Online-Journals.org

iJOE International Journal of Online and Biomedical Engineering

iJOE | eISSN: 2626-8493 | Vol. 19 No. 13 (2023) | OPEN ACCESS

https://doi.org/10.3991/ijoe.v19i13.40523

PAPER

Multiple Disease Detection using Machine Learning Techniques

Dipanjan Acharya[1], K. Eashwer[1], Soumya Kumar[1], R. Sivakumar[1], P.C. Kishoreraja[1](✉), Ramasamy Srinivasagan[2]

[1]Vellore Institute of Technology, Vellore, India

[2]Faculty of Computer Engineering, King Faisal University, Hofuf, Saudi Arabia

kishoreraja.pc@vit.ac.in

ABSTRACT

The COVID-19 disease outbreak resulted in a worldwide pandemic. Currently, the reverse transcription-polymerase chain reaction (RT-PCR), which relies on nasopharyngeal swabs to examine the existence of the ribonucleic acid (RNA) of SARS-CoV-27, is still a popular approach to testing for the disease. Despite the high level of specificity of testing with RT-PCR, the sensitivity of the method could be relatively low, and there is significant variability in efficacy depending on different sampling methods and the time of occurrence of symptoms. It is therefore essential for us to develop a machine-learning algorithm that can analyze computerized tomography images to detect the presence of COVID-19. Besides COVID-19, lung computerized tomography (CT) scan images can detect many other diseases, such as lung cancer, pneumonia, etc. This paper deals with the implementation of an algorithm that takes lung CT scans and lung X-ray images as input and predicts a list of probable diseases and possible diagnoses that infect the lungs. Machine learning algorithms will be able to predict disease by scanning the tiniest of regions easily missed by the human eye. This paper presents a survey of various machine learning algorithms that aid in detecting multiple diseases in lung CT scan images. Apart from the study of standard algorithms best suited for COVID-19 detection, this paper also includes recent trends. One of the major recent trends that can be incorporated into COVID-19 detection is TinyML. Tiny ML is an emerging area in machine learning algorithms that can be used to detect multiple diseases in lung CT scan images with better accuracy and in less time. This tool can aid doctors in their diagnosis and treatment of patients and help increase the efficiency of the treatment process. While understanding the features and mapping them using a hidden layer, there is a probability of compressing the dataset, as well as the model to process and classify the low-bit images in real-time using TinyML.

KEYWORDS

convolutional neural network (CNN), KNN, KNN classifier, VGG-16, MobileNet, InceptionNet, Alex Net, TinyML

Acharya, D., Eashwer, K., Kumar, S., Sivakumar, R., Kishoreraja, P.C., Srinivasagan, R. (2023). Multiple Disease Detection Using Machine Learning Techniques. *International Journal of Online and Biomedical Engineering (iJOE)*, 19(13), pp. 120–137. https://doi.org/10.3991/ijoe.v19i13.40523

Article submitted 2023-04-16. Revision uploaded 2023-06-11. Final acceptance 2023-07-01.

© 2023 by the authors of this article. Published under CC-BY.

1 INTRODUCTION

Today's world is being ravished by the numerous diseases that are hindering the day-to-day lives of people. There are numerous diseases that are often mistaken due to the surface-level symptoms that people are aware of, and thus they mistake them for different diseases and are often led by this premonition, while this is the case in mostly rural areas where the medical facilities are not enough. The hospitals in urban areas also have adequate amounts of medical facilities, but those are not economically feasible and they are also geographically inaccessible to the people in the rural areas. Therefore, the hospitals are the frontrunners of the initiative, where the healthcare department has put emphasis on telecommunication and telemedicine, so as to provide access to the patients, as well as to provide reports, results, and initiatives of the medical department in the digital bulletin of cloud. This will help in bridging the gap. In a populous country such as India, there is always a higher density of patients in hospitals that have adequate facilities. Therefore, managing the patients' records seems cumbersome and time-consuming. Over the course of time, machine learning algorithms have been used to analyze and detect the diseases that plague us today [1–3]. The concept of deep learning (DL) can also be introduced in identifying the patterns at a more comprehensible, or rather, more elemental level, of each and every section of data, possibly due to the complex network structure of the network structure using different sources of dataset. In this review paper, we take a look at the various machine learning algorithms that have been used to accomplish this objective. The paper proposes to find the disease using TinyML which uses a low-bit precision method and a time-compatible process. This paper focuses on the analysis of computerized tomography (CT) images using machine learning algorithms and their effectiveness. Emerging fields such as TinyML have also garnered a lot of interest and study.

1.1 Related works

The reverse transcription-polymerase chain reaction (RT-PCR) test relies on nasopharyngeal swabs to examine the existence of the ribonucleic acid (RNA) of SARS-CoV-27 to test the disease. The problem with this test is that sensitivity and efficacy differ with time. It is necessary to develop a system that detects COVID-19 with the help of CT images. Besides COVID-19, lung CT scan images can detect many other diseases, such as lung cancer, pneumonia, etc. This paper deals with the implementation of an algorithm that takes lung CT scans and lung X-ray images as input [8–11] and predicts a list of probable diseases and possible diagnoses that infect the lungs. Machine learning algorithms will be able to predict disease by scanning the tiniest of regions easily missed by the human eye.

One of the major recent trends that can be incorporated into COVID-19 detection is TinyML. Tiny ML is an emerging area in machine learning algorithms that can be used to detect multiple diseases in lung CT scan images with better accuracy and in less time. This tool can aid doctors in their diagnosis and treatment of patients and help increase the efficiency of the treatment process. While understanding the features and mapping them using a hidden layer, there is a probability of compressing the dataset, as well as the model to process and classify the low-bit images in real-time using TinyML.

Convolutional neural network. A convolutional neural network (CNN), or artificial neural network (ANN), is a machine learning tool that acts as a classifier and is used to train architecture models. It provides a quicker identification process for similar datasets and gives results in real-time [4–5]. The different stages of CNN are given in Figure 1. It works well on larger datasets that have handwritten machine input as well as image recognition. On the other hand, in the medical examination process of CT scan images, the background data can also be omitted, and the target image of the examination can also be identified by the red green blue (RGB) layers for classification. The contrast of images can be highlighted by segmenting out background data using deep CNN at preliminary levels. The resultant images that are reduced at dimensional levels can be evaluated by identifying the correlation, covariance, standard deviation, and so on. The cumulative variance in the images of wounds with a trained dataset is obtained for identification and classification based on the skin proportions and identification of overlapping images to avoid overfitting. Image segmentation is carried out, and an evaluation index is used for classification parameters for sensitivity, precision, accuracy, and data loss.

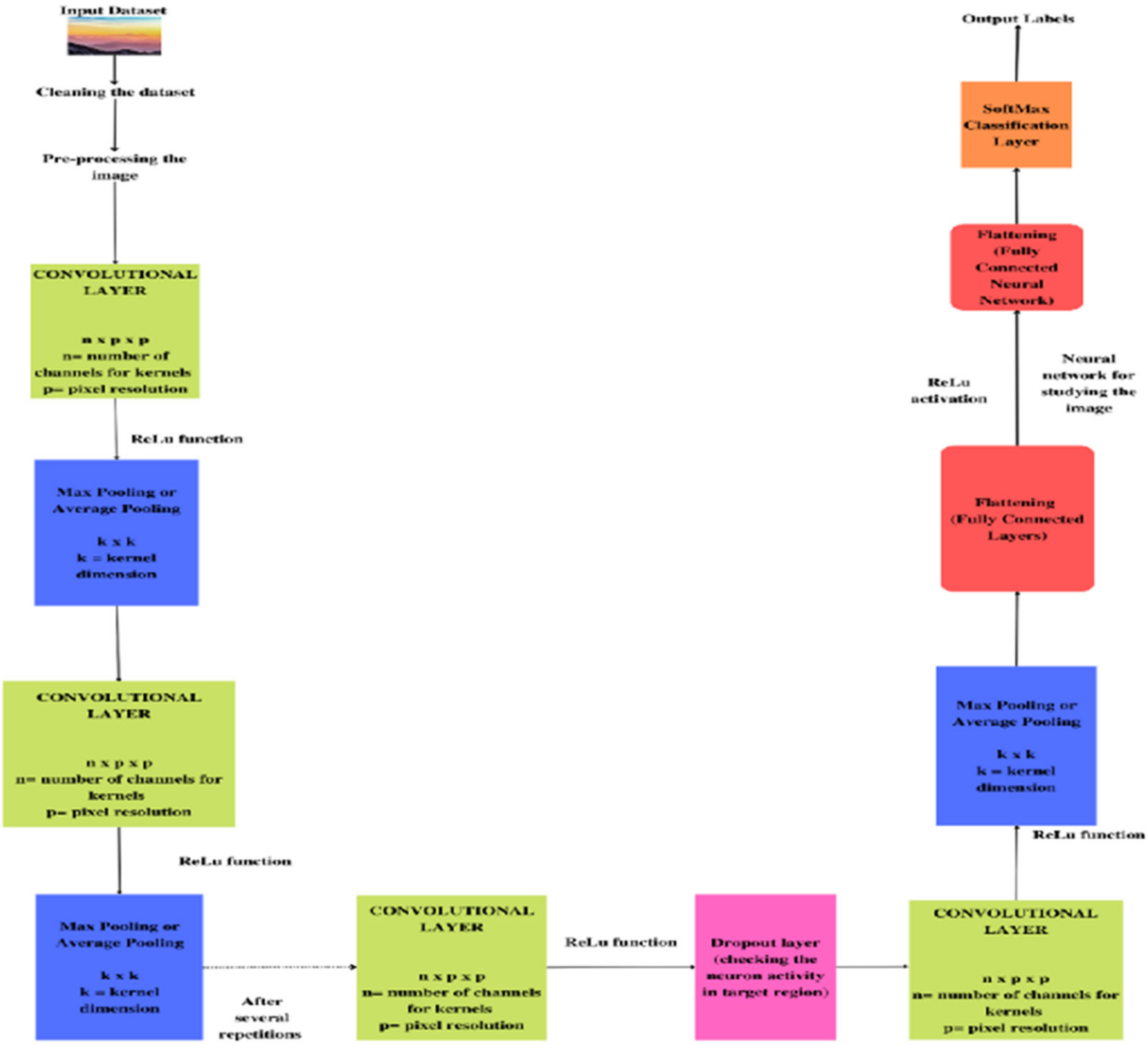

Fig. 1. Convolutional neural network

The conversion of one-dimensional arrays takes place with the identification of the concatenated models that are being used to provide better data segmentation and augmentation to enable the efficiency of the model while dealing with the CT scan images [6–7]. These elements comprise kernels, pooling, padding, flattening, and finally classification. The CNN architecture is used for better feature extraction [8] by dimensionality reduction using likely filter kernels for filtering the data to subsample them into successive layers of transformation, where the learning process is also being undertaken by the convolutional layers.

The kernel size, the striding points (i.e.), the number of pixel distances the kernel moves in the convolution process, the padding dimension, and the kernel dimensions being used for the very same determine the dimensionality reduction process, where the same can be represented by the [8–10] following mathematical formula:

$$\frac{z - f + 2 * p}{s} + \tag{1}$$

(z = dimension of the input image, f = dimension of the kernel; p = dimension of the padding used in the input image, s = striding length of the kernel)

The one thing that is significant about the CNN network is that it is known for learning a large amount of data in image recognition without spatially altering the input image. This is possible by using the rectified linear unit for obtaining the feature maps for activation, the sigmoid, or the tanh function for the same [9–10]. This is a cumbersome and time-consuming process when working on non-identifiable or unlabeled data. Thereby, the learning of the data required to reduce the dimensionality to a 1-dimensional array for studying the feature maps is achieved by the interconnection of the nodes.

The flattening process refers to the breakdown of the 3-dimensional data of the image into a 1-dimensional feature vector for studying, as well as further breakdown to reduce neuron activation to reduce the load of features. This is to reduce the number of neurons to be learned in the fully connected layer, where a sufficient number of neurons are left to learn. This is followed by the SoftMax layer function, which plays a significant role in the classification process based on the activation and non-linear feature map activation points that are used for learning. The classifiers usually assigned for the labeling of the data are the SoftMax function, the SVM classifier, and so on [10–12]. On the other hand, stochastic gradient algorithms are used for optimal parameters for likelihood calculation and entropy, where contrast divergence is used for approximate values. The primary usage of classifiers is the SoftMax layer in CNN, which uses the results obtained from the neurons of the fully connected layers using the relative exponential value, which we can mathematically represent in this [10–11] formula, where n is the number of labels.

$$\frac{e^i}{\sum_i^n e^i} \tag{2}$$

There are multiple classifiers that are compatible with CNN architecture in supervised learning. However, the dataset chosen, the kernel depth, convolutional layers, and activation layers that are being used for the feature extraction are as equally essential as the classifier. Some of the classifiers that are used with CNN are discussed in the following subsections.

SVM classifiers. A support vector machine (SVM) algorithm is used for the classification of the input data into binary-encoded labels that are being tested and pre-trained in the model. The CNN architecture helps in prior feature extraction in the form of 2D kernels with successive depth significance and for regression analysis. ANN in individual classification using the gray level co-occurrences matrix is used to examine the covariance and standard deviation from the pattern trained with the dataset, where the contrast in the data is identified. Contrast identification is also possible using active contouring with the gray levels of the wound on the skin using the snake model. The SVM classifier takes into account all of the subspaces and helps in image classification, which is based on feature vectors [12–13]. The Eigenvalues of the closest point of the positive case mark the plotting of data points or vectors henceforth away from the hyperplane or line away from the other label, and the negative point acts as the same as the above but in a mutually opposite direction, as illustrated in Figure 2.

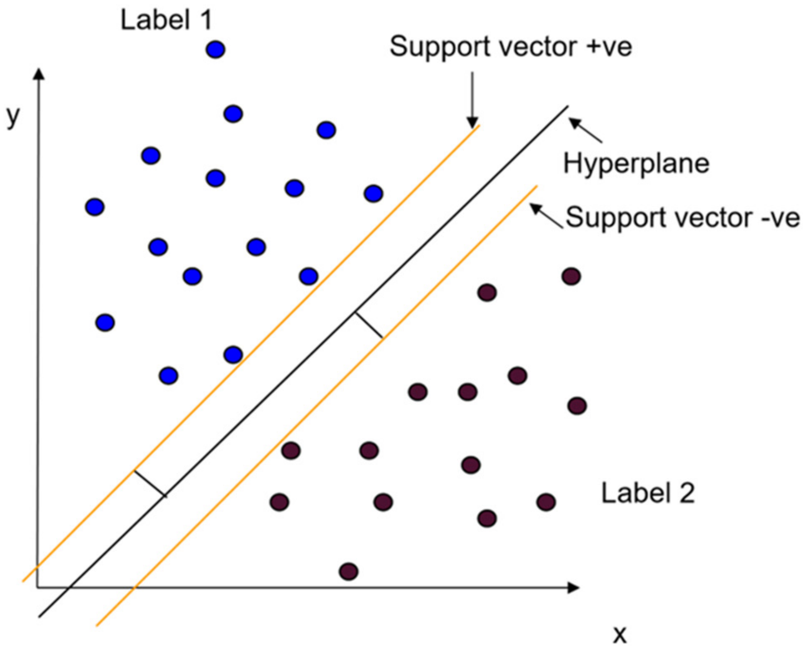

Fig. 2. Representation of binarized classification done by SVM classifier

Pixelwise image features are extracted using computer vision language using SVM and DL classifications. Machine learning algorithms for mapping the input with the expected output into a fully connected layer are done by SVM and CNN. The testing of the unlabeled image dataset is done for an unbiased evaluation of the result. SVM and DL classifiers are utilized for unlabeled images and prediction to process certain unclassified images using the Spark framework. The spark framework's main function is to convert visual images into feature vectors using prediction and querying for classification using ML libraries [14]. An instance where the comparison was done with the SoftMax layer function of the CNN for classification as well as using the SVM classifier showed that the 30 CT scan images used for the testing of the model had 94% accuracy, while 86% accuracy was seen in the usage of the SVM classifier. The usage of the X-ray images for the classification using the SVM classifier, where the linear version and radial basis kernel function are used, provides around 94.9% and 97.8% accuracy, as well as 85.74% and 94.43% precision in the classification of lung disease detection [15].

Decision tree classifier. The decision tree-based supervised machine learning classifier is used for identifying the labeled classes in the format of the decision tree.

A similar type of decision tree classifier is J4.8, where the defects in the image can also be detected. Where the repetitive peaks in the image of the data are monitored and the defects in the image are identified using the standard deviation, pixel count of the image, and size of the image, this classifier is usually combined with the Naive Bayes classifier to calculate the combination of frequencies of different image paradigms in supervised classifiers [19]. When the decision tree classifiers are being used to study the chest x-ray images, the bifurcation of the input data into normal and abnormal had an identification rate of 98%, while the parallel study of the abnormal images was studied to detect any trace of tuberculosis and had around 80% accuracy, while the COVID detection rat in the chest x-ray images was around 95% accurate [16–18]. This classifier is used for supervised machine learning, and due to its branched-out tendency to learn the data, the loss function is considerably prevalent, and this can only provide about 82% accuracy in the utility of MLP models along with ResNet.

1.2 Binary cross entropy

Binary cross entropy is the method that is used for COVID detection. The basic methodology of binary cross entropy is to determine the probability of an image being part of one of two classes.

A loss function is used to predict the probability of a point belonging to one of the classified groups. This function can also be used to keep track of the loss of data in a given model, where this function binaries' the input data only in the case of the pre-trained model in the presence of supervised learning, where it best performs in the detection of black lung disorder in the presence of the CheXNet model of neural network and the sigmoid function as the classifier [20–22]. A comparison of different classifiers with different parameters is given in Table 1.

Table 1. Comparison of different classifiers with different parameters

Classifier	Time Taken	Size of Dataset	Accuracy	Datatype
SVM	Least due to simple and discretely distant boundaries	Not preferable due to higher time complexity	Highest accuracy if the size of data is small, simple and binarized labels	Works efficient for binary and non-binary categorical conditions for discrete data
KNN	More time taken due to extensive memory storage in training phase and lack of generalizing data	Performs well in the case of multi-labelled bigger trainable datasets, only second to decision tree in terms of training time	Second to SVM only if the number of features is more than the trainable dataset, else very close to SVM	Better for non-binary categorical or discrete data
Fuzzy Logic	More than SVM and decision trees but lesser than KNN	Not suitable for datasets that have larger training, imprecise and lesser features to be extracted	Individually has a lesser accuracy rate, while a hybrid with other classifier can amplify the binarization and huge training datasets	Better for standardizing continuous data in a binarized distribution
Decision Tree	Lesser than KNN and Fuzzy logic, but more than SVM as it has a complex study of the heavier dataset	Performs well in the case of larger datasets and lesser features to be extracted	Higher if the usage of continuous labels in data	Better at dealing with continuous and categorical data

Similar to the classifiers, there are multiple networks to the CNN that are being used, some of them are described in the following subsections.

1.3 Alexa net

The Alex Net is a deep architecture that is used for checking the data up to a permissible degree, and the histogram of oriented gradients with the local binary pattern can be used for texture classification and threshold establishment for image classification. The supervised classification of the image is done by administering the Alex Net Convolutional Neural Network, and the image is retrieved [21–23], where the color level correlation matrix is also considered as a parameter of contrast to identify the image label, as well as the joint probability and the pixel distance with the entropy to identify the contrast in the fields of movement, rotation, and magnitude for obtaining the feature vector. Generative adversarial networks (GAN) are employed as supervisory networks for multiple dataset training, where this stability in training is maintained by Wasserstein GAN. The MLP takes input into the fully connected layers to identify the class and provides the classification result for the input data. The class-based dermatological conditions are considered in the epochs, where the confusion matrix is obtained for the dataset for the best and worst cases. The accuracy of the data, the validation of the input dataset, and the trained dataset are compared, as well as the data loss being visualized. This is said to be advantageous in the applications of image classification for multiple lung diseases that are practiced in a supervised environment. This is possible due to the efficiency and quicker classification of the RGB image input, which helps identify images accurately and doesn't limit the input of images due to the feature extraction possible due to the 8 layers, where the average accuracy of classification of diseases is around 91.2% [22–24].

1.4 LeNet 5

A neural network model is used for the classification of data with appropriate convolution layers, fully connected layers, and subsampling layers, like the LeNet-5 architecture of feature extraction, which has 7 layers of feature extraction with 2 subsampling layers, 2 fully connected layers, and 3 convolutional layers. The application of this network architecture is considered to have accuracy of around 88% in identifying the diversity in the pulmonary nodules in patients who are affected by lung cancer. This it achieves by analyzing the CT scan images, where the input images were resized to around 53x53, so as to identify the handwritten images and natural images from the dataset CifarNet, as well as normalizing the input data after it is followed by the rectilinear unit, as done in AlexNet, which branches out as the LeNet-5 architecture, where transfer learning increases its accuracy from around 65% to 88%, which is also affecting the size of the images that are imputed to 53x53 pixels but are increased to 256 after the model is subjected to transfer learning.

1.5 MobileNet V2

This neural network is used for dimensionality reduction in the depth-wise parameters that are used in the kernels, where the number of channels incorporated can be convoluted depth-wise as well as point-wise. This model can be used in mobile and embedded systems as it is a smaller model that utilizes depth-wise convolution in the classification process. Adam optimizer can be used, where it extrapolates the data following the average of the input data in the set of values taken in the kernel at different learning rates if the diffusion convolution network is used on the ImageNet database [24] [26], while the accuracy in the prediction model of the same dataset is seen in Inception Net V3. This architecture is used for low-bit compression analysis training, where it is power-compatible, as well as a real-time efficient method for smaller dataset training and identification.

1.6 Inception V3

InceptionV3 is a neural network architecture that is pre-trained with the ImageNet dataset. The architecture has around 48 convolution layers, and a couple of CNN network nodes are being utilized to avoid the overfitting of data and to minimize the complexity of the image [28] [29]. Each and every layer is connected at the nodes of the concatenation layer. The weight bias is managed at the halfway point. In the case of memory problems encountered in certain embedded systems [30] [31], a fully connected SoftMax layer is also connected to the concatenation layer towards the cycle of processing segmentation, which is responsible for the flattening of the data as well as the hidden layer nonlinear function that is learned [32] [33]. This can be used in the case of the training model load to be smaller in size and provide better classification results at an increased rate, which is not compromised in the density of the network, where the network layers are deeper than the V1 and V2 models, which helps in reading the COVID-19 X-ray dataset [25] [27] at a higher rate and have an accuracy of around 90% and a relatively higher precision in a time-convenient learning method, where it scales to around 91%, second to INASNET, which takes relatively longer to compute the costs of the learning as well as train the model.

1.7 VGG 16 and 19

The visual geometry group (VGG) is a pre-trained CNN architecture comprising around 16–19 convolution layers. It is an extension used to increase the depth of the CNN network, using small 3x3 kernels as standard filters. Since a pre-trained version of this model from the ImageNet dataset has a dense amount of feature representation, this architecture is efficient for the transfer learning process [30], which can be used for the data augmentation, which can be used to implement transfer learning in the CNN model itself by a transfer of the knowledge to the minor dataset, which can accelerate the process of prediction by only using the final layers of fully conducted layers as well as the hidden layers of the activation layers, which can measure up to around 138 million parameters, making it a bit tedious to operate with. The efficiency of the VGG model can be seen if the utility of the ResNet architecture is being used, where the bruises are easy to identify with an accuracy of around 96%, which shows proficiency in the extraction of features from the chest X-ray images to detect pneumonia. The VGG-16 model is said to be able to detect the chest X-rays better than CT scan images, as the 16-convolution layer model provides less precision in the results compared to its 19-layered successive model. The same model had been utilized in the multiple disease classification model; the images that were trained to the model had to be resized to around 299x299 resolution, where the images were fed to the VGG-19 model, where the accuracy of the model on an average classification of diseases is around 95.61%, where the chest X-rays were extracted to ensure mostly average pooling. A comparison of different architectures with different parameters is given in Table 2.

Table 2. Different architecture with different parameters

Architecture	Time Taken	Size of Dataset	Accuracy
AlexNet	Least due to increased channels or layers, also RGB images can be used	Preferable for larger dataset with primary focus on max pooling	Higher than LeNet due to increase in classes and layers
LeNet	More time taken in case of larger images, lesser in case of GLCM, due to lesser convolutional layers	Performs well in the case of heavy dataset and usage of global average pooling	Less accurate than its successor AlexNet due to lesser classes and lack of enough dropout layers
Inception V3	Lesser than LeNet, but more than AlexNet and VGG if it has a complex study of the heavier dataset due to high computation	Performs well in the case of smaller datasets due to extensive computation required due to lesser layers	Better performance than LeNet, lesser than or close to AlexNet and MobileNet V2 due to more layers than AlexNet in a bigger dataset
VGG 16 and 19	Lesser receptive fields suitable for small dataset and is second to MobileNet	Useful in the case of heavy datasets, but is slower than Inception and handles better complexity of data in neuron layers	More accurate than AlexNet due to smaller receptive fields for better precision in classification and more than MobileNet V2 for marginal increase in computations
MobileNet V2	Least time taken in case of a small and good dataset	Not suitable for datasets that have larger training, due to lesser power consumption and latency	Better performance than AlexNet, VGG, Inception V3 in case of lighter neural network needed and lesser parameters in lighter dataset

2 LONG SHORT-TERM MEMORY

Long short-term memory (LSTM) is a branch of the recurrent neural network (RNN) that is being used to identify patterns by studying sequential data, in the case of alphanumeric and speech. The diagram representation for an RNN is given in Figure 3. This model is often considered to identify relatively complex and longer data sequences, but only at the cost of the time taken by this model. The availability of the memory gates and cells enhances its utility in comparison to the RNN due to the availability of memory cells and gates that control the significance of the information from the previous gate. The states of each gate are as follows: Forget gate: The current state and previous state results are tested by the sigmoid function, ranging from 0 to 1, to update the cell status for the following step, gauging the necessity of the information, with 0 least and 1 most important, that is multiplied by the previous timestamp cell state information in the context of point by point. The mathematical representation of the forget gate function f(t) is given by [34] [35].

$$f(t) = \text{Sigmoid}\{W(f)[h(t-1), X(t)] + b(f)\} \tag{3}$$

(t = timestamp, W(f) = weight matrix between the input gate and forget gate, h(t – 1) = previous hidden state, X(t) = current input state, b(f) = connection bias)

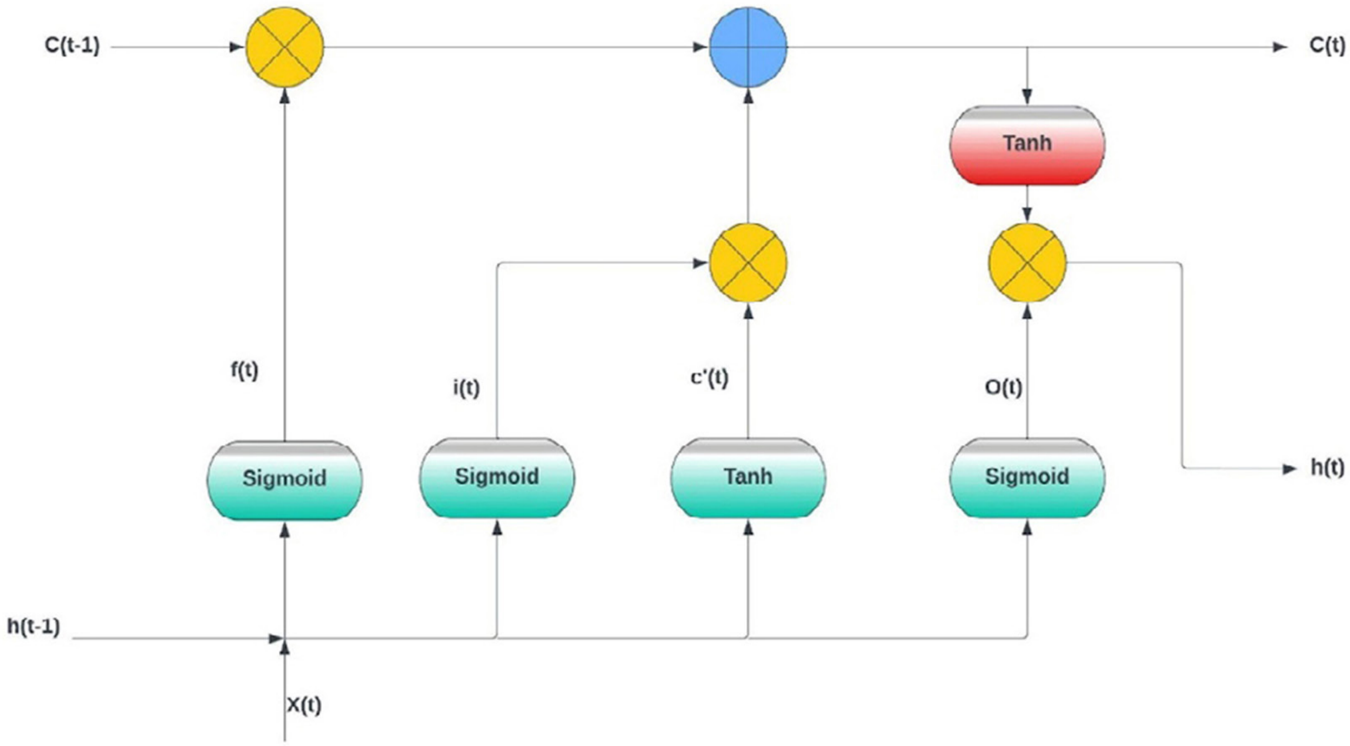

Fig. 3. Recurrent neural network

Input gate: The similar setup, along with the passage of the current state and result of the previous hidden state, is subjected to point-by-point multiplication when the current state and results of the previous hidden state form a current vector after being subjected to the TanH function that regulates the network with results ranging from –1 (least) to 1 (most), where both the results are multiplied. The mathematical representation of the input gate function i(t) is represented as:

$$f(t) = \text{Sigmoid}\{W(i)[h(t-1), X(t)] + b(i)\} \tag{4}$$

Where t = timestamp, W(i) = weight matrix between the input and output gates of the sigmoid operator, h(t – 1) = previous hidden state, X(t) = current input state, b(i) = bias vector at t. The mathematical representation of the function c'(t), generated by TanH is represented as:

$$c'(t) = \text{Tanh}\{W(c)[h(t-1),X(t)] + b(c)\} \tag{5}$$

Where t = timestamp, W(c) = weight matrix between the cell state data and network output of the Tanh operator, h(t – 1) = previous hidden state, X(t) = current input state, b(c) = bias vector with respect to W(c).

Cell state: The results of steps 1 and 2 are necessary for the cell state information, where both are added point by point, updating the cell state information for the current process. The mathematical representation of the cell state data function C(t) is represented as:

$$C(t) = f(t)*C(t-1) + i(t)*c'(t) \tag{6}$$

where t = timestamp, C(t) = Cell state data at t, C(t – 1) = previous cell state data, f(t) = forget gate at t, i(t) = input gate at t, c'(t) = value generated by TanH function at t.

Output state: A branch of the end result of step 3 is subjected to the TanH function to regulate the network, where this gate is to decide the value of the successive hidden gates. The current input and the previous hidden state data are subjected to another sigmoid function to test for the necessity of the information. The output gate is multiplied by the end result of the TanH function to provide the result for the current hidden state, which is responsible for the prediction. The mathematical representation of the output gate function O(t) is represented as:

$$O(t) = \text{sigmoid}\{W(o)[h(t-1), X(t)] + b(o)\} \tag{7}$$

where t = timestamp, W(o) = weight matrix of output gate r, h(t – 1) = previous hidden state, X(t) = current input state, b(o) = bias vector with respect to W(o). The LSTM output h(t) is represented as:

$$h(t) = o(t)*\text{Tanh}(C(t)) \tag{8}$$

It takes time to identify long patterns in the data and helping in the identification of sequential data, classify, and determine the probability distribution when using an LSTM-CNN model for enhanced feature selection in the input. This aids in identifying sequential data to train the model, Both the CNN and bidirectional LSTM networks can be used alongside with conditional random fields to expand the scope of pattern recognition in the biomedical imaging process. This process helps in selective access to the data relevant to the model, which, when tested with the JNLPBA and NCBI-disease datasets, tends to have an F1 score of around 87%, which is a potentially plausible model to increase the range for pattern recognition and be time-efficient for the very same. The CNN is responsible for the data retention, embedding, feature selection, and extraction, where the pattern recognition is identified by the bidirectional LSTM, where the results of the LSTM process from both directions are stacked onto opposite indices in the character level representation of the output, where they are concatenated. The same hybrid model is also identified as efficient in the COVID- 19 identification, with a relatively lower error rate after comparison of the performance over multiple COVID-19 cases and time series data over the impact countries such as Brazil, France, India, Mexico, Russia, Saudi Arabia, and the United States of America. This is due to the computation of the 1D data obtained in the CNN section of the model, which makes it a time-efficient model. The datasets are subjected to data normalization, followed by window slicing transformation, where the trained model identifies the modeling time dependencies, temporal feature space in the next step, and the CNN utilized for the feature extraction, while the resultant forecasting model utilizes the six different fundamental deep learning models to test for the identification process.

3 K NEAREST NEIGHBOR ALGORITHM

The K nearest neighbor algorithm is one of the most widely used and researched algorithms based on supervised learning techniques. The KNN algorithm makes the assumption that the new case and the existing cases are comparable, and it places the new instance in the category that is most like the existing categories [33].

Since KNN is a non-parametric technique, it makes no assumptions about the underlying data. It is also known as a lazy learner algorithm since it saves the training dataset rather than learning from it immediately. Instead, it uses the dataset to perform an action when classifying data. The KNN method simply saves the information during the training phase, and when it receives new data, it categorizes it into a category that is quite similar to the new data [34]. The categorization of new data points using similarity in KNN is given in Figure 4.

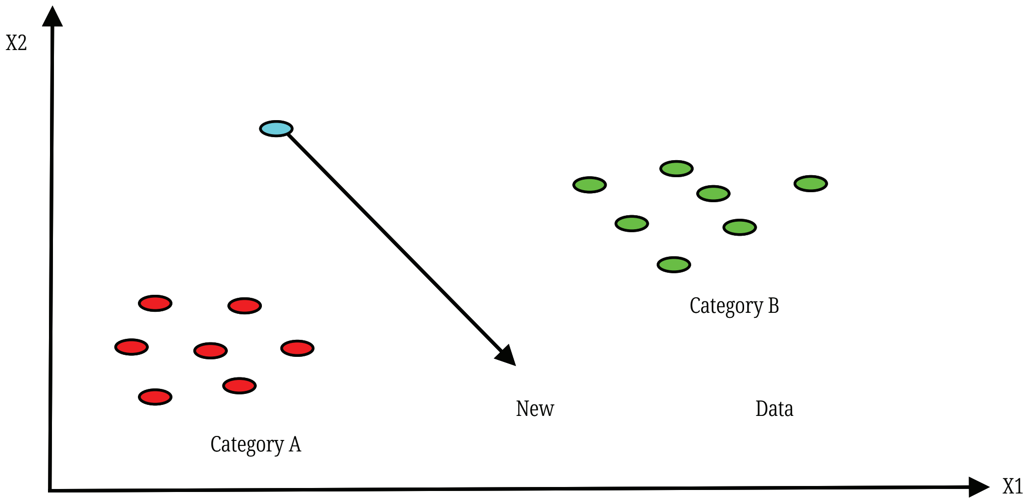

Fig. 4. Categorization of new data point using similarity in KNN

The KNN is particularly useful in problems where we need to classify a data point into one of two given categories. With the help of KNN, we can easily identify the category of a particular data entry by analyzing the similarities of the data. The algorithm employs the training data and simply remembers it throughout training. Every test image is sorted by KNN during testing by comparison to every training image and the transfer of labels from the most comparable training examples. While KNNs are no longer utilized for picture categorization, they still perform better than random guessing (around 35% accuracy). KNN doesn't iterate for parameter adjustment because it is a non-parametric method. With each test image, the distance to the closest neighbor is determined, yielding a matrix of these distance values. For the value of k, the k nearest neighbors are chosen, as well as the minimum distance rule, to determine who the closest neighbor is. If there is a tie, any rule can be used to choose the neighbor. [35]

The KNN clustering algorithms use a dynamic intelligence to detect and identify various plant diseases [36]. The k-NN algorithm has the major benefit of being very easy to use and comprehend. Working with larger datasets is computationally prohibitive since classifying a new testing point necessitates comparison to every single data point in our training data, which scales O(N). KNN is more suited for low-dimension feature spaces. It is also important to note that KNN does not really 'learn' anything in the true sense of the term. This means that this algorithm is not able to learn from its mistakes, and it merely depends on distance in the dimensional space to classify. Therefore, even though the KNN implementation is fairly simple and easy to train and use, it is not preferred for image classification and disease detection. Other variants, such as Hassan at KNN, are used for disease detection [36].

4 TINYML

The term "TinyML" describes the interdisciplinary idea of integrating machine learning capabilities into embedded devices. With this approach, users can resolve issues that require extensive amounts of time to process larger datasets. TinyML not only resolves it in real time but also has the ability to operate without the need for a computer system every single time. This is made possible by the embedded system software data being dumped into it and by giving a direct link to the acquisition device and observation monitor, which is an advanced replacement for the advantageous pathways in the telemedicine and IoT healthcare units of the hospitals [37] [38]. Once the ML model has been trained, it is compressed without affecting its accuracy. Pruning and knowledge distillation are the two major steps toward compressing. Pruning is the process of removing unnecessary data from the algorithm, which rarely affects the output. After this, the network can be trained with the new lean architecture. Knowledge distillation refers to the transfer of knowledge from a larger model to a smaller one. However, as its name implies, TinyML has been observed to experience delays in some situations when using specific neural network models that are not pre-trained and studying the photos directly. As a result, using architectures such as MobileNet V1 or V2 is recommended when developing TinyML solutions for data analytics. The diagrammatic representation of TinyML is given in Figure 5.

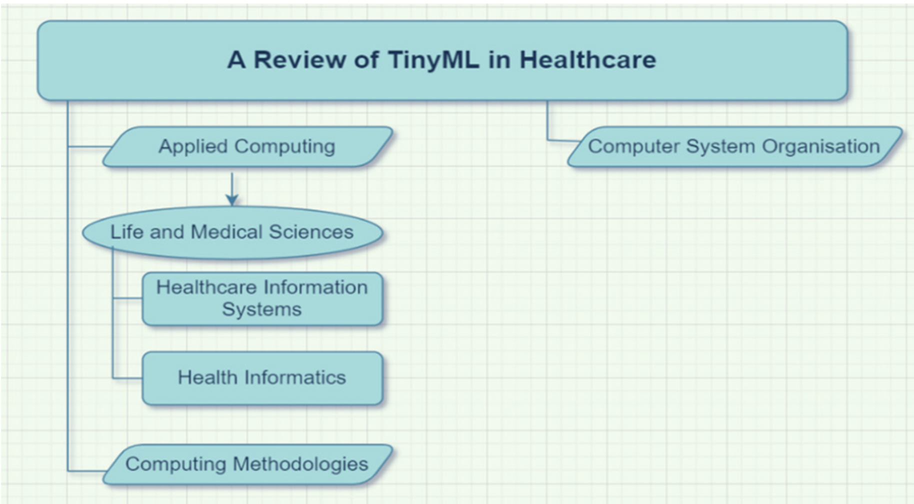

Fig. 5. Booming technology of TinyML in healthcare sector

If the neural network can be trained to identify low-bit compression and analyze the data precisely, along with having a pre-trained model to work on the same in real time, while being somewhat more laborious for heavy operations than general transfer learning and CNNs, then TinyML can be applied. The generalization modified for spreading the weight value of the nodes for improved accuracy and precision while retaining data can reduce quantization errors. Post-quantization and quantization-aware hybrid models correct for data loss, analyze learning rate using the QGT lambda hyper parameter, and keep track of training time for the chip memory limit of 100kb of data, which can give rise to new models that are efficient and provide a vast scope of edge computing in the healthcare field [39]. MobileNet V1 and V2 data sets were used to assess the accuracy and data loss of the model I developed using the ImageNet dataset. The effectiveness of the suggested model is evaluated against the baseline floating point model to converge with the post-training quantization model from 4 bit precision and an 81kb small model for customized detection up to a 2 bit result [40] [41] [42]. There are certain software's that are used for different kinds of classifications, where the C4.5 decision tree model is seemingly economical for the classification of fruits, while the features extracted were exported to a.csv file and used with the e-Cognition software and WEKA software. However, the combinations of different classifiers, as well as the usage of different architectures of neural networks, also provide unique results, where the SVM has been as efficient as the decision tree algorithms while using the C4.5 models. While using MobileNet V1, V2 or InceptionNet V3, the usage of SVM or, in condition, the usage of multi-layer perception is also being experimented with.

Some advantages of using TinyML include low power consumption, low latency, and no need for time-consuming cloud services. When compared to manual examination, the TinyML model's accuracy of over 80% and lack of Internet access make it significantly superior. In heavily impacted areas where numerous examinations are required for the same disease, such as Sub-Saharan Africa and South Asia, this can prove to be quite helpful. It is a new field of study that has sparked a lot of interest in the scientific community.

The future scope in rural areas for TinyML is checking the basic health conditions for common cold or minor inconveniences, examining vaccine dosage in cases of congenital disease, and providing details on the better healthcare service in the closest proximity. TinyML has extremely useful applications in healthcare due to it's cost-effectiveness and can definitely take the detection and diagnosis of diseases to the next level.

The other applications of TinyML have proven successful in the field of irrigation systems, where it needs to check the irrigation procedure based on soil type, land dimension, time period of the year, rotation of crops, and the crops cultivated. Also in livestock regulation, based on the morphological and genetic conditions of the farm animals, to testify to their fertility and consistency in producing competent offspring and products. Household groceries and rations are being checked for each family member in the ration shops for earlier purchase and acquisition of special government relief funds.

5 CONCLUSION

The main contribution of this review is to discuss the various machine-learning techniques employed in multiple disease detection. The paper also gives a relative comparison of all the techniques based on their applications, advantages, and limitations. After analyzing all the techniques, we cannot state that any one technique is the best, but we can surely state that TinyML has immense potential. Each technique has different application areas and is useful in different domains based on its advantages. Thus, keeping in mind the limitations of each of the techniques and the prime focus being the improvement in performance and efficiency, we should use the technique that best suits a particular application. For instance, image classification, feature extraction methods combined with classifiers suit disease detection the most. Similarly, TinyML is being developed with the added advantages of reduced delay, energy efficiency, low bandwidth, and independence from connectivity. It is a relatively new application and may be employed for multi-disease detection in the future. Our study also focuses more on CNN-based models, KNN-based models, and TinyML, which helps us visualize the entire scope of machine learning in the field of disease detection. In order to implement TinyML in a real-time project, it requires the use of the MobileNet V2 neural network and transfer learning. As TinyML is an emerging field, there is a strong future scope for better insight into the validity and generality of this technique. In particular, we plan to continue with research on how accurate TinyML machine-learning techniques are and whether they are superior by a huge margin or not. Besides that we also plan to study how select and combine a set of test cases for an effective estimation technique and to get better results.

6 REFERENCES

[1] Ziang Li, Jie Zhang, Dong Liu, and Jiangfeng Du, "CT image-guided electrical impedance tomography for medical imaging," *IEEE Transactions On Medical Imaging*, vol. 39, no. 6, pp. 1822–1832, 2020. https://doi.org/10.1109/TMI.2019.2958670

[2] A. Sivasangari, B.J.K. Reddy, A. Kiran, and P. Ajitha, "Diagnosis of Liver Disease using Machine Learning Models," *Fourth International Conference on IoT in Social, Mobile, Analytics and Cloud*, pp. 627–630, 2020. https://doi.org/10.1109/I-SMAC49090.2020.9243375

[3] A. Ampavathi and T.V. Saradhi, "Multi disease-prediction framework using hybrid deep learning: An optimal prediction model," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 24, no. 10, pp. 1146–1168, 2021. https://doi.org/10.1080/10255842.2020.1869726

[4] H. Lee and H. Kwon, "Going deeper with contextual CNN for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4843–4855, 2017. https://doi.org/10.1109/TIP.2017.2725580

[5] M.M. Mehdy, P.Y. Ng, E.F. Shair, N.I. Saleh, and C. Gomes, "Artificial neural networks in image processing for early detection of breast cancer," *Computational and Mathematical Methods in Medicine*, vol. 2017, pp. 1–15, 2017. https://doi.org/10.1155/2017/2610628

[6] B. Wang, K. Yager, D. Yu, and M. Hoai, "X-ray scattering image classification using deep learning," *IEEE Winter Conference on Applications of Computer Vision,* vol. 697–704, 2017. https://doi.org/10.1109/WACV.2017.83

[7] A. Amyar, R. Modzelewski, H. Li, and S. Ruan, "Multi-task deep learning-based CT imaging analysis for COVID-19 pneumonia: Classification and segmentation," *Computers in Biology and Medicine*, vol. 126, p. 104037, 2020. https://doi.org/10.1016/j.compbiomed.2020.104037

[8] A. Oulefki, S. Agaian, T. Trongtirakul, and A.K. Laouar, "Automatic COVID-19 lung infected region segmentation and measurement using CT-scans images," *Pattern Recognition*, vol. 114, p. 107747, 2021. https://doi.org/10.1016/j.patcog.2020.107747

[9] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017. https://doi.org/10.1145/3065386

[10] F.A. Spanhol, L.S. Oliveira, C. Petitjean, and L. Heutte, "Breast cancer histopathological image classification using convolutional neural networks," *International Joint Conference on Neural Networks (IJCNN)*, pp. 2560–2567, 2016. https://doi.org/10.1109/IJCNN.2016.7727519

[11] O.M. Al-hazaimeh, A.A. Abu-Ein, N.M. Tahat, and M.A. Al-Smadi, "Combining artificial intelligence and image processing for diagnosing diabetic retinopathy in retinal fundus images," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 18, no. 13, pp. 131–151, 2022. https://doi.org/10.3991/ijoe.v18i13.33985

[12] Y. Zhou, L. Huang, T. Zhou, and L. Shao, "Many-to-one distribution learning and k-nearest neighbor smoothing for thoracic disease identification," In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, pp. 768–776, 221. https://doi.org/10.1609/aaai.v35i1.16158

[13] L.H. Thai, T.S. Hai, and N.T. Thuy, "Image classification using support vector machine and artificial neural network," *International Journal of Information Technology and Computer Science*, vol. 4, no. 5, pp. 32–38, 2012. https://doi.org/10.5815/ijitcs.2012.05.05

[14] C. Tchito Tchapga, T.A. Mih, A. Tchagna Kouanou, T. Fozin Fonzin, P. Kuetche Fogang, B.A. Mezatio, and D. Tchiotsop, "Biomedical image classification in a big data architecture using machine learning algorithms," *Journal of Healthcare Engineering*, vol. 2021, pp. 1–11, 2021. https://doi.org/10.1155/2021/9998819

[15] P. Hattikatti, "Texture based interstitial lung disease detection using convolutional neural network," *International Conference on Big Data, IoT and Data Science (BID)*, Pune, India, pp. 18–22, 2017. https://doi.org/10.1109/BID.2017.8336567

[16] N. Dey, Y.D. Zhang, V. Rajinikanth, R. Pugalenthi, and N.S.M. Raja, "Customized VGG19 architecture for pneumonia detection in chest X-rays," *Pattern Recognition Letters*, vol. 143, pp. 67–74, 2021. https://doi.org/10.1016/j.patrec.2020.12.010

[17] G. Capizzi, G.L. Sciuto, C. Napoli, D. Połap, and M. Woźniak, "Small lung nodules detection based on fuzzy-logic and probabilistic neural network with bioinspired reinforcement learning," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 6, pp. 1178–1189, 2019. https://doi.org/10.1109/TFUZZ.2019.2952831

[18] X.W. Gao and R. Hui, "A deep learning-based approach to classification of CT brain images," *SAI Computing Conference*, pp. 28–31, 2016. https://doi.org/10.1109/SAI.2016.7555958

[19] D.A. Moses, "Deep learning applied to automatic disease detection using chest X-rays," *Journal of Medical Imaging and Radiation Oncology*, vol. 65, no. 5, pp. 498–517, 2021. https://doi.org/10.1111/1754-9485.13273

[20] Seung Hoon Yoo, Hui Geng, Tin Lok Chiu, Siu Ki Yu, Dae Chul Cho, Jin Heo, and Min Sung Choi, "Deep learning-based decision-tree classifier for COVID-19 diagnosis from chest X-ray imaging," *Front. Med.,* vol. 7, p. 27, 2020. https://doi.org/10.3389/fmed.2020.00427

[21] Nasr Gharaibeh, Ashraf Abu-Ein, Obaida M. Al-hazaimeh, Khalid M.O. Nahar, Waleed Abdel Karim Abu-Ain, and Malek Al-Nawashi, "Swin transformer-based segmentation and multi-scale feature pyramid fusion module for Alzheimer's disease with machine learning," *International Journal of Online Biomedical Engineering*, vol. 19, no. 4, pp. 22–50, 2023. https://doi.org/10.3991/ijoe.v19i04.37677

[22] Liton Devnath, Suhuai Luo, Peter Summons, and Dadong Wang, "Performance comparison of deep learning models for black lung detection on chest X-ray radiographs," In *Proceedings of the 3rd International Conference on Software Engineering and Information Management*, pp. 150–154, 2020. https://doi.org/10.1145/3378936.3378968

[23] B.D.M. Rostami, D.M.C. Anisuzzaman, C. Wang, S. Gopalakrishnan, J. Niezgoda, and Z. Yu, "Multiclass wound image classification using an ensemble deep CNN-based classifier," *Computers in Biology and Medicine*, vol. 134, p. 104536, 2021. https://doi.org/10.1016/j.compbiomed.2021.104536

[24] L. Gang, Z. Haixuan, E. Linning, Z. Ling, L. Yu, and Z. Juming, "Recognition of honeycomb lung in CT images based on improved MobileNet model," *Medical Physics*, vol. 48, no. 8, pp. 4304–4315, 2021. https://doi.org/10.1002/mp.14873

[25] V.K. Singh and M.H. Kolekar, "Deep learning empowered COVID-19 diagnosis using chest CT scan images for collaborative edge-cloud computing platform," *Multimedia Tools and Applications*, vol. 81, no. 1, pp. 3–30, 2022. https://doi.org/10.1007/s11042-021-11158-7

[26] N. Mahendran and S. Kavitha, "A MobileNet-V2 COVID-19: Multi-class classification of the COVID-19 by using CT/CXR images," In *Advances in Electrical and Computer Technologies,* Springer, Singapore, pp. 727–738, 2022. https://doi.org/10.1007/978-981-19-1111-8_55

[27] K. Sengupta and P.R. Srivastava, "Quantum algorithm for quicker clinical prognostic analysis: An application and experimental study using CT scan images of COVID-19 patients," *BMC Medical Informatics and Decision Making*, vol. 21, no. 1, pp. 1–14, 2021. https://doi.org/10.1186/s12911-021-01588-6

[28] C. Wang, D. Chen, L. Hao, X. Liu, Y. Zeng, J. Chen, and G. Zhang, "Pulmonary image classification based on inception-v3 transfer learning model," *IEEE Access*, vol. 7, pp. 146533–146541, 2019. https://doi.org/10.1109/ACCESS.2019.2946000

[29] Robert David, Jared Duke, Advait Jain, Vijay Janapa Reddi, Nat Jeffries, Jian Li, Nick Kreeger, Ian Nappier, Meghna Natraj, Tiezhen Wang, Pete Warden and Rocky Rhodes, "Tensor flow lite micro: Embedded machine learning on TinyML systems," 2021.

[30] D. Das, K.C. Santosh, and U. Pal, "Truncated inception net: COVID-19 outbreak screening using chest X-rays," *Physical and Engineering Sciences in Medicine*, vol. 43, no. 3, pp. 915–925, 2020. https://doi.org/10.1007/s13246-020-00888-x

[31] O.M. Al-Hazaimeh and M. Saraee, "A novel framework for intelligent surveillance system based on abnormal human activity detection in academic environments," *Neural Computing and Applications*, vol. 28, no. 59, pp. 565–572, 2017. https://doi.org/10.1007/s00521-016-2363-z

[32] P. Servanshi, S.K. Bindra, M. Gera, and R. Kaushal, "Covid-19 Detection from CT-scan Images: Empirical Evaluation and Explainability," *Sixth International Conference on Image Information Processing (ICIIP)*, vol. 6, pp. 395–400, 2021. https://doi.org/10.1109/ICIIP53038.2021.9702596

[33] Murukessan Perumal, Akshay Nayak, R. Praneetha Sree, and M. Srinivas, "INASNET: Automatic identification of coronavirus disease (COVID-19) based on chest X-ray using deep neural network," *ISA Transactions*, vol. 124, pp. 82–89, 2022. https://doi.org/10.1016/j.isatra.2022.02.033

[34] H. Malik, T. Anees, and M. Din, et al., "CDC_Net: Multi-classification convolutional neural network model for detection of COVID-19, pneumothorax, pneumonia, lung Cancer, and tuberculosis using chest X-rays," *Multimed Tools Appl*., vol. 82, pp. 13855–13880, 2023. https://doi.org/10.1007/s11042-022-13843-7

[35] I. Chamveha, T. Tongdee, P. Saiviroonporn, and W. Chaisangmongkon, "Local adaptation improves accuracy of deep learning model for automated x-ray thoracic disease detection: A Thai Study," *arXiv preprint*, vol. 2004, p. 10975, 2020.

[36] Vasileios Tsoukas, Eleni Boumpa, Georgios Giannakas, and Athanasios Kakarountas, "A Review of Machine Learning and TinyML in Healthcare," *25th Pan-Hellenic Conference on Informatics*, pp. 69–73, 2022. https://doi.org/10.1145/3503823.3503836

[37] Doyu, Hiroshi Morabito, and Roberto Höller, "Bringing machine learning to the deepest IoT edge with TinyML as-a-service," 2020.

[38] R. Sanchez-Iborra and A.F. Skarmeta, "TinyML-enabled frugal smart objects: Challenges and opportunities," *IEEE Circuits and Systems Magazine*, vol. 20, no. 3, pp. 4–18, 2020. https://doi.org/10.1109/MCAS.2020.3005467

[39] Stanislava Soro, "TinyML for ubiquitous edge AI," *Computer Science and Machine Learning*, 2021.

[40] Colby Banbury, Chuteng Zhou, Igor Fedorov, Ramon Matas Navarro, Urmish Thakker, Dibakar Gope, Vijay Janapa Reddi, Matthew Mattina, and Paul N. Whatmough, "MicroNets: Neural network architectures for deploying TinyML," 2020.
[41] S. Ghamari, K. Ozcan, T. Dinh, A. Melnikov, J. Carvajal, J. Ernst, and S. Chai, "Quantization-guided training for compact TinyML models," 2021.
[42] M. Al-Hazaimeh and M. Saraee, "Geometrical-based approach for robust human image detection," *Multimedia Tools and Applications*, vol. 78, pp. 7029–7053, 2019. https://doi.org/10.1007/s11042-018-6401-y

7 AUTHORS

Dipanjan Acharya, K. Eashwer and Soumya Kumar are students at the Division of Sensors and Biomedical Technology, School of Electronics Engineering, VIT University, Vellore (E-mail: acharya.dipanjan02@gmail.com, eashwerkailasam01@gmail.com, ksoumya1910@gmail.com).

R. Sivakumar is a Professor at the Division of Sensors and Biomedical Technology, School of Electronics Engineering, VIT University, Vellore. His research interest include Signal Processing, Image Processing, and Biomedical Engineering (E-mail: rsivakumar@vit.ac.in).

P.C. Kishoreraja is a Professor at the School of Information Technology in Vellore Institute of Technology (VIT), Vellore, India. His research interests include Machine Learning Algorithms and Internet of Things (E-mail: kishoreraja.pc@vit.ac.in).

Dr. Ramasamy Srinivasagan is a Professor at the Faculty of Computer Engineering, in King Faisal University, Saudi Arabia (E-mail: rsamy@kfu.edu.sa).