**Hybrid CNN-LSTM Network for Lung Cancer Detection from Chest X-ray Images**

**Abstract.** This study investigates a novel hybrid technique that integrates deep learning and recurrent neural networks to identify lung cancer in chest X-ray images. The researchers leveraged a dataset of 247 chest X-rays obtained from the Kaggle platform's JSRT (Japanese Society for Radiological Technology) collection. This dataset included 56 X-rays with confirmed lung cancer and 191 without. The inserted image was processed by changing its size to 480 × 480 pixels with the same extension (png) without affecting the image quality. Normalization is an interesting advanced processing step in image processing applications, and data augmentation techniques have also been used to make the data look more diverse. Three incremental strategies have been used to create new training groups, called sphenic conversion (rotation, shear, scale). Then CCN's VGG16 development was used with higher generalization and accuracy compared to other networks in terms of extracting different level features, producing a vector with all the distinctive characteristics associated with LSTM which keeps data in weights form during the training to do the classification. The proposed technique achieved impressive recognition rates, with a testing accuracy of 90.99%. This result translates to highly accurate detection and classification of lung cancer in chest X-rays, differentiating between cancerous and non-cancerous lung images. These findings highlight the promising potential of deep learning for various medical applications. Index Terms Deep Learning, Recurrent Neural Network, Lung Cancer, Convolutional Neural Network, Long Short-Term Memory.

***Keywords:*** *Deep Learning, Recurrent Neural Network, Lung Cancer, Convolutional Neural Network, Long Short-Term Memory.*

1 **Introduction**

While many cancers pose a serious threat, lung cancer stands out as the deadliest, claiming more lives than any other cancer type among both men and women globally (Chapaliuk and Zaychenko, 2018). Early detection of lung cancer will improve the survival rate from 2% to 47%, compared to detection at the fifth stage. Despite this, only 15% of early-stage lung cancers are currently detected (Sun, 2017). Analyzing changes in images of infected areas in patients suspected of early-stage lung cancer could help prevent more severe complications. X-rays are a common method for detecting lung disease.

X-ray images result from recording the differential absorption of X-ray radiation as it passes through various tissues in the body. Tissues with high radiation absorption, such as bones, appear white due to their high density. Conversely, fat and soft tissues, which absorb less radiation, are rendered in shades of gray. Air absorbs minimal X-ray radiation, appearing black. Therefore, the lungs, being air-filled organs, are depicted as dark regions on an X-ray image (Moturu and Chang, 2018).

Over the past years, the increase in computer solutions has enabled significant improvements in the healthcare sector. Machine learning and deep learning have consistently delivered high-accuracy predictions in various research areas related to health diagnosis. Deep learning techniques will be investigated and compared to help clinicians suggest the most suitable method for identifying changes in affected areas. Whereas Neural Networks (NN) are making impressive progress in computer vision tasks, Convolutional Neural Networks (CNN) have achieved better performance than NN and humans (Garcia et al., 2017).

Many data sets representing different types of cancer are categorized using computer-assisted diagnostic techniques (Jeyaraj and Samuel Nadar, 2019). The methods used have imposed different trade-offs on practitioners. Deep learning techniques are employed to detect, identify, and classify various types of cancer (Benhammou et al., 2020). CNN diagnoses cancer by distinguishing between tumor cell areas (Bhargavi et al., 2024). In addition, several methods have recently been used in cancer detection, including 3D torsion networks that fragment the process of examining and categorizing patients into the most malignant sectors, the frequent neural network is used to know the dependency between patient segments, a deep learning framework in H2O platform is also used to detect head and neck cancer (Shimazaki et al. 2022).

Additionally, numerous other deep-learning techniques are available for cancer detection and prediction (Khuriwal and Mishra, 2018). LSTM is the most widely used Recurrent Neural Network (RNN) for classifying or predicting sequential data (Gao et al., 2019). Due to its features, it was used in this research.

This research primarily addresses the diagnosis of lung cancer disease using a proposed deep learning technique and LSTM to classify images and explore potential lung cancer. This process will improve the existing manual methods used by doctors to detect lung cancer. Additionally, early disease detection increases the patient's survival rate by enabling the prescription of treatment or the implementation of necessary medical measures. Inspection of deep Learning in analyzing images of infected areas can provide a more coherent summary of the disease and support specialists in their decisions.

This paper is structured as follows: Section 2 introduces some previous works; Section 3 presents the Methodology of the Proposed Model applied to X-ray images of a human chest to distinguish between infected and normal.

Ones. The results of the conducted experiments and the conclusions are presented in Sections 5 and 6, respectively.

2 **Related Work**

Lung cancer reigns as the deadliest and most prevalent cancer globally, accounting for 12.3% of all cancers and claiming the most lives among both men and women (Tsukamoto et al., 2022; Jenkins et al., 2023). In the United States, it remains the top cause of cancer death for both genders (Thandra et al., 2021). Since chest X-rays are the primary tool for diagnosing lung issues, accurately distinguishing between cancerous and non-cancerous lungs becomes crucial. Researchers proposed several techniques for cancer detection using deep learning. For instance, Shimizu et al. 2016) developed a diagnosis system for lung cancer using deep learning. This model uses Gas Chromatography-Mass Spectrometry (GC-MS) to convert human urine into 3D data, which then serves as input for a Neural Network. The neural network's hidden layer is a Stacked Autoencoder, essentially a multilayer autoencoder. GC-MS distinguishes 364 features in urine based on retention time and mass-to-charge ratio, with each feature having a value representing its ionic strength. Since the Stacked Autoencoder uses a sigmoid activation function, requiring outputs between 0 and 1, the input data was normalized to this range. The researchers experimented to determine which normalization method was more effective. This system achieved 90% accuracy in diagnosing lung cancer. This approach promises a non-invasive and easy method for personalized diagnosis, as urine collection is simple and harmless.

In another study, the authors expand upon existing deep learning approaches to cancer detection (Gupta and Kaur Malhi, 2018) by leveraging H2O, a powerful open-source platform known for its speed, scalability, and ability to streamline machine learning and deep learning workflows. The researchers utilized a vast dataset of over 26,000 CT scan images from The Cancer Imaging Archive (TCIA), a valuable resource for medical images. This study introduces a deep learning framework within H2O specifically designed to detect head and neck cancer solely through CT scans. To achieve this result, the framework leverages various image processing techniques. First, preprocessing enhances the image quality. Next, Segmentation isolates relevant regions within the image for analysis. Finally, feature extraction identifies critical parameters from the segmented areas. These extracted features then serve as the foundation for the classification stage, where the presence or absence of head and neck cancer is determined.

Teramoto et al. Gupta and Kaur Malhi (2018) pioneered an automated system for classifying lung cancer in microscopic images. Their approach leveraged a deep convolutional neural network (DCNN), a cornerstone technique in deep learning. This DCNN architecture comprised three convolutional layers for feature extraction, followed by three pooling layers for dimensionality reduction, and culminated in two fully connected layers for final classification. The image dataset comprises seventy-six (76) cancer cell cases.

Collected via exfoliative or interventional cytology under bronchoscopy or CT-guided needle aspiration cytology.

The researchers evaluated their DCNN on a dataset containing 40 cases of adenocarcinoma, 20 cases of squamous cell carcinoma, and 16 cases of small cell carcinoma. While the results yielded an accuracy of approximately 70%, further research is needed to improve upon this. However, these findings demonstrate the potential of DCNNs as a valuable tool for lung cancer classification and cyst diagnosis.

LSTM is a widely used Recurrent Neural Network (RNN) for classifying and predicting sequential data, particularly relevant in cancer research. To address this, Gao et al. (2019) introduced the Distanced LSTM (DLSTM) model. This model is a generalization of the standard LSTM, specifically designed to handle both regular and irregular longitudinal samples. A key component of the DLSTM is the Temporal Emphasis Model (TEM), which allows the model to learn effectively from both regularly and irregularly sampled time intervals. The contributions of this research include representing the first study to model the time distance from the last point for LSTM in lung cancer detection, proposing the novel DLSTM framework to model temporal distance with adaptive forget and input gates effectively, and releasing a toy dataset called Tumor-CIFAR, which simulates benign and malignant cancer on natural images for research purposes. They evaluated the proposed model using a substantial dataset, including 1794 subjects from the National Lung Screening Trial (NLST) and

1420 subjects from two institutional cohorts.

3 **Methodology of the Proposed Model**

This Section discusses the theoretical part of the project, which is divided into several parts. Many deep-learning techniques are used to detect and classify different types of cancer. According to the characteristics of CNN and LSTM, a lung cancer forecasting model based on CNN-LSTM is established. However, the forecasting accuracy of CNN alone is relatively low; therefore, using the LSTM neural network, which has a high forecasting accuracy and can effectively predict the time series of lung cancer (Lu et al., 2020), is a vital process. Moreover, combining the advantages of CNN, which can extract compelling features from the data, and LSTM has the benefit of analyzing relationships among time series data through its memory function. The Proposed technique used in this paper is described in the following paragraphs:

**3.1 System Framework**

The research methodology is summarized as follows: The proposed method is illustrated in Figure 1, which provides a general overview of the research approach and its architecture. This figure illustrates the use of the X-ray image

dataset to distinguish between infected and non-infected lungs. Then, the system will use the stored data to interpret it into a suitable form and show the results to the responsible person.

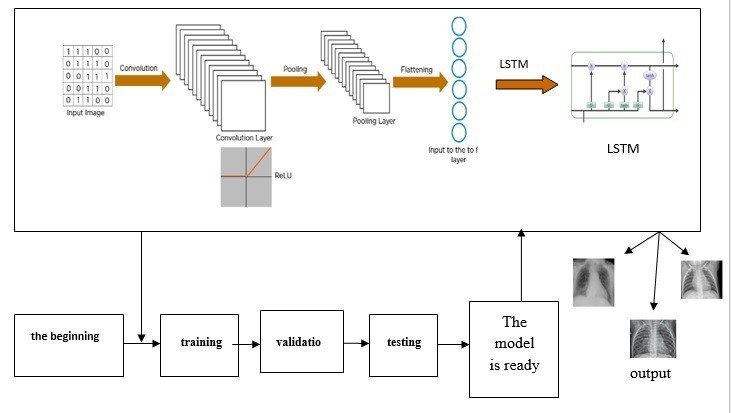


Fig.1. An overview of the proposed methodology.

**3.2 Dataset Description**

This study leverages a real-world dataset obtained from the Kaggle platform. This dataset originates from the JSRT (Japanese Society of Radiological Technology), a valuable resource for researchers worldwide. The JSRT database has been a cornerstone for various research endeavors in image processing, compression, display evaluation, medical record archiving (PACS), and computer-aided diagnosis, utilizing machine learning techniques for over two decades. The widespread adoption of this dataset is evident by its 326 citations (as of February 19, 2021) across research articles, conference proceedings, reviews, and book chapters, according to the Web of Science Core Collection (*http*: *//db.jsrt.or.jp/eng.php*). The specific c source employed here (Automatic Lung Cancer Prediction from Chest X-ray Images Using Deep Learning Approach) has also been utilized in other relevant studies, including Dimensionality Reduction in Deep Learning for Chest X-Ray Analysis of Lung Cancer and Deep Learning with Lung Segmentation and Bone Shadow Exclusion Techniques for Chest X-Ray Analysis of Lung Cancer.

The dataset contains 154 chest radiographs with a lung nodule (confirmed malignant = 100, confirmed benign = 54) and 93 typical images, with a matrix size of (2048*x*2048) (0.175 mm pixels) and a 12-*bit* grayscale (no header, big endian raw data). The database also includes additional information, such as

patient age, gender, diagnosis (malignant or benign), X and Y coordinates of a nodule, and a simple diagram of the nodule's location. Lung nodule images were classified into five groups based on the degree of subtlety.

4 **The Proposed Model**

The proposed model combines convolutional neural networks (CNNs) and long short-term memory (LSTM) to detect infected X-ray images. In contrast to LSTM, it is used to analyze data because of its capability of learning order dependence in sequence prediction problems. It is well-suited to classifying, processing, and making predictions based on time series data. The model is tested with our dataset. It is showing promising results in estimating the presence of cancer in the lungs. The details are presented in the following Sections.

**4.1 Convolutional Neural Networks (CNN)**

CNNs are a class of Deep Neural Networks that can distinguish and classify features from images (Wang et al., 2021). Conventional neural networks consist of multiple layers of neurons, with each neuron typically connected to all neurons in adjacent layers. Each connection has a unique weight (Rajpurkar et al., 2017). Five blocks of the CNN Model comprised the convolutional layer used to build the proposed model. Max pooling and batch normalization are applied on top of the blocks. An attended layer is followed by LSTM layers as well. Additionally, the dropouts are used in between to reduce overfitting. The activation function was Relu throughout, except for the last layer, where it was Sigmoid.

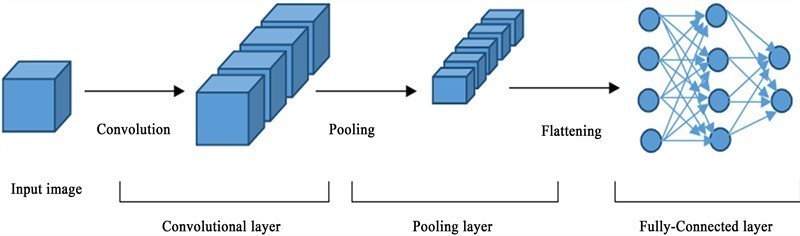


Fig.2. Convolutional Neural Networks. (Ahmed et al., 2020)

**Block 1** *is the first convolutional* layer and contains Conv2D and MaxPool2D.

Conv2D: Conv2D is a 2*D* Convolution Layer; the most common type of convolution that is used is the 2*D* convolution layer. Conv2D layers are generally used to achieve high accuracy in image recognition tasks,

applying a 2D convolution over an input signal composed of multiple input planes. In convolutional 2*D* (Conv2D) networks, the *Conv*2*D* layers are a stacked array of cross-correlation filters that act on multiple channels and sum the results (Rajpurkar et al., 2017). The mandatory Conv2D parameter is the number of filters that convolutional layers will learn from. It is an integer value and determines the number of output filters in the convolution. Conv2D (filters = 16, kernel \_size = (3*,*3), activation = relu , padding = same

) Arguments:

Filters: The integer represents the number of filters the convolutional layer should learn.

Kernel size: represents the number of pixels in height and width, i.e., the two-dimensional width and height of the filter. It can be a single integer that specifies the same value for all spatial dimensions. This parameter determines the dimensions of the kernel. Typical dimensions include 11*,*

33, 55, and 77, which can be represented as (1, 1), (3, 3), (5, 5), or (7, 7)

tuples. This parameter must be an odd integer.

Activation function: it is used to which the linear output of the *Conv*2*D* layer is fed to make it nonlinear, whereas ReLU is a pioneer and the most popular nonlinear activation function in deep CNN. It is a simple yet highly effective segmented function that forces negative-valued input features to zero, retaining only the non-negative features. upGrad (2022). Padding: one of valid or same; valid means no padding. The same results can be achieved by padding the input with zeros on its left and right edges or top and bottom edges evenly.

MaxPool2D: MaxPooling2D class. Max pooling operation for 2*D* spatial data. The samples the input along its spatial dimensions (height and width) by taking the maximum value over an input window (of size de ned by pool\_size) for each input channel. Max pooling is a sample-based discretization process. The objective is to down-sample an input representation (image, hidden-layer output matrix, etc.), reducing its dimensionality and allowing for assumptions to be made about the features contained in the sub-regions binned by MaxPool2D (pool\_size = (2, 2)). Arguments: the pool size can be an integer or a tuple of two integers, specifying a window size over which to calculate the maximum. (2*,*2) will take the Max value over a 2*x*2 pooling window. If only one integer is specified, the same window length will be used for both dimensions.

**Block2** *Block*2 is the second convolution layer containing SeparableConv2D, Batch Normalization, and MaxPool2D.

*SeparableConv*2*D*: Separable convolutions first perform a depthwise spatial convolution (which acts on each input channel separately), followed by a pointwise convolution that mixes the resulting output channels. Intuitively, separable convolutions can be understood as factorizing a convolution kernel into two smaller kernels or as an extreme version of an inception block. The SeparableConv2D is a variation of the traditional

convolution proposed to compute it faster. It performs a depthwise spatial convolution followed by a pointwise convolution, which mixes the resulting output channels. The main difference between standard convolution and separable convolution is that, in standard convolution, the images are transformed 256 times.

Furthermore, every transformation uses up 5*x*5*x*3*x*8*x*8 = 4800 multiplications. In the separable convolution, the image is transformed once in the depthwise convolution. Then, the transformed image was passed through and elongated to 256 channels, eliminating the need to repeatedly convert the image, which can save computational power. *SeparableConv*2*D* ( lters = 32, kernel \_size = (3*,*3), activation = relu , padding = same )

Arguments: Filters: Integer means the dimensionality of the output space (i.e., the number of output filters in the convolution). Kernel size: An integer or tuple/list of 2 integers specifying the height and width of the 2D convolution window. It can be a single integer that specifies the same value for activation of all spatial dimensions. If nothing is selected, no activation is applied.

Padding: one of valid or same (case-insensitive). Valid means no padding. The same results are achieved by padding the input with zeros evenly on the left, right, top, or bottom, ensuring the output has the same height and width dimensions as the input. Batch Normalisation: A technique for training deep neural networks that standardizes the inputs to a layer for each mini- batch. This procedure stabilizes the learning process and dramatically reduces the number of training epochs required to train deep networks. Batch normalization is a layer that allows every layer of the network to learn more independently. It is used to normalize the output of the previous layers. The layer is added to the sequential model to standardize the input or output. It can be used at several points between the model's layers. *MaxPool*2*D*: As mentioned before in *Block* 1

**Block3** *Block*3 is the third convolution layer containing SeparableConv2D, Batch Normalization, and MaxPool2D.

*SeparableConv*2*D*: As mentioned before in *Block*2

Batch Normalisation: As mentioned before in *Block* 2

*MaxPool*2*D*: As mentioned before in *Block*1 and *Block*2

**Block4** This block is the fourth convolution layer that contains

SeparableConv2D, Batch Normalisation, MaxPool2D, and Dropout

*SeparableConv*2*D*: As mentioned before in *Block*2

Batch Normalisation: As mentioned before in *Block* 2

*MaxPool*2*D*: As mentioned before in *Block*1 and *Block*2

Dropout: Dropout is a technique used to prevent a model from overfitting. It can be implemented on any or all hidden layers in the network, as well as the visible or input layer. It is not used on the output layer. The term dropout refers to the process of dropping out units; it can be applied after convolutional layers (e.g., Conv2D) and after pooling layers (e.g., MaxPooling2D), thereby improving model performance. Dropout(rate =

0*.*2).

Arguments: the rate of oats between 0 and 1, a fraction of the input units to drop.

**Block 5**: This block comprises the fifth convolutional layer, which contains a SeparableConv2D, Batch Normalization, and MaxPool2D layer, as well as a Dropout and Flatten layer.

*SeparableConv*2*D*: As mentioned before in *Block*2

Batch Normalisation: As mentioned before in *Block* 2

*MaxPool*2*D*: As mentioned before in *Block*1 and *Block*2

Dropout: As mentioned before in *Block* 4

Flatten layer: The flattening operation converts the data into a one- dimensional array for input to the next layer. We flatten the output of the convolutional layers to create a single long feature vector. Moreover, it is connected to the LSTM layer for the final classification model, a fully connected layer. Flatten in Python is a function that returns a copy of the array collapsed into one dimension.

**4.2 Long Short-Term Memory (LSTM)**

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed for sequential data, excelling at learning order dependence, classification, processing, and prediction in time series. A core feature of LSTMs is their memory cell, which is capable of retaining information over extended periods of time. This long-term memory function is regulated by a set of gates: input, output, and forget.

While traditional RNNs process sequential data through interconnected neurons, LSTMs offer enhanced control over information flow and output management (Birch et al., 2020). The fundamental principle of LSTM architecture lies in its memory cell and gates (input, output, and forget), which maintain the network's state over time. These nonlinear gating units precisely regulate the flow of information into and out of the cell. Similar to RNNs, LSTMs propagate data forward, with internal cell operations enabling the retention or discarding of information.

**4.3 Softmax**

The softmax function is a function that turns a vector of K actual values into a vector of K actual values that sum to 1. The input values can be positive, negative, zero, or greater than one, but the softmax transforms them into values between 0 and 1; therefore, they can be interpreted as probabilities. If one of the inputs is small or negative, the softmax turns it into a slight possibility. If an input is significant, it becomes a large probability, but it will always remain between 0 and 1. Many multilayer neural networks terminate in a penultimate layer that outputs real-valued scores, which are not conveniently scaled and may be challenging to work with. Here, the softmax is particularly useful because it converts the scores into a normalized probability distribution, which can be displayed to the user or used as input by other systems. For this reason, it is usual to append a softmax function as the final layer of the neural network. Figure 3 shows an example of the softmax calculation in a neural network.

The softmax Formula is presented in Eq. 1, and the softmax formula symbols are explained in Table 1.

(1)

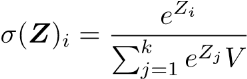


Table 1. Softmax Formula Symbols Explained.



|  |  |
| --- | --- |
| Symbol | Meaning |
| *Z* | The input vector to the softmax function is made up of (*Z*0, ..., *ZK*) |
| *Zi* | The zi values serve as the input vector to the softmax function and can take any real value. |
| *eZj* | Each element of the input vector could be implemented as a standard exponential function. This process produces a positive value, which could be very small if the input is negative and very large if the input is large in magnitude. |
|  | The term at the bottom of the Formula is the normalization term. It ensures that all function output values sum to 1 and each is in the range (0, 1), thus constituting a valid probability distribution. |
| *k* | The number of classes in the multi-class classifier. |

**4.4 Backpropagation**

Backpropagation algorithms are a set of methods used to efficiently train artificial neural networks following a gradient descent approach, which exploits the

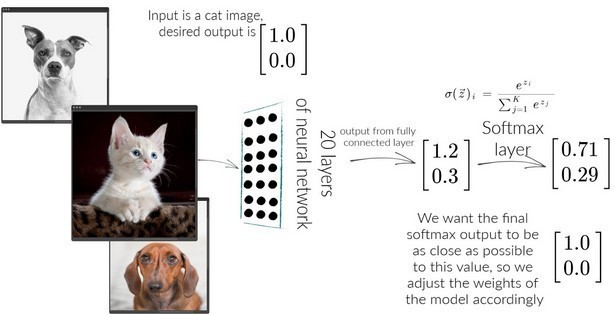


Fig.3. Example Calculation of Softmax in a Neural Network

Chain rule learning and discovering intricate structures in large datasets by using the backpropagation algorithm to indicate how a machine should adjust its internal parameters, which are used to compute the representation in each layer from the representation in the previous layer (LeCun et al., 2015). The backpropagation procedure for calculating the gradient of an objective function concerning the weights of a multilayer stack of modules is a practical application of the chain rule for derivatives. The critical insight is that the derivative (or gradient) of an objective function for the input of a module can be computed by working backward from the gradient relating to the output of that module (or the input of the subsequent module). The backpropagation equation can be applied.

**4.5 VGG**

VGG Transfer: VGG is a pre-trained neural network that utilizes the ImageNet dataset to classify raw images. The most familiar VGG model is VGG-16, which was proposed by Simonyan et al. in 2014 as a deep learning model. The model has 16 convolution layers, and the input size for this model is fixed at 224 x

224. Images are passed through a stack of convolutional layers (transforms)

with small (3 x 3) receive filters, 144 million parameters, and a maximum of five pooling layers (2 x size). 2) The three layers are fully connected. The final layer employs the softmax activation function (Ayan and Unver, 2019).

**4.6 ModelCheckpoint**

Iterations are often required when training requires a significant amount of time to achieve a satisfactory result. In this case, saving a copy of the best- performing model is beneficial only when an epoch that improves the metrics

ends. The ModelCheckpoint function in Keras was used to save the best model based on validation accuracy.

**4.7 Evaluation Metric**

There are several methods for evaluating the performance of a binary classifier. The performance of the classification algorithm for the proposed model was assessed using the accuracy performance scale, one of the simplest and most widely used performance measures. Eq. 2 provides the Formula for accuracy.

*Accuracy* = (*TP* + *TN*)*/*(*TP* + *FN* + *TN* + *FP*) (2) Equation 2 parameters stand for:

TP: Number of positive samples classified as positive (true positive). FN: Number of positive samples classified as negative (false negative). FP: Number of negative samples classified as positive (false positive). TN: Number of negative samples classified as negative (true negative).

**4.8 Log loss (cross-entropy loss)**

The most common loss function is the entropy loss, where the log loss function is frequently used in classification problems, and the loss formula is given by Eq.3

−(*y* log(*p*) + (1 − *y*)log(1 − *p*)) (3)

**4.9 Training the Proposed Model**

Once the model is built and ready for training. The training data is divided into batches for each part separately. It is fed into the neural network for training. One repetition on each training set is referred to as an epoch. After each period, a validation process is carried out to ensure the neural network gives a better result and stops at the lowest error rate. The value accuracy increases, meaning the built model learns and works well. It has an accuracy rate of approximately 99*.*73% and a loss rate of roughly 0*.*008%. However, there is a decrease in the loss and an increase in the validation loss.

**4.10 Validation of the Model**

The validation process is essential for checking the model's performance on unseen images. The validation process shows that the model has an accuracy rate of approximately 97*.*49% and a loss rate of roughly 0*.*23%. Figures 4 and

5 represent the validation results.

**4.11 Testing the Model**

The dataset is organized into two volumes (cancerous and non-cancerous). It contains 49 images of chest X-rays (PNG). The X-ray images have been uploaded to Google Drive to share with the Azure cloud programming language,

which will be used to implement algorithms and a convolutional neural network during the testing phase. This process differs from the images used in the original training dataset. These images are classified into two categories, with and without lung cancer, distributed as in Table 2.

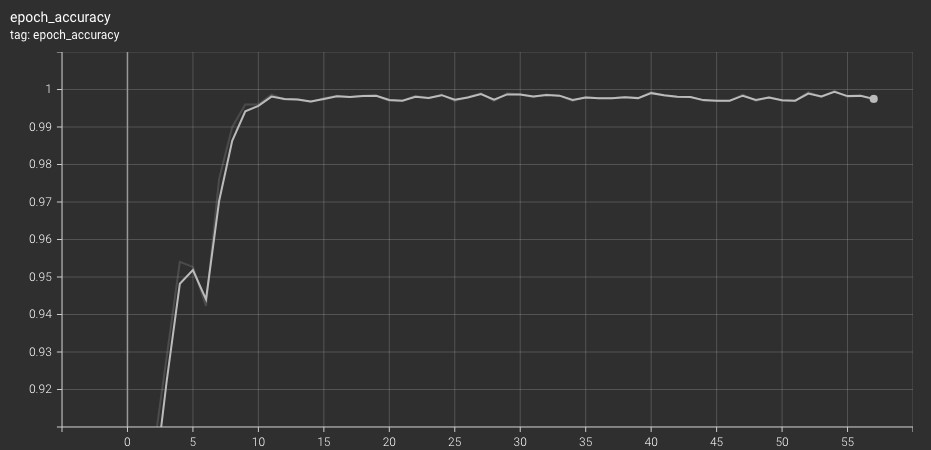


Fig.4. Plot of initial validation, 38 epochs, accuracy.

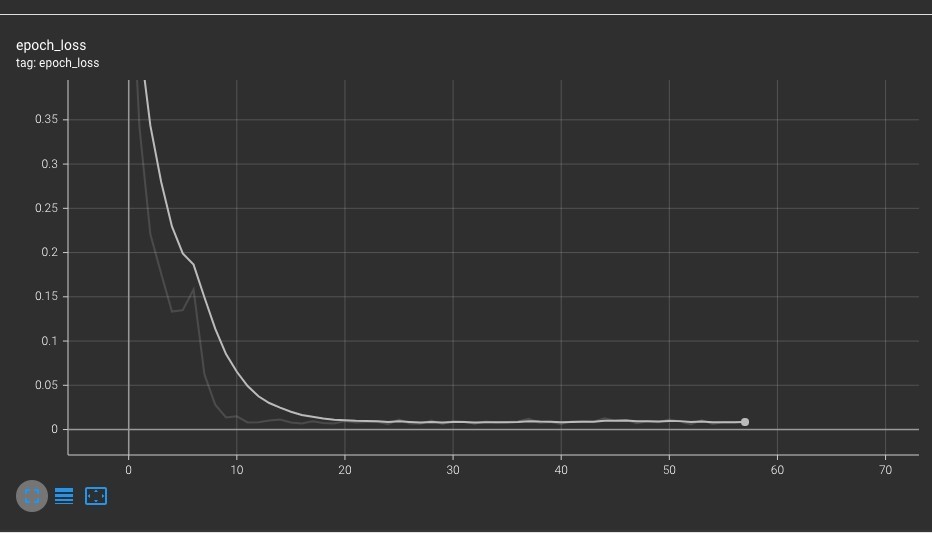


Fig.5. Plot of initial validation, 38 epochs, loss.

Table 2. Samples of the chest X-ray images were used for testing the networks.

Categories Number images

of testing Image size

|  |  |  |
| --- | --- | --- |
| Cancer | 11 | 480*x*480*pixels* |
| Non-Cancer | 38 | 480*x*480*pixels* |

5 **Results Discussion**

The results presented in this research demonstrate that the proposed model is implemented based on a deep learning approach for diagnosing chest X-ray images by classifying them into images with and without lung cancer. The process begins by processing the dataset, and then a neural network is built to extract features from the images using VGG16. The features are converted into

a vector of size 1 ∗ × 57600 and passed to the LSTM network. Once the result

has been classified, a softmax function is applied to the output of the LSTM

network. As mentioned in this chapter, the neural network was trained and validated using the Python programming language in Azure Cloud ML Studio, with approximately 38 epochs trained over 12 batch sizes and a learning rate of 0.0001.

To show the results of tracking and monitoring the training model status, Matplotlib is used to present the accuracy and loss of the training, as illustrated in Figures 6 and 7.



Fig.6. Plot of initial training, 38 epochs, accuracy.

Table 3 presents the final results of the proposed model using a dataset comprising 247 images trained for 20 epochs.

Table 3. Comparison of the Model used in the training, validation, and testing results.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Training Accu- racy | Training loss | Validation Ac- curacy | Validation loss | The testing model classification rate |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 99*.*73 | 0*.*008 | 97*.*49 | 0*.*23 | 90*.*99*pixels* |

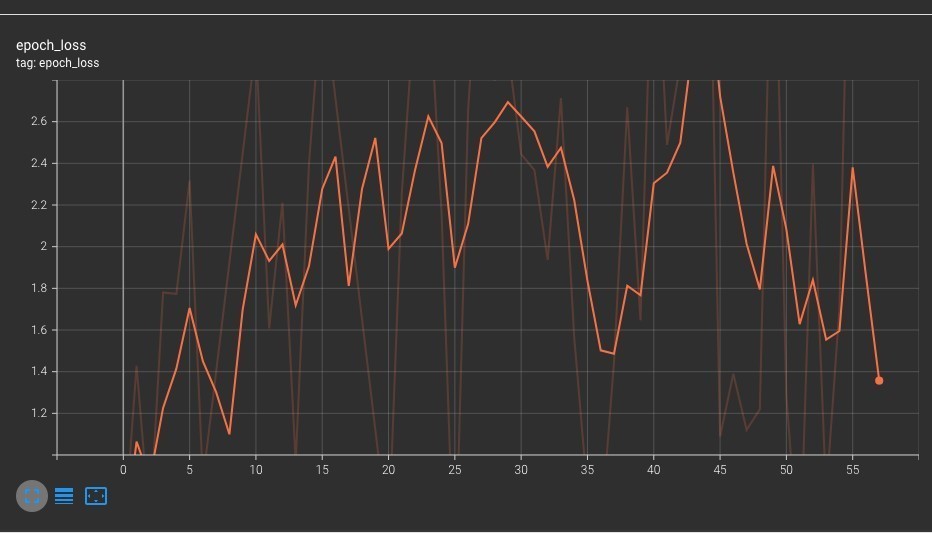


Fig.7. Plot of initial training, 38 epochs, loss.

Moreover, the proposed model is trained using 149 chest radiographs, comprising 34 cancerous and 115 normal cases, with an accuracy rate of 99.73%. The proposed work was validated using 49 chest radiographs, of which 11 were cancerous and 38 were normal, yielding an accuracy rate of 97.49%.

The results of the proposed model were tested using 49 chest radiographs, 11 of which were cancerous, and 38 were typical, with a predetermined classification according to the medical report of each patient. The accuracy rate was 90.99%.

6 **Conclusions**

This paper aims to analyze the identification and prediction of lung cancer using a hybrid system that combines Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) to extract information about lung cancer from chest X-ray images. On 249 chest radiographs, 56 had lung cancer, and 191 did not. This model is written in Python and is included in the Microsoft Azure cloud platform, which features a high-level language with an easy and accessible user interface. The superior performance is mainly due to the deep structure of CNN models such as *V GG*16 networks, which have higher generalization capabilities and accuracy compared to other networks in terms of extracting different level features, as it produces a vector with all the characteristic properties linked to LSTM that holds data in the form of weights during training to do Classi cation.

A training accuracy of 99*.*73%, a verification accuracy of 97*.*49%, and a test accuracy of 90*.*99% were obtained. It has been observed that the precision and accuracy of training and validation are increasing, resulting in more accurate outcomes that enable the detection of lung cancer. The results showed that the proposed model achieved high recognition rates, yielding very precise results in diagnosing chest X-ray images of the lungs and then classifying them into two types: lung cancerous and normal. These results indicate that deep Learning methods can be much better than other high-performance algorithms. Because deep learning methods have wide applications in the medical field,

medical diagnosis is made through the use of deep learning networks, including detection, Segmentation, classification, prediction, and others.

Further investigation could be conducted to examine the implementation of the approaches presented in this thesis using different datasets containing the five stages of lung cancer. This study will serve as a foundation for future research on the implementation of other convolutional network techniques, such as VGG19, AlexNet, and GoogleNet. Building a social user interface system to train and test these techniques online would be interesting, as it could be applied to lung diagnosis in chest X-ray images, serving the most significant number of slums that lack healthcare and do not have a diagnosis price.

Disclaimer (Artificial intelligence)

Option 1:

Authors hereby declares that NO generative AI technologies such as Large Language Models (ChatGPT, COPILOT, etc.) and text-to-image generators have been used during the writing or editing of this manuscript.

**References**

1. Thandra, K. C., Barsouk, A., Saginala, K., Aluru, J. S., & Barsouk, A. (2021). Epidemiology of lung cancer. *Contemporary Oncology/Współczesna Onkologia*, *25*(1), 45-52.

2. Shimazaki, A., Ueda, D., Choppin, A., Yamamoto, A., Honjo, T.,

Shimahara, Y., & Miki, Y. (2022). Deep learning-based algorithm for lung cancer detection on chest radiographs using the segmentation method. *Scientific Reports*, *12*(1), 727.

3. Ayan, E., & Ünver, H. M. (2019, April). Diagnosis of pneumonia from chest X-ray images using deep learning. In *2019 Scientific meeting on electrical-electronics & biomedical engineering and computer science (EBBT)* (pp. 1-5). Ieee.

4. Benhammou, Y., Achchab, B., Herrera, F., & Tabik, S. (2020). BreakHis

based breast cancer automatic diagnosis using deep Learning: Taxonomy, survey and insights. *Neurocomputing*, *375*, 9-24.

5. Birch, P., Rahimi, N., Overburry, P., Young, R., & Chatwin, C. (2020,

April). Implementations and optimisations of optical Conv2D networks designs. In *Semiconductor Lasers and Laser Dynamics IX* (Vol. 11356, pp. 178-185). SPIE.

6. Chapaliuk, B., & Zaychenko, Y. (2018, October). Deep learning

approach in computer-aided detection system for lung cancer. In *2018*

*IEEE First International Conference on System Analysis & Intelligent*

*Computing (SAIC)* (pp. 1-4). IEEE.

7. Gao, R., Huo, Y., Bao, S., Tang, Y., Antic, S. L., Epstein, E. S., ... & Landman, B. A. (2019). Distanced LSTM: Time-distanced gates in long short-term memory models for lung cancer detection. In *Machine learning in medical imaging: 10th international workshop, MLMI 2019, held in conjunction with MICCAI 2019, Shenzhen, China, October 13,*

*2019, proceedings 10* (pp. 310-318). Springer International Publishing.

8. Garcia, E., Hermoza, R., Castanon, C. B., Cano, L., Castillo, M., & Castanneda, C. (2017, June). Automatic lymphocyte detection on gastric cancer IHC images using deep learning. In *2017 IEEE 30th international symposium on computer-based medical systems (CBMS)* (pp. 200-204). IEEE.

9. Gupta, P., & Malhi, A. K. (2018, July). Using deep learning to enhance head and neck cancer diagnosis and classification. In *2018 ieee international conference on system, computation, automation and networking (icscan)* (pp. 1-6). IEEE.

10. Jeyaraj, P. R., & Samuel Nadar, E. R. (2019). Computer-assisted

medical image classification for early diagnosis of oral cancer employing deep learning algorithm. *Journal of cancer research and clinical oncology*, *145*, 829-837.

11. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep

Learning. *nature*, *521*(7553), 436-444.

12. Lu, W., Li, J., Li, Y., Sun, A., & Wang, J. (2020). A CNN‐LSTM‐based

model to forecast stock prices. *Complexity*, *2020*(1), 6622927.

13. Jenkins, R., Walker, J., & Roy, U. B. (2023). 2022 cancer statistics: Focus on lung cancer. *Future oncology*, 1-11.

14. Moturu, A., & Chang, A. (2018). Creation of synthetic X-rays to train a

neural network to detect lung cancer. *Journal Beyond Sciences*

*Initiative, University of Toronto, in Toronto*.

15. Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., ... & Ng, A. Y. (2017). Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*.

16. Shimizu, R., Yanagawa, S., Monde, Y., Yamagishi, H., Hamada, M.,

Shimizu, T., & Kuroda, T. (2016, October). Deep learning application trial to lung cancer diagnosis for medical sensor systems. In *2016*

*International SoC Design Conference (ISOCC)* (pp. 191-192). IEEE.

17. Silaparasetty, V. (2020). *Deep Learning Projects Using TensorFlow 2*.

Apress.

18. Tsukamoto, T., Teramoto, A., Yamada, A., Kiriyama, Y., Sakurai, E., Michiba, A., ... & Fujita, H. (2022). Comparison of fine-tuned deep convolutional neural networks for the automated classification of lung cancer cytology images with integration of additional classifiers. *Asian Pacific Journal of Cancer Prevention: APJCP*, *23*(4), 1315.

19. Ivanova, M., Pescia, C., Trapani, D., Venetis, K., Frascarelli, C., Mane,

E., ... & Fusco, N. (2024). Early breast cancer risk assessment:

integrating histopathology with artificial intelligence. *Cancers*, *16*(11),

1981.

20. Adhikari, S. K., Neupane, P., Mainali, S., Regmi, U., & Chapagain, P. (2024). American Sign Language Classification using CNNs: A Comparative Study. *International Journal on Engineering Technology*, *1*(2), 283-295.

21. Wang, Z., Xiao, Y., Li, Y., Zhang, J., Lu, F., Hou, M., & Liu, X. (2021).

Automatically discriminating and localizing COVID-19 from community- acquired pneumonia on chest X-rays. *Pattern recognition*, *110*,

107613.

22. Ahmed, T., Parvin, M. S., Haque, M. R., & Uddin, M. S. (2020). Lung cancer detection using CT image based on 3D convolutional neural network. *Journal of Computer and Communications*, *8*(03), 35.

23. Bhargavi, K., & Krishna, C. V. P. (2024, August). A Survey on Applications and Performance of Deep Convolution Neural Network Architecture for Image Segmentation. In *2024 International Conference*

*on Emerging Techniques in Computational Intelligence (ICETCI)* (pp.

408-413). IEEE.