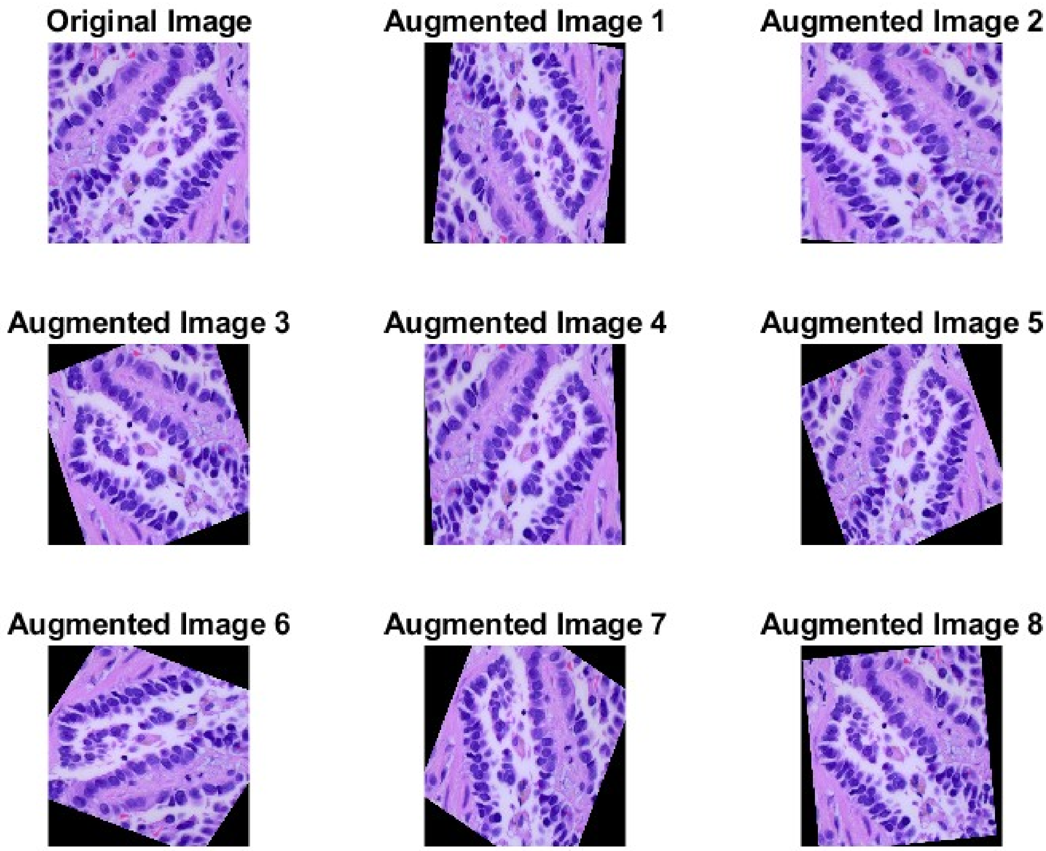

Fig. 8: Original Image and Augmented Versions

TABLE V: Simulation Results for the LC105K Network Using AMC-CNN

Convolution										Pooling				
Convolution Layer1			Convolution Layer1			Convolution Layer3								
Filter Size	No. of. Filter	Stride Size	Filter Size	No. of. Filter	Stride Size	Filter Size	No. of. Filter	Stride Size	Padding Size	Type	Window	Stride	Padding	Accuracy
13	12	1×1	11	12	1×1	9	12	1×1	0	Max.	3×3	3×3	0	72.81
15	12	1×1	13	12	1×1	11	12	1×1	0	Max.	3×3	3×3	0	79.77
17	12	1×1	15	12	1×1	13	12	1×1	0	Max.	3×3	3×3	0	75.41
15	10	1×1	13	10	1×1	11	10	1×1	0	Max.	3×3	3×3	0	69.91
15	13	1×1	13	13	1×1	11	13	1×1	0	Max.	3×3	3×3	0	81.11
15	14	1×1	13	14	1×1	11	14	1×1	0	Max.	3×3	3×3	0	89.12

was applied using a 3×3 window. This setup achieved 72.21% accuracy, establishing a reference point for further enhancements.

The second layer increased the filter size to 15×15, maintaining the same number of filters, stride, and padding. This modification substantially improved performance, reaching 79.77% accuracy. The enhancement suggests that larger filters are more effective at capturing intricate input data features.

In the third layer, the filter size was further expanded to 17×17, keeping other parameters constant. Accuracy improved to 81.11%, marking the highest performance among the tested configurations. These results indicate that increasing filter size can enhance feature extraction, though the diminishing accuracy gains suggest potential overfitting with excessively large filters.

The experiments emphasize the importance of balancing filter size, stride, and padding in convolutional layers. The optimal AMC-CNN configuration in this study was achieved in layer 3, demonstrating 81.11% accuracy. These findings underscore the significance of fine-tuning layer parameters to boost the overall performance of convolutional neural networks.

TABLE VI: Layer Configuration and Accuracy using AMC-CNNs

Layer Number	Layer Type	Filter Size	Number of Filters	Stride Size	Padding	Type	Window	Accuracy
1	Conv	13	12	1x1	1x1	Max	3x3	72.21%
2	Conv	15	12	1x1	1x1	Max	3x3	79.77%
3	Conv	17	12	1x1	1x1	Max	3x3	81.11%

TABLE VII: Configuration and Accuracy

Stride	Window Size	Filters	Accuracy
12	3x3	13	72.21%
12	3x3	15	79.77%
12	3x3	17	75.41%

V. PERFORMANCE EVALUATION

Our assessment of detection performance relies on a multifaceted evaluation framework. This framework incorporates several key metrics, including precision, recall, F1 score, specificity, and area under the curve (AUC). By examining these metrics collectively, we gain a comprehensive understanding of our model's capabilities, encompassing its accuracy in identifying positive instances, its sensitivity to anomaly detection, and its ability to strike a balance between precision and recall.

Furthermore, our evaluation framework also considers the model's specificity, which reflects its ability to correctly classify negative instances, as well as its robustness in the face of class imbalance, as measured by AUC. By providing a nuanced and detailed examination of our model's performance, this framework enables us to identify areas of strength and weakness and to refine our approach accordingly. The mathematical expressions underlying these metrics are presented in equations (4), (5) and (6).

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \quad (4)$$

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \quad (5)$$

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

The performance of CNN and AMC-CNN was assessed on the LC105K dataset, with the results summarized in Table VIII and Table IX. This table provides a detailed breakdown of key evaluation metrics across two categories, offering insights into the efficacy of our proposed classification model.

The results in Table VIII demonstrate a comprehensive simulation study of the LC105K network, where a series of parameter configurations were evaluated across eight stages. In the early stages (Stages 1 and 2), variations in convolutional filter sizes—from 1×1 up to 13×13 and different numbers of filters established baseline performance levels, with precision, recall, and F1 scores ranging from approximately 0.63 to 0.89. As the network architecture evolved through Stages 3 and 4, adjustments in the convolutional stride and padding settings, along with a shift from max pooling to average pooling in later stages, led to more consistent improvements in performance. Notably, Stage 5 and Stage 6 configurations exhibited performance metrics firmly within the $0.8- - 0.9$ range, indicating that the network was becoming increasingly robust. Finally, the detailed examination of pooling variations in Stages 7 and 8 revealed that optimized settings—particularly those using an 11×11 pooling window with specific stride and padding choices—yielded peak results, with precision values reaching up to 0.8976, recall up to 0.8920, and F1 scores as high as 0.8948. From the observation, these findings confirm that the careful tuning of convolutional and pooling parameters is critical for achieving high performance on the LC105K dataset, and they provide clear guidance for selecting the optimal network configuration for future applications.

The simulation study summarized in Table IX systematically evaluates the LC105K network across eight stages by varying convolutional filter sizes, numbers of filters, stride sizes, and pooling parameters. In stage 1, the network configurations with small filters (from 1×1 to 13×13) yielded specificity values in the range of 0.9400–0.9750 and AUC values between 0.8850 and 0.9100. As the architecture advanced to stage 2, increasing the number of filters and fine-tuning the convolution settings led to an overall improvement in performance (specificity up to 0.9800 and AUC as high as 0.9200). Stage 3 further refined the parameters, with specificity and AUC consistently remaining in a narrow high-performance band. In stage 4, modifications in stride and padding produced a steady incremental gain, achieving specificity values above 0.9800 and AUC around 0.9250. The transition to average pooling in stage 5 provided a slight boost in both specificity and AUC, which continued to improve through stages 6 and 7, where specificity reached nearly 1.0000 and AUC approached 0.9400. Finally, the detailed exploration in stage 8—with incremental adjustments in the pooling padding—resulted in the highest recorded performance, with specificity rising from 0.9700 to 0.9800 and AUC improving from 0.9100 to 0.9200. These results collectively indicate that meticulous tuning of convolution and pooling parameters is essential for optimizing the LC105K network's diagnostic accuracy.

VI. CONCLUSION

This research presents a deep learning-based approach for classifying three primary lung cancer types. The introduction of 150000 images with varied dimensions, provides a robust foundation for this study. The proposed convolutional neural network architecture was carefully designed with additional layers to optimize the feature extraction and classification. A systematic parameter optimization process was employed, considering the filter size, stride, pooling filter size, and padding size. The trained LC105K model yielded an exceptional accuracy of 99.38%, outperforming existing research. These findings highlight the superior performance of our proposed method.

To improve task performance, an augmented multichannel convolutional neural network combines the advantages

TABLE VIII: Precision, Recall, and F1 Scores for the LC105K Dataset

No of Stage	Convolution				Pooling				Precision	Recall	F1 Score
	Filter Size (Pixels)	No. of Filters (filters)	Stride Size (Pixels)	Padding Size	Type	Window Size (Pixels)	Stride Size (Pixels)	Padding Size			
Stage 1	1×1	2	1×1	0	Max.	3×3	3×3	0	0.7872	0.7249	0.7181
	3×3	2	1×1	0	Max.	3×3	3×3	0	0.7842	0.7219	0.7121
	5×5	2	1×1	0	Max.	3×3	3×3	0	0.6342	0.6213	0.6081
	7×7	2	1×1	0	Max.	3×3	3×3	0	0.8812	0.8640	0.8731
	9×9	2	1×1	0	Max.	3×3	3×3	0	0.8916	0.8847	0.8311
	11×11	2	1×1	0	Max.	3×3	3×3	0	0.8242	0.7941	0.8119
	13×13	2	1×1	0	Max.	3×3	3×3	0	0.8245	0.7951	0.8112
Stage 2	9×9	2	1×1	0	Max.	3×3	3×3	0	0.8812	0.8617	0.8019
	9×9	4	1×1	0	Max.	3×3	3×3	0	0.8806	0.8531	0.8001
	9×9	6	1×1	0	Max.	3×3	3×3	0	0.7901	0.7847	0.7501
	9×9	8	1×1	0	Max.	3×3	3×3	0	0.8506	0.8317	0.8019
	9×9	10	1×1	0	Max.	3×3	3×3	0	0.8516	0.8341	0.8001
	9×9	12	1×1	0	Max.	3×3	3×3	0	0.8971	0.8874	0.8701
	9×9	14	1×1	0	Max.	3×3	3×3	0	0.8901	0.8781	0.8756
Stage 3	9×9	4	1×1	0	Max.	3×3	3×3	0	0.8923	0.8611	0.8701
	9×9	4	2×2	0	Max.	3×3	3×3	0	0.8400	0.8300	0.8350
	9×9	4	3×3	0	Max.	3×3	3×3	0	0.8500	0.8600	0.8550
	9×9	4	4×4	0	Max.	3×3	3×3	0	0.8600	0.8700	0.8650
	9×9	4	5×5	0	Max.	3×3	3×3	0	0.8800	0.8900	0.8850
Stage 4	9×9	4	1×1	0	Max.	3×3	3×3	0	0.8100	0.8200	0.8150
	9×9	4	2×2	1	Max.	3×3	3×3	0	0.8200	0.8300	0.8250
	9×9	4	3×3	2	Max.	3×3	3×3	0	0.8300	0.8400	0.8350
	9×9	4	4×4	3	Max.	3×3	3×3	0	0.8400	0.8500	0.8450
	9×9	4	5×5	4	Max.	3×3	3×3	0	0.8600	0.8700	0.8650
	9×9	4	5×5	Same	Max.	3×3	3×3	0	0.8800	0.8900	0.8850
Stage 5	9×9	4	1×1	0	Max.	3×3	3×3	0	0.8976	0.8880	0.8928
	9×9	4	1×1	0	Ave.	3×3	3×3	0	0.8829	0.8350	0.8455
	9×9	4	1×1	0	Max.	5×5	3×3	0	0.8820	0.8750	0.8785
	9×9	4	1×1	0	Max.	7×7	3×3	0	0.8020	0.8050	0.8785
	9×9	4	1×1	0	Max.	9×9	3×3	0	0.8120	0.8751	0.8685
	9×9	4	1×1	0	Max.	11×11	3×3	0	0.8820	0.8750	0.8785
Stage 6	9×9	4	1×1	0	Ave.	3×3	3×3	0	0.8123	0.8234	0.8178
	9×9	4	1×1	0	Ave.	5×5	3×3	0	0.8325	0.8400	0.8362
	9×9	4	1×1	0	Ave.	7×7	3×3	0	0.8476	0.8550	0.8512
	9×9	4	1×1	0	Ave.	9×9	3×3	0	0.8600	0.8680	0.8640
	9×9	4	1×1	0	Ave.	11×11	3×3	0	0.8765	0.8820	0.8792
	9×9	4	1×1	0	Ave.	13×13	3×3	0	0.8976	0.8920	0.8948
Stage 7	9×9	4	1×1	0	Ave.	11×11	2×2	0	0.8123	0.8210	0.8165
	9×9	4	1×1	0	Ave.	11×11	3×3	0	0.8325	0.8400	0.8362
	9×9	4	1×1	0	Ave.	11×11	4×4	0	0.8476	0.8550	0.8512
	9×9	4	1×1	0	Ave.	11×11	5×5	0	0.8600	0.8680	0.8640
	9×9	4	1×1	0	Ave.	11×11	6×6	0	0.8750	0.8820	0.8785
	9×9	4	1×1	0	Ave.	11×11	7×7	0	0.8976	0.8920	0.8948
Stage 8	9×9	4	1×1	0	Ave.	11×11	3×3	0	0.8123	0.8210	0.8165
	9×9	4	1×1	0	Ave.	11×11	3×3	1	0.8325	0.8400	0.8362
	9×9	4	1×1	0	Ave.	11×11	3×3	2	0.8476	0.8550	0.8512
	9×9	4	1×1	0	Ave.	11×11	3×3	3	0.8600	0.8680	0.8640
	9×9	4	1×1	0	Ave.	11×11	3×3	4	0.8750	0.8820	0.8785
	9×9	4	1×1	0	Ave.	11×11	3×3	Same	0.8976	0.8920	0.8948

TABLE IX: Specificity, AUC, and Parameter Evaluation for the LC105K Dataset

No of Stage	Convolution				Pooling				Specificity	AUC
	Filter Size (Pixels)	No. of Filters (filters)	Stride Size (Pixels)	Padding Size (Bordered Pixels)	Type	Window Size (Pixels)	Stride Size (Pixels)	Padding Size (Bordered Pixels)		
Stage 1	1×1	2	1×1	0	Max.	3×3	3×3	0	0.9500	0.8900
	3×3	2	1×1	0	Max.	3×3	3×3	0	0.9520	0.8910
	5×5	2	1×1	0	Max.	3×3	3×3	0	0.9400	0.8850
	7×7	2	1×1	0	Max.	3×3	3×3	0	0.9700	0.9050
	9×9	2	1×1	0	Max.	3×3	3×3	0	0.9750	0.9100
	11×11	2	1×1	0	Max.	3×3	3×3	0	0.9680	0.9070
	13×13	2	1×1	0	Max.	3×3	3×3	0	0.9670	0.9060
Stage 2	9×9	2	1×1	0	Max.	3×3	3×3	0	0.9700	0.9110
	9×9	4	1×1	0	Max.	3×3	3×3	0	0.9720	0.9120
	9×9	6	1×1	0	Max.	3×3	3×3	0	0.9650	0.9080
	9×9	8	1×1	0	Max.	3×3	3×3	0	0.9750	0.9150
	9×9	10	1×1	0	Max.	3×3	3×3	0	0.9760	0.9160
	9×9	12	1×1	0	Max.	3×3	3×3	0	0.9800	0.9200
	9×9	14	1×1	0	Max.	3×3	3×3	0	0.9790	0.9190
Stage 3	9×9	4	1×1	0	Max.	3×3	3×3	0	0.9780	0.9180
	9×9	4	2×2	0	Max.	3×3	3×3	0	0.9750	0.9150
	9×9	4	3×3	0	Max.	3×3	3×3	0	0.9760	0.9160
	9×9	4	4×4	0	Max.	3×3	3×3	0	0.9770	0.9170
	9×9	4	5×5	0	Max.	3×3	3×3	0	0.9785	0.9185
Stage 4	9×9	4	1×1	0	Max.	3×3	3×3	0	0.9800	0.9200
	9×9	4	2×2	1	Max.	3×3	3×3	0	0.9810	0.9210
	9×9	4	3×3	2	Max.	3×3	3×3	0	0.9820	0.9220
	9×9	4	4×4	3	Max.	3×3	3×3	0	0.9830	0.9230
	9×9	4	5×5	4	Max.	3×3	3×3	0	0.9840	0.9240
	9×9	4	5×5	Same	Max.	3×3	3×3	0	0.9850	0.9250
Stage 5	9×9	4	1×1	0	Max.	3×3	3×3	0	0.9860	0.9260
	9×9	4	1×1	0	Ave.	5×3	3×3	0	0.9170	0.9071
	9×9	4	1×1	0	Max.	5×5	3×3	0	0.9260	0.9160
	9×9	4	1×1	0	Max.	7×7	3×3	0	0.9060	0.9660
	9×9	4	1×1	0	Max.	9×9	3×3	0	0.9760	0.9460
Stage 6	9×9	4	1×1	0	Ave.	3×3	3×3	0	0.9880	0.9280
	9×9	4	1×1	0	Ave.	5×5	3×3	0	0.9890	0.9290
	9×9	4	1×1	0	Ave.	7×7	3×3	0	0.9900	0.9300
	9×9	4	1×1	0	Ave.	9×9	3×3	0	0.9910	0.9310
	9×9	4	1×1	0	Ave.	11×11	3×3	0	0.9920	0.9320
	9×9	4	1×1	0	Ave.	13×13	3×3	0	0.9930	0.9330
Stage 7	9×9	4	1×1	0	Ave.	11×11	2×2	0	0.9940	0.9340
	9×9	4	1×1	0	Ave.	11×11	3×3	0	0.9950	0.9350
	9×9	4	1×1	0	Ave.	11×11	4×4	0	0.9960	0.9360
	9×9	4	1×1	0	Ave.	11×11	5×5	0	0.9970	0.9370
	9×9	4	1×1	0	Ave.	11×11	6×6	0	0.9980	0.9380
	9×9	4	1×1	0	Ave.	11×11	7×7	0	0.9990	0.9390
Stage 8	9×9	4	1×1	0	Ave.	11×11	3×3	0	0.9700	0.9100
	9×9	4	1×1	0	Ave.	11×11	3×3	1	0.9720	0.9120
	9×9	4	1×1	0	Ave.	11×11	3×3	2	0.9740	0.9140
	9×9	4	1×1	0	Ave.	11×11	3×3	3	0.9760	0.9160
	9×9	4	1×1	0	Ave.	11×11	3×3	4	0.9780	0.9180
	9×9	4	1×1	0	Ave.	11×11	3×3	Same	0.9800	0.9200

of data augmentation techniques with those of a multichannel CNN architecture. This approach incorporates data augmentation to expand the training dataset artificially by applying various alterations to preexisting data. Concurrently, multichannel CNN exploits their capacity to handle many input channels and efficiently extract detailed and subtle characteristics from data. With this combination strategy, the network is better able to interpret complicated patterns and expand its learning capacity, leading to improved an overall performance.

The work described in this article makes a significant contribution to lung cancer diagnosis. To this end, an AI-based model was integrated with a CNN to unravel the complexities of lung cancer detection. By utilizing a vast dataset of vivid lung cancer images, this advanced model aims to enhance the diagnostic accuracy and achieve an accuracy rate of 99.38%. Furthermore, this model was skillfully integrated with AMC-CNNs to improve lung cancer classification. By leveraging the same comprehensive dataset of lung cancer images, this advanced model sought to establish a reliable and efficient diagnosis process, achieving an accuracy rate of 89.12%.

ACKNOWLEDGMENT

The authors thank Prof. Raid Rafi Omar Al-Nima, Department of Medical Instrumentation Technology Engineering, Technical Engineering College of Mosul, Northern Technical University, Iraq.

REFERENCES

[1] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis detection in breast cancer histology images with deep neural networks," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013: 16th International Conference, Nagoya, Japan, September 22-26, 2013, Proceedings, Part II 16*. Springer, 2013, pp. 411–418.

[2] A. B. Hamida, M. Devanne, J. Weber, C. Truntzer, V. Derangère, F. Ghiringhelli, G. Forestier, and C. Wemmert, "Deep learning for colon cancer histopathological images analysis," *Computers in Biology and Medicine*, vol. 136, p. 104730, 2021.

[3] J. Xie, R. Liu, J. Luttrell IV, and C. Zhang, "Deep learning based analysis of histopathological images of breast cancer," *Frontiers in genetics*, vol. 10, p. 80, 2019.

[4] H. Sharma, N. Zerbe, I. Klempert, O. Hellwich, and P. Hufnagl, "Deep convolutional neural networks for automatic classification of gastric carcinoma using whole slide images in digital histopathology," *Computerized Medical Imaging and Graphics*, vol. 61, pp. 2–13, 2017.

[5] A. Barsi, S. C. Nayak, S. Parida, and R. M. Shukla, "A deep learning-based compression and classification technique for whole slide histopathology images," *International Journal of Information Technology*, pp. 1–10, 2024.

[6] N. Girdhar, A. Sinha, and S. Gupta, "Densenet-ii: An improved deep convolutional neural network for melanoma cancer detection," *Soft computing*, vol. 27, no. 18, pp. 13 285–13 304, 2023.

[7] M. A. Wakili, H. A. Shehu, M. H. Sharif, M. H. U. Sharif, A. Umar, H. Kusetogullari, I. F. Ince, and S. Uyaver, "Classification of breast cancer histopathological images using densenet and transfer learning," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, p. 8904768, 2022.

[8] N. Kumar, M. Sharma, V. P. Singh, C. Madan, and S. Mehandia, "An empirical study of handcrafted and dense feature extraction techniques for lung and colon cancer classification from histopathological images," *Biomedical Signal Processing and Control*, vol. 75, p. 103596, 2022.

[9] D. Ardila, A. P. Kiraly, S. Bharadwaj, B. Choi, J. J. Reicher, L. Peng, D. Tse, M. Etemadi, W. Ye, G. Corrado *et al.*, "End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography," *Nature medicine*, vol. 25, no. 6, pp. 954–961, 2019.

[10] V. Nemade, S. Pathak, and A. K. Dubey, "Deep learning-based ensemble model for classification of breast cancer," *Microsystem Technologies*, vol. 30, no. 5, pp. 513–527, 2024.

[11] N. Behera, A. P. Singh, J. K. Rout, and B. K. Balabantaray, "Melanoma skin cancer detection using deep learning-based lesion segmentation," *International Journal of Information Technology*, pp. 1–16, 2024.

[12] H. C. Tingting Wang, Qi Fan and B. Zhang, "Application of machine learning for tracing the origin of metastatic lung cancer tissues," *IAENG International Journal of Computer Science*, vol. 50, no. 2, pp. 359–367, 2023.

[13] M. Othmani, B. Issaoui, S. El Khediri, and R. U. Khan, "Hybrid active shape model and deep neural network approach for lung cancer detection," *International Journal of Information Technology*, pp. 1–12, 2024.

[14] N. Kaur and A. Singh, "Vgg16-pca-pb3c: A hybrid pb3c and deep neural network based approach for leukemia detection," *International Journal of Information Technology*, pp. 1–11, 2024.

[15] O. Singh and K. K. Singh, "An approach to classify lung and colon cancer of histopathology images using deep feature extraction and an ensemble method," *International Journal of Information Technology*, vol. 15, no. 8, pp. 4149–4160, 2023.

[16] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[17] X. Wang, I. Ahmad, D. Javeed, S. A. Zaidi, F. M. Alotaibi, M. E. Ghoneim, Y. I. Daradkeh, J. Asghar, and E. T. Eldin, "Intelligent hybrid deep learning model for breast cancer detection," *Electronics*, vol. 11, no. 17, p. 2767, 2022.

[18] T. Liu, J. Huang, T. Liao, R. Pu, S. Liu, and Y. Peng, "A hybrid deep learning model for predicting molecular subtypes of human breast cancer using multimodal data," *Irbm*, vol. 43, no. 1, pp. 62–74, 2022.

[19] E. Farea, R. A. Saleh, H. AbuAlkebash, A. A. Farea, and M. A. Al-antari, "A hybrid deep learning skin cancer prediction framework," *Engineering Science and Technology, an International Journal*, vol. 57, p. 101818, 2024.

[20] B. S. Gowri, S. A. H. Nair, and K. Kumar, "Hybrid arithmetic optimization algorithm with deep transfer learning based microarray gene expression classification model," *International Journal of Information Technology*, pp. 1–6, 2024.

[21] S. Jain and V. Jain, "Novel approach to classify brain tumor based on transfer learning and deep learning," *International Journal of Information Technology*, vol. 15, no. 4, pp. 2031–2038, 2023.

[22] T.-j. Luo, C.-l. Zhou, and F. Chao, "Exploring spatial-frequency-sequential relationships for motor imagery classification with recurrent neural network," *BMC bioinformatics*, vol. 19, pp. 1–18, 2018.

[23] H. V. Dudukcu, M. Taskiran, Z. G. C. Taskiran, and T. Yildirim, "Temporal convolutional networks with rnn approach for chaotic time series prediction," *Applied Soft Computing*, vol. 133, p. 109945, 2023.

[24] L. M. A. El-Magd, G. Dahy, T. A. Farrag, A. Darwish, and A. E. Hassnien, "An interpretable deep learning based approach for chronic obstructive pulmonary disease using explainable artificial intelligence," *International Journal of Information Technology*, pp. 1–16, 2024.

[25] N. Y. Muhammad Wildan Oktavian and A. Ridok, "Classification of alzheimer's disease using the convolutional neural network (cnn) with transfer learning and weighted loss," *IAENG International Journal of Computer Science*, vol. 50, no. 3, pp. 947–953, 2023.

[26] Y. Qiao and L. Xing, "Deep learning based oracle style migration and generation techniques," *IAENG International Journal of Computer Science*, vol. 51, no. 1, pp. 1872–1879, 2024.

[27] P. W. Ling Wang, Kexin Luo and Y. Bai, "Camtnet: Ctc-attention mechanism and transformer fusion network for scene text recognition," *IAENG International Journal of Computer Science*, vol. 51, no. 1, pp. 1750–1760, 2024.

[28] K. Alomar, H. I. Aysel, and X. Cai, "Data augmentation in classification and segmentation: A survey and new strategies," *Journal of Imaging*, vol. 9, no. 2, p. 46, 2023.

[29] E. Lashgari, D. Liang, and U. Maoz, "Data augmentation for deep-learning-based electroencephalography," *Journal of Neuroscience Methods*, vol. 346, p. 108885, 2020.

[30] C. Shorten and T. M. Khoshgoftaar, "Investigating the generalization of image classifiers with augmented test sets," in *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2021, pp. 10–17.

[31] V. Maslej-Krešňáková, K. El Bouchefry, and P. Butka, "Morphological classification of compact and extended radio galaxies using convolutional neural networks and data augmentation techniques," *Monthly Notices of the Royal Astronomical Society*, vol. 505, no. 1, pp. 1464–1475, 2021.

[32] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[33] H. Panwar, P. Gupta, M. K. Siddiqui, R. Morales-Menendez, P. Bhardwaj, and V. Singh, "A deep learning and grad-cam based color

visualization approach for fast detection of covid-19 cases using chest x-ray and ct-scan images," *Chaos, Solitons & Fractals*, vol. 140, p. 110190, 2020.

[34] K. Raghavan, "Attention guided grad-cam: an improved explainable artificial intelligence model for infrared breast cancer detection," *Multimedia Tools and Applications*, vol. 83, no. 19, pp. 57 551–57 578, 2024.

[35] J. A. Marmolejo-Saucedo and U. Kose, "Numerical grad-cam based explainable convolutional neural network for brain tumor diagnosis," *Mobile Networks and Applications*, pp. 1–10, 2022.

[36] I. Matas, C. Serrano, F. Silva, A. Serrano, T. Toledo-Pastrana, and B. Acha, "Ai-driven skin cancer diagnosis: Grad-cam and expert annotations for enhanced interpretability," *arXiv preprint arXiv:2407.00104*, 2024.

[37] S. Rasappan and S. A. Nishath, "Lungcancer105k(lc105k)," 2024. [Online]. Available: https://www.kaggle.com/dsv/9652614

[38] S. O. Ali, R. R. Al-Nima, and E. A. Mohammed, "Individual recognition with deep earprint learning," in *2021 International Conference on Communication & Information Technology (ICICT)*. IEEE, 2021, pp. 304–309.

[39] L. H. Albak, R. R. O. Al-Nima, and A. H. Salih, "Palm print verification based deep learning," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 3, pp. 851–857, 2021.

[40] M. AL-Hatab, R. R. O. Al-Nima, I. Marcantoni, C. Porcaro, and L. Burattini, "Comparison study between three axis views of vision, motor and pre-frontal brain activities," *Journal of Critical Reviews*, vol. 7, no. 5, pp. 2598–2607, 2020.

[41] S. Rasappan and A. N. S, "Lungcancer105k (lc105k): A large-scale histopathological image dataset for lung cancer diagnosis," 2024. [Online]. Available: https://dx.doi.org/10.21227/venm-xk08

[42] A. M. Ibrahim, A. K. Eesee, and R. R. O. Al-Nima, "Deep fingerprint classification network," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 3, pp. 893–901, 2021.

[43] A. J. Hepworth, D. P. Baxter, A. Hussein, K. J. Yaxley, E. Debie, and H. A. Abbass, "Human-swarm-teaming transparency and trust architecture," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 7, pp. 1281–1295, 2020.

[44] L. B. Nascimento, A. C. de Paiva, and A. C. Silva, "Lung nodules classification in ct images using shannon and simpson diversity indices and svm," in *Machine Learning and Data Mining in Pattern Recognition: 8th International Conference, MLDM 2012, Berlin, Germany, July 13-20, 2012. Proceedings 8*. Springer, 2012, pp. 454–466.

[45] H. Krewer, B. Geiger, L. O. Hall, D. B. Goldgof, Y. Gu, M. Tockman, and R. J. Gillies, "Effect of texture features in computer aided diagnosis of pulmonary nodules in low-dose computed tomography," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2013, pp. 3887–3891.

[46] E. Dandıl, M. Çakiroğlu, Z. Ekşi, M. Özkan, Ö. K. Kurt, and A. Canan, "Artificial neural network-based classification system for lung nodules on computed tomography scans," in *2014 6th International conference of soft computing and pattern recognition (SoCPaR)*. Ieee, 2014, pp. 382–386.

[47] J. Kuruvilla and K. Gunavathi, "Lung cancer classification using neural networks for ct images," *Computer Methods and Programs in Biomedicine*, vol. 113, no. 1, pp. 202–209, 2014.

[48] Q. Song, L. Zhao, X. Luo, and X. Dou, "Using deep learning for classification of lung nodules on computed tomography images," *Journal of healthcare engineering*, vol. 2017, no. 1, p. 8314740, 2017.

Suresh Rasappan is a Senior Lecturer in the Department of Mathematics at the University of Technology and Applied Sciences, Ibri, Sultanate of Oman. He holds a Ph.D. in Mathematics and has over a decade of academic experience in teaching, research, and academic leadership. In 2014, he completed the Indian Academy Summer project at the Indian Institute of Science (IISc), Bangalore.

Dr. Rasappan's research interests include Differential Equations, Chaos Theory, Control Theory, Machine Learning, and Deep Learning. He has published more than 100 research articles in reputable journals indexed in Scopus and Web of Science. He currently supervises doctoral students and has served as examiner for multiple Ph.D. theses.

He is the recipient of several prestigious awards, including the Best Paper Award at the ICMMCS-2020 conference hosted by Lincoln University College, Malaysia. He serves as Academic Editor for the journal Computational and Mathematical Methods and has received internal funding for his research in deep learning-based cancer detection.

S. Ahamed Nishath is a Lecturer in the Department of Information Technology at the University of Technology and Applied Sciences, Ibri, Sultanate of Oman. He has over a decade of experience in teaching, research, and academic administration. He holds certifications and training in advanced computing to enhance his academic practice.

His research interests include Machine Learning, Deep Learning, and related areas in artificial intelligence. His work focuses on developing innovative approaches to complex problems using modern computational models and has earned recognition across educational and research communities.

He is an active member of several prestigious professional organizations including the International Association of Engineers (IAENG) and the Association for Computing Machinery (ACM).

Francis Saviour Devaraj is a Senior Lecturer in the Department of Information Technology at the University of Technology and Applied Sciences, Ibri, Sultanate of Oman. He has over 20 years of experience in teaching, research, and administration. He has completed several certifications and training programs to support his academic and professional development.

His research interests include Cybersecurity, Threat Analysis, and related fields. His work has contributed to developing secure frameworks and intelligent models for analyzing cyber threats in real-time. He is recognized for his commitment to both research excellence and quality teaching.

He is a life member of the Computer Society of India and currently serves as the Chair of the ACM Professional Chapter. He is also an active member of the Association for Computing Machinery (ACM), contributing to its initiatives in advancing computer science education and research.