Multimedia Tools and Applications (2023) 82:39211–39254
https://doi.org/10.1007/s11042-023-14831-1

MediNet: transfer learning approach with MediNet medical visual database

Hatice Catal Reis[1] · Veysel Turk[2] · Kourosh Khoshelham[3] · Serhat Kaya[4]

Received: 4 February 2022 / Revised: 6 April 2022 / Accepted: 6 February 2023 /
Published online: 20 March 2023

Abstract

The rapid development of machine learning has increased interest in the use of deep learning methods in medical research. Deep learning in the medical field is used in disease detection and classification problems in the clinical decision-making process. Large amounts of labeled datasets are often required to train deep neural networks; however, in the medical field, the lack of a sufficient number of images in datasets and the difficulties encountered during data collection are among the main problems. In this study, we propose MediNet, a new 10-class visual dataset consisting of Rontgen (X-ray), Computed Tomography (CT), Magnetic Resonance Imaging (MRI), Ultrasound, and Histopathological images such as calcaneal normal, calcaneal tumor, colon benign colon adenocarcinoma, brain normal, brain tumor, breast benign, breast malignant, chest normal, chest pneumonia. AlexNet, VGG19-BN, Inception V3, DenseNet 121, ResNet 101, EfficientNet B0, Nested-LSTM + CNN, and proposed RdiNet deep learning algorithms are used in the transfer learning for pre-training and classification application. Transfer learning aims to apply previously learned knowledge in a new task. Seven algorithms were trained with the MediNet dataset, and the models obtained from these algorithms, namely feature vectors, were recorded. Pre-training models were used for classification studies on chest X-ray images, diabetic retinopathy, and Covid-19 datasets with the transfer learning technique. In performance measurement, an accuracy of 94.84% was obtained in the traditional classification study for the InceptionV3 model in the classification study performed on the Chest X-Ray Images dataset, and the accuracy was increased 98.71% after the transfer learning technique was applied. In the Covid-19 dataset, the classification success of the DenseNet121 model before pre-trained was 88%, while the performance after the transfer application with MediNet was 92%. In

✉ Hatice Catal Reis
hatice.catal@yahoo.com.tr; hcatal@gumushane.edu.tr

[1] Department of Geomatics Engineering, Gumushane University, 2900 Gumushane, Turkey

[2] Department of Computer Engineering, University of Harran, Sanliurfa, Turkey

[3] Department of Infrastructure Engineering, The University of Melbourne, Parkville 3052, Australia

[4] Department of Mining Engineering, Dicle University, Diyarbakir, Turkey

the Diabetic retinopathy dataset, the classification success of the Nested-LSTM + CNN model before pre-trained was 79.35%, while the classification success was 81.52% after the transfer application with MediNet. The comparison of results obtained from experimental studies observed that the proposed method produced more successful results.

Keywords MediNet · Medical images · Classification · Transfer learning · RdiNet · Deep neural networks

1 Introduction

Availability of data consisting of images [1], high level of distinguishing ability of deep convolutional neural networks (CNNs), artificial intelligence (AI) technologies produce innovative solutions with convolutional neural network-based deep learning algorithms in medical researches [52]. Artificial intelligence technologies are essential in detecting current diseases or predicting possible future diseases [39].

Artificial neural networks have been applied in many disciplines for cancer detection and recently to detect and prevent COVID-19 disease [11, 38, 41]. Deep convolutional neural networks from artificial neural networks have been proven successful in image processing applications [3]. The deep learning algorithm is a supervised machine learning technique to find the correct match between input data and output data with the relevant labeled data during the training of a model with a labeled dataset [29]. Machine learning methods are a common solution technique in classification problems and one of the leading research topics [31]. Machine learning methods are widely discussed for use in therapeutic applications [31]. One of the machine learning methods, training of CNN-based deep learning architecture generally performs well when done with large amounts of labeled datasets [27]. Deep neural networks have been a frequently used method in problem-solving because they effectively solve complex linear or nonlinear problems [40]. Deep learning in medical image analysis is generally used in classification, segmentation, registration, detection, and localization studies [24, 32]. It is possible to see examples of these studies in the literature. Researchers conducted a six-class classification and segmentation study with 13,673 fundus images [33]. Another research group used 129,450 clinical data to generate dermatologist-level results using deep neural networks for a skin cancer classification study [16]. The researchers conducted a classification study using pre-trained deep learning models on a brain dataset consisting of 3064 MRI images [42]. In general, for deep learning studies to be successful, large amounts of labeled datasets containing hundreds of thousands of samples can be used [43]. However; There are more unlabeled datasets in many areas, including the medical imaging field; many of the labeled datasets that can be used in the medical field are limited, and labeling requires expert radiologists or experts' knowledge, meeting these needs is time-consuming and costly [15]. For example, in a manual segmentation study with brain magnetic resonance (MRI) images of six-month-old babies in the iSeg2017 competition, an experienced neuroradiologist spent an average of one week for segmentation of each brain MRI data [51]. As a result, new and efficient approaches are needed in order to produce high-performance results with a small amount of data in studies to be carried out in the medical field.

Data augmentation method for increasing synthetic data in order to overcome the problem arising from the insufficient datasets in medical image processing studies and solving the overfitting problem in CNN algorithms [3, 46], semi-supervised learning, which is a method where the network trained

with a small amount of labeled data, where predictions are produced for untagged data, then the most appropriate ones among the predicted tags are selected, the network is retrained with the labeled data, and the training process continues in this way [8] and inspired by human thought processes [4], training a model on a source dataset (big data) and transferring the learned information to be used in a target dataset (small size data) and techniques such as the transfer learning method [44], which is a deep learning application, can be applied. The focus is on using pre-trained networks with transfer learning to increase the success/performance of deep learning algorithms trained from scratch in detecting Covid-19, detecting the severity level of eye diabetic retinopathy, and detecting brain tumors [14, 36, 37, 50]. To achieve more successful results in image processing, it can often be necessary to have large datasets. However, collecting and classifying these data is a time-consuming and costly process. In addition, these data should be categorically separated to be used in image processing, and specialists are needed here. To overcome the problem caused by the insufficiency of available data sets in medical image processing studies, a heterogeneous MediNet dataset consisting only of medical images has been proposed. In other words, with this project, we present a dataset that can be used in transfer learning studies.

In this study, we introduce MediNet, a visual database consisting only of medical images for transfer learning applications. MediNet dataset has ten classes consisting of foot, colon, brain, breast, and chest images. MediNet dataset was trained one by one with deep learning algorithms AlexNet, VGG19-BN, InceptionV3, DenseNet121, RdiNet, ResNet101, EfficientNetB0, Nested LSTM + CNN models. Feature sets and weight vectors obtained from pre-training models with transfer learning were retrained with the fine-tuning method for chest, diabetic retinopathy, and Covid-19 datasets with all transfer learning algorithms and used in disease detection. By transfer learning, we have pre-training a network with one dataset and applying it to another dataset. The experimental results in the study are presented in the form of products before and after transfer learning. Accuracy, Sensitivity, Specificity, Confusion Matrix, ROC-AUC metrics were used to evaluate the models' performance. As a result, we believe that the proposed method will contribute significantly to medical image processing studies. The general framework for the study is given in Fig. 1.

The contributions of our work can be summarized as follows:

1. The 10-category MediNet dataset consisting of only medical images that can be used in the transfer learning process in medical imaging applications has been introduced.

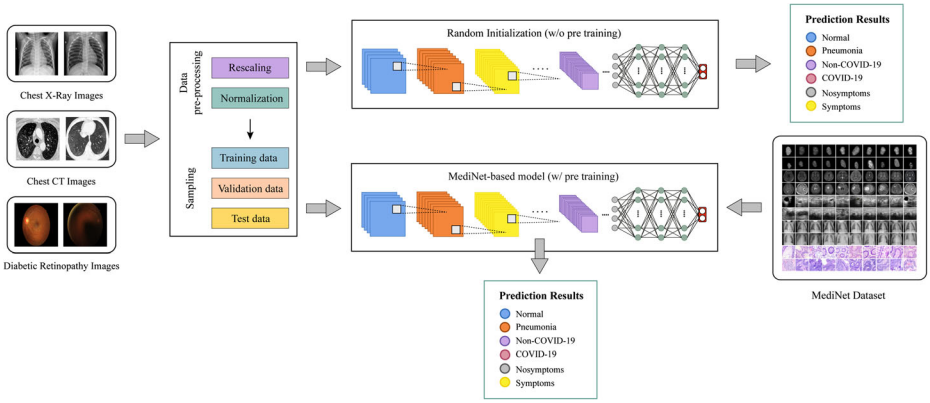


Fig. 1 Framework of study

2. In the evaluation of the transfer learning application proposed with the MediNet dataset; AlexNet, VGG19-BN, InceptionV3, DenseNet121, ResNet101, EfficientNetB0, Nested-LSTM + CNN, and proposed RdiNet deep neural networks and LSTM, GRU (eg, DenseNet121 + LSTM, InceptionV3 + GRU, etc.), deep neural networks (eg., DenseNet121 + RdiNet, InceptionV3 + DenseNet121, etc.) recommended hybrid methods were used.
3. The proposed transfer learning method has been evaluated in binary classification studies with three datasets consisting of chest X-ray (Chest X-Ray Images Dataset) and CT (COVID19-CT Dataset, Diabetic Retinopathy Dataset) medical imaging techniques.
4. A training and validation testing approach was used in the classification studies with deep neural networks. In addition, the K-fold cross-validation method was applied in the transfer learning study conducted with the chest X-ray (CXR) dataset and the InceptionV3 model. Experimental results in the application are given before and after pre-training.
5. Another contribution within the scope of the study, the RdiNet deep neural network, for which residual networks and additional feature vectors are developed, is proposed. The proposed neural network can offer an alternative view of the vanishing gradient problem.
6. Fine-tuning method was applied in the transfer learning process.
7. Experimental results in the study are given comparatively before pre-trained (random initialization) and after pre-trained (MediNet-based model).

The study has been designed under the following headings;

MediNet dataset is introduced in the second section; material and methods are explained in Section 3, and experimental results are presented in Section 4. Next, Section 5 describes the limitations of the study and the solution methods. Then, a discussion in Section 6 and conclusions and future work are given in Section 7.

2 MediNet dataset

Data from five different imaging techniques, foot dataset as CT data, colon dataset as pathology data, brain dataset as MRI data, breast dataset as ultrasound data, and chest dataset as X-ray data were used. Each dataset has two categories within itself. In other words, there is a dataset with ten classes in total and there are also 4348 medical images in the MediNet dataset. In the pre-processing study, each of the images was resized to 224 × 224 with the image module's resize function in the PIL library. During the training phase, the data was divided into 9: 1 (training dataset 3913; validation dataset 435 for model training). Table 1 shows the medical imaging techniques of the data used in the MediNet dataset, the categories in the dataset, the labels determined for ten categories, and the number of images belonging to each category are given.

2.1 Description of images in MediNet dataset

Computed Tomography (CT): CT-scan consists of foot images. Calcaneus bone, one of the foot bones, has 94 CT-images positive for calcaneal tumors and 155 CT -images negative. The CT images in the dataset have different sizes, such as 784 × 862, 743 × 874, 868 × 880, and

Table 1 MediNet Data Category Labels

Medical Imaging Technique	Label	Category	Number of Images	Year
Computed Tomography (CT)	0	calcaneal healthy (tumor no)	155	2017
	1	calcaneal tumor (tumor yes)	94	
Microscopic Imaging	2	colon benign tissue	800	2020
	3	colon adenocarcinoma	800	
Magnetic Resonance Imaging (MRI)	4	brain healthy (tumor no)	98	2019
	5	brain tumor (tumor yes)	154	
Ultrasound	6	breast benign	437	2018
	7	breast malignant	210	
X-ray	8	chest normal	800	2018
	9	chest pneumonia	800	

all images used in the study were resized to 224 × 224 [45]. Histopathology: For histopathology data, the LC25000 dataset was used. This dataset contains colon_image_sets and lung_image_sets datasets. There are 25,000 images in total. All images are 768 × 768 in size and .jpg format. In the colon dataset; there are two classes: colon adenocarcinoma and benign colon tissue. There are 5000 images in colon adenocarcinoma and 5000 images in Colon benign tissue. There are a total of 10,000 images in the colon dataset. In the lung dataset; there are three different classes: lung adenocarcinoma, lung benign tissue, and lung squamous cell carcinoma. There are 5000 images in Lung adenocarcinoma, 5000 images in benign lung tissue, and 5000 images in lung squamous cell carcinoma. There are a total of 15,000 images in the lung dataset. However, due to the RAM capacity of our personal computer, the data numbers are used as follows. 800 images were used in the Colon adenocarcinoma class, and 800 images were used in the Colon benign tissue class. All images used in the MediNet dataset were resized to 224 × 224 [6, 7]. Magnetic Resonance Imaging (MRI): brain MRI-images was used. It consists of two classes as tumorous and non-tumorous (i.e., healthy) data according to the condition of the tumor. The number of tumor data is 155, and the number of healthy data is 98. The medical images in the dataset are in different sizes, such as 630 × 630, 300 × 168, 200 × 252, and all images used in the study were resized to 224 × 224 [9, 26]. Ultrasound; It consists of breast images. There are two types of classes in the Breast dataset: Breast Benign and Breast Malignant. There are 437 images in the Breast Benign dataset and there are 210 images in the Breast Malignant dataset. All images in the dataset used in the study were resized to 224 × 224 [2]. X-ray: Chest X-Ray images were used for the X-ray dataset. The dataset consists of two categories: 5856 X-Ray images in .jpg format and Pneumonia / Normal. There are 1583 data in the normal images, 4273 images in the Pneumonia dataset, and 5856 images in total. In the project, 800 images were used in the Normal and 800 images were used in the Pneumonia dataset. The dataset consists of different sizes such as 712 × 439, 1240 × 840, 2090 × 1358. All images in the dataset used in the study were resized to 224 × 224 [23, 25]. MediNet dataset is a database of medical images. We aim to use the dataset in transfer learning applications. In Fig. 2, examples of images used in the database are given. While 3922 medical images in the MediNet visual database were used in the training of the deep neural networks within the scope of the study, 436 medical images were used for the validation dataset of the deep neural networks (Table 2).

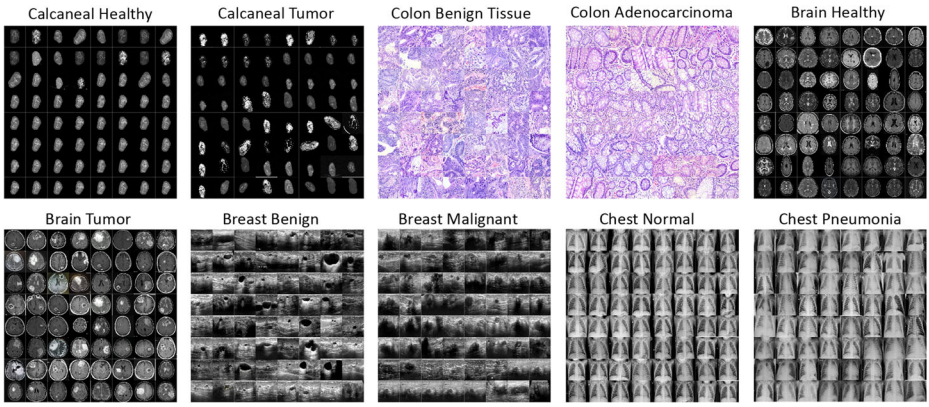


Fig. 2 Medical image examples

3 Material and method

The MediNet dataset, by ten classes, was trained with deep learning algorithms. Then we used these pre-training models for classification applications in the chest, Covid-19, and eye datasets. Ten-classes training was carried out with deep learning algorithms AlexNet, VGG19-BN, InceptionV3, DenseNet121, ResNet101, EfficientNetB0, Nested-LSTM + CNN with the Medinet dataset. A binary classification study was carried out on three separate datasets consisting of medical images with the transfer learning study of the obtained weight vectors. Accuracy, Sensitivity, Specificity, Confusion Matrix, and ROC AUC metrics were used to evaluate the results of binary classification studies.

In Section 1, datasets used in binary classification are introduced in the following. Information about the deep learning models used in the study is given in Section 2. Section 3, the theoretical framework for training deep neural networks, is given. Then, in Section 4, hyperparameters used in deep neural networks are given. The transfer learning study is explained in Section 5. In Section 6, the hybrid models suggested in the study are given. In Section 7, the Fine-Tuning application is explained. In section 8, the theoretical framework of the K-fold cross-validation method is given and in section 9, information about the performance metrics used.

3.1 Datasets used in the classification

In this study, datasets consisting of chest X-ray and CT images were used. Chest X-ray images [25] consist of chest X-ray images (1547); it consists of normal, pneumonia categories and randomly selected data.

COVID-19 CT-Scans [19] consist of CT images consisting of non-COVID-19, COVID-19 categories, and 746 images. Another dataset of CT images used is Diabetic Retinopathy images (915) [17]; It consists of two categories, symptoms and no symptoms. In the classification, 90% of the chest X-ray and CT dataset samples (training dataset + validation dataset) were used to train the models, and 10% of the data was used in the testing process (Table 3). Sample data of Chest X-ray and CT images are shown in Fig. 3.

Table 2 MediNet Dataset

Medical Imaging Technique	Author	Dataset	Categories	Train/ Validation	Link
Computed Tomography (CT)	Reis	Calcaneus Benign Tumor	calcaneal healthy (tumor no) calcaneal tumor (tumor yes)	3922/436	None
Microscopic Imaging	Borkowski et al.	LC25000	colon benign tissue colon adenocarcinoma		https://www.kaggle.com/andrewmvd/lung-and-colon-cancer-histopathological-images (Accessed 1 March. 2021)
Magnetic Resonance Imaging (MRI)	Chakrabarty	Brain MRI Images for Brain Tumor Detection	brain healthy (tumor no) brain tumor (tumor yes)		https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection (Accessed 1 March. 2021)
Ultrasound	Al-Dhabyani et al.	Breast Ultrasound Dataset	breast benign breast malignant		https://scholar.cu.edu.eg/Dataset_BUSI.zip (Accessed 1 March. 2021)
X-ray	Kermany et al.	Chest X-Ray Images (Pneumonia)	chest normal chest pneumonia		https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia (Accessed 1 March. 2021)

Table 3 Data Statistics in Classification

Medical Imaging Technique	Author	Dataset	Categories	Data	Train+Validation / Test	Link
Chest X-Ray	Kermany et al.	Chest X-Ray Images Dataset	Normal	683	1392 / 155	https://data.mendeley.com/datasets/rscbjbr9sj/2 (Accessed 1 March. 2021)
			Pneumonia	864		
CT	He et al.	COVID19-CT Dataset	Non-COVID-19	397	671 / 75	https://github.com/UCSD-AI4H/COVID-CT (Accessed 1 March. 2021)
			COVID-19	346		
CT	Arjun	Diabetic Retinopathy Dataset	No symptoms	320	823 / 92	https://github.com/javathunderman/retinopathydataset (Accessed 1 March. 2021)
			Symptoms	595		

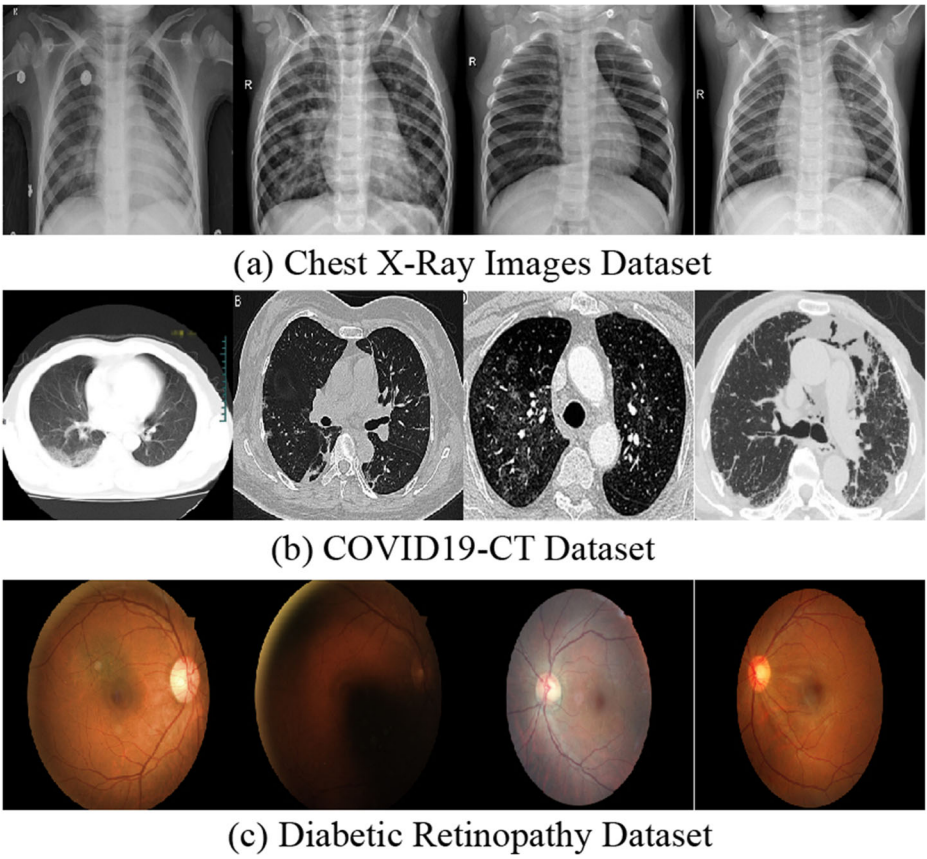


Fig. 3 Examples of chest x-ray images, covid-19 CT images and diabetic retinopathy images

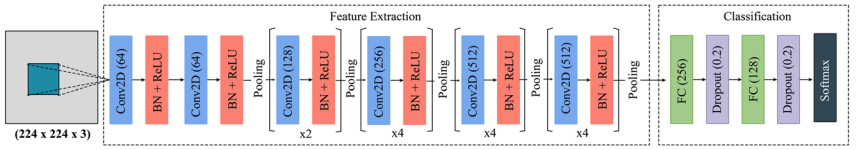
3.2 Deep learning models

In obtaining weight vectors and classification in the study, AlexNet [30], VGG19-BN [47], InceptionV3 [48], DenseNet121 [21], ResNet101 [18], EfficientNetB0 [49], Nested-LSTM [35] + CNN, and proposed RdiNet deep learning algorithms were used. The flow chart of VGG19-BN, InceptionV3, Nested-LSTM + CNN, ResNet101, and DenseNet121 architectures is given in Fig. 4.

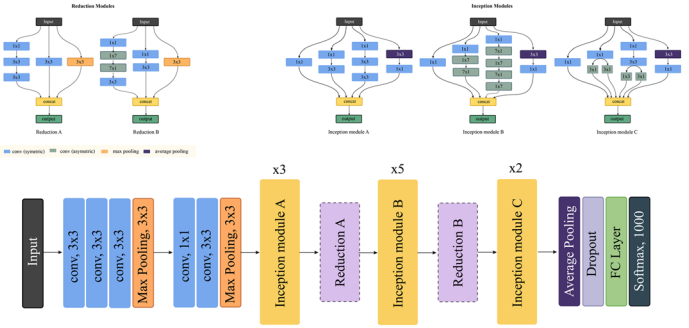
3.2.1 AlexNet

Krishevsky et al. (2012) proposed the AlexNet model, 5 convolution layers, it is a Convolutional Neural Networks (CNN) based architecture with 1000 classes in its initial architecture, consisting of 3 pooling layers and 3 fully connected layers. The output layer of the architecture consists of the softmax layer.

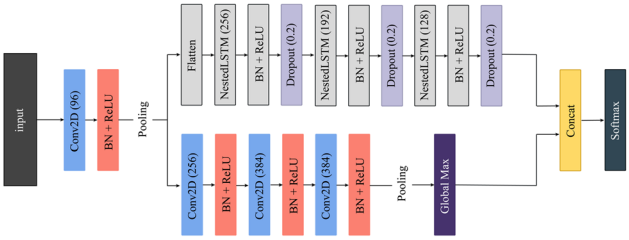
The image input size of the architecture is 227 × 227.



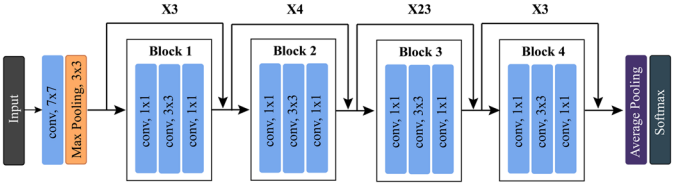
(**a**) VGG19-BN



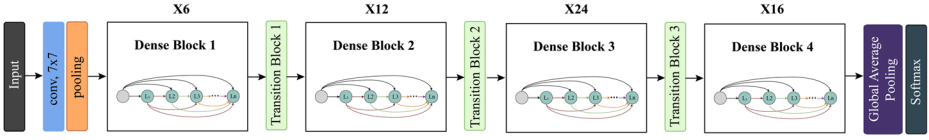
(**b**) Inception v3



(**c**) Nested-LSTM + CNN



(**d**) ResNet-101



(**e**) DenseNet-121

Fig. 4 Architectures of the four CNN base classifiers: (**a**) VGG19-BN, (**b**) InceptionV3, (**c**) Nested-LSTM + CNN, (**d**) ResNet101, and (**e**) DenseNet121

3.2.2 VGG19

According to Simonian et al. (2014) is a CNN-based architecture with 16 convolution layers, three fully connected layers, five max-pooling, and 1000 classes in the initial architecture. The output layer of the architecture consists of the softmax layer. The image input size of the architecture is 224 × 224.

3.2.3 InceptionV3

Szegedy et al. (2014) is the GoogLeNet architecture introduced in the ILSVRRC ImageNet competition.

The architecture was the most successful model with the lowest error value in the competition.

Additional feature vectors have been proposed to solve the gradient disappearance problem in hierarchical feed-forward neural networks. The architecture consists of approximately 24 million parameters and consists of 154 layers. The image input size of the architecture is 299 × 299. The backbone of the Inception architecture consists of Inception modules A, B, C modules and Reduction modules.

In Inception module A, there are four 1 × 1, three 3 × 3 convolution layers, and 3 × 3 average pooling. Inception module B includes 4 1 × 1, three 7 × 7 convolutions, and 3 × 3 average pooling. The symmetrical 7 × 7 convolution layer in the module is divided into asymmetrical 7 × 1 and 1 × 7 convolution layers. There are 4 1 × 1, 3 × 3 symmetric convolution layers and 3 × 3 average pooling in Inception module C. The symmetrical 3 × 3 convolution layer is divided into asymmetrical 3 × 1 and 1 × 3 convolution layers.

The reduction module includes an asymmetric convolution network, max pooling, and a parallel symmetric convolution network method is applied in the module.

3.2.4 DenseNet121

Huang et al. (2017) suggested a proposed method that increases features' use in solving the gradient disappearance problem. In architecture, each layer is directly linked to all previous layers. Attributes collected in a specific layer are combined with the concatenate layer. DenseBlock in the DenseNet-121 architecture forms the backbone of the architecture. The architecture consists of 224 × 224 input size, 121 layers, four dense blocks, three transition layers, a softmax, and an output layer.

3.2.5 ResNet101

He et al. (2016) proposed the residual network method in ResNet architecture to solve the gradient disappearance problem. ResNet architecture consists of 50, 101, and 152 layers according to the number of layers. In this study, ResNet-101 architecture was used. The architecture consists of approximately 45 million parameters and consists of 224 × 224 input sizes and 101 layers.

3.2.6 EfficientNetB0

According to Tan et al. (2019) suggested. The architecture is built using the squeeze-and-excitation blocks and inverted bottleneck residual blocks structures in the MobileNetV2 architecture. The authors proposed the Swish (product of linear and sigmoid activation) activation function within the scope of the study. The architecture, which consists of approximately 5.3 million parameters, consists of a 224 × 224 input size.

3.2.7 Nested-LSTM

Moniz et al. (2018) suggested. Nested LSTM architectures are used to solve the vanishing gradient problem experienced in the proposed architecture RNN architecture. Nested LSTMs increase the depth of LSTM architectures over stacked LSTM. The proposed architecture outperforms LSTM and stacked LSTM architectures.

3.2.8 Proposed RdiNet model

Deep convolutional neural networks have produced high performances in image processing [22].

Here, the depth of the architectures is generally increased for the performance increase in the classification, detection, and segmentation studies of the architectures. However, as the depth of the deep neural networks increases, the problem of gradient loss arises in the backpropagation process. As a result, the weights cannot be updated in the networks due to the gradients lost in the backpropagation process, and in this case, the training of the architectures will be difficult. The RdiNet deep neural network architecture proposed in this study is developed with residual networks and additional feature vectors, offering an alternative view of solving gradient disappearance. RdiNet architecture is designed using the TensorFlow Keras library. RdiNet architecture is basically developed with Residual Blocks (ResBlocks), Inception module, Dense Blocks, and Transition Layer. The flow chart of the RdiNet architecture is given in Fig. 5, and the technical information of the architecture is shown in Table 4.

3.3 Training of deep learning algorithms

In this section, the theoretical framework of the training and testing phases of deep neural networks is given.

3.3.1 Implementation details

90% of the datasets consisting of CXR and CT images were used in training (the validation dataset consists of 20% of the training dataset), and 10% were used in the test set. In the study, input shape 224x224x3, number of epochs = 50, batch size = 64, patience = 2, factor = 0.5, save best only = true, mode = min, monitor = "val_loss", metrics = ['accuracy' in deep neural networks], loss = categorical crossentropy, optimization = adam, and learning rate = 1e−06 parameters were used (Table 5). The learning rate value of the models was reduced by 0.5 (factor) in training in cases where learning did not occur (if the validation loss value does not decrease during 2 (patience = 2) cycles). The categorical cross-entropy loss function used

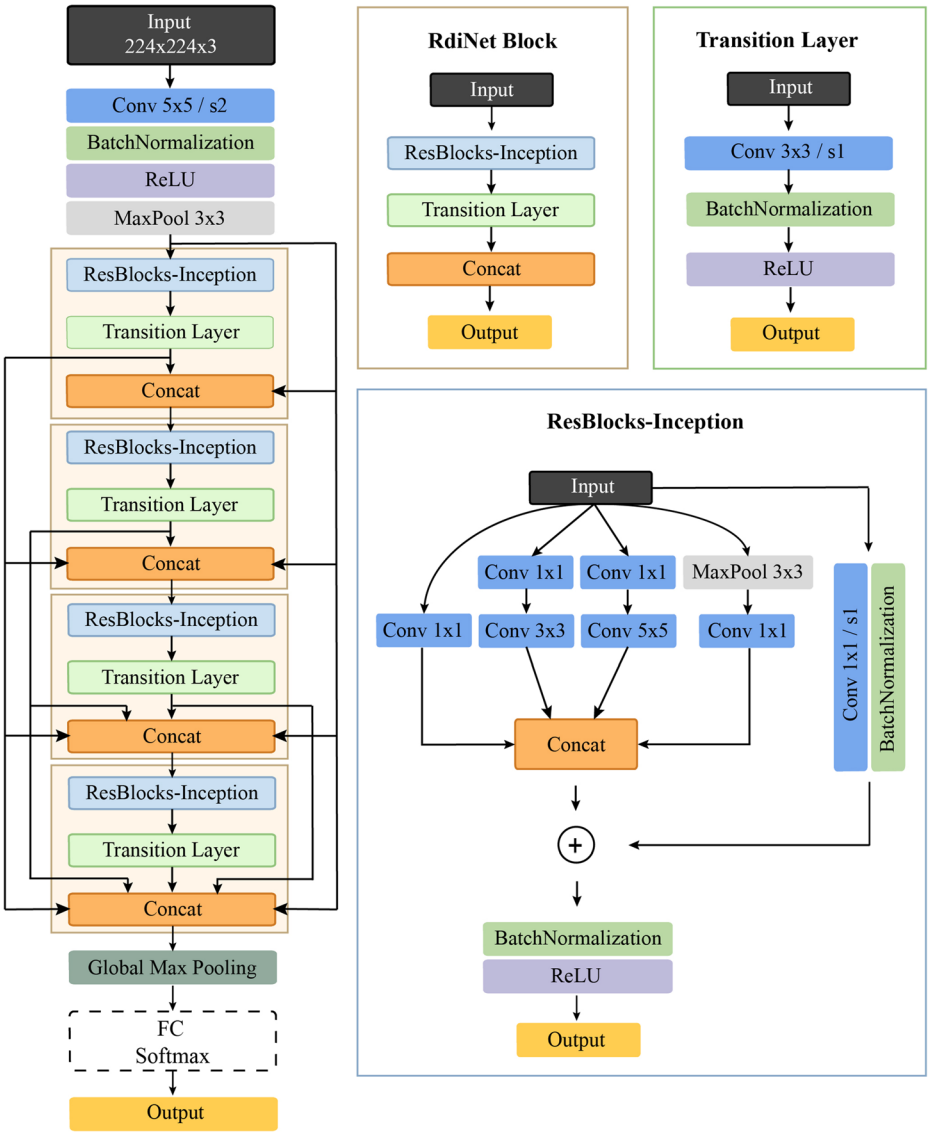


Fig. 5 Architecture of the proposed RdiNet algorithm

in deep neural networks measures the difference between the actual and the predicted value. The man optimization function used in deep neural networks is one of the standard and most effective method used in gradient descent optimization to reduce cross-entropy loss. *Gradient descent optimization* is a popular method used to minimize the training error of models. Deep neural networks were trained with the training/validation dataset for 100 cycles on 50 CT medicine images from CXR medical images. Among the weight vectors obtained from the loop, the lowest vector according to the validation loss value was used for the test.

Table 4 The detailed structure of the proposed RdiNet. The "@" represents the kernel size value, and the "&" represents the strides value. "||" or connector. BN: BatchNormalization. The output size values in the 4 x RdiNet module are the outputs of the latest module

Stages	Layers	Output size
Input Image	Input Layer	224x224x3
Stem Layer	Conv2D(64) @5×5 &2×2+BN+ReLU	112x112x64
	MaxPooling2D	37×37×64
4 x RdiNet Block	ResNet-Inception Blok (64, 128, 156, 128, 64, 32) @1×1||3×3||5×5 &1×1	37x37x316
	Transition Katmanı (128, 196, 256) @5×5 &2×2+BN+ReLU	37x37x128
	Concatenate Katmanı	37x37x772
Classification Block	GlobalMaxPooling2D	772
	Softmax	2

3.3.2 A performance audit of their architecture

Binary classification of datasets consisting of deep neural network architectures CXR and CT medical images were also used. In classification, deep neural networks were trained and tested with the train-validation-test approach. To improve the performance of deep neural networks, the proposed MediNet medical dataset and transfer learning technique, LSTM, and GRU from RNN methods, hybrid method, and hybrid methods in which deep learning algorithms are used together were applied. In addition, the K-fold cross-validation method with CXR medical images was applied to evaluate the transfer learning approach proposed with the MediNet dataset. Confusion metrics (CM), accuracy (ACC), sensitivity (SN), positive predictive value (PPV), cohen's kappa (κ), f1-score (F1), training loss/training accuracy, validation loss in empirical outcome evaluation of deep neural networks /validation accuracy graphs were used.

3.4 Deep learning algorithms hyper-parameters

This section gives the theoretical background of the hyperparameters used in the proposed deep convolutional neural networks. Hyperparameters are given in Table 5. The hyperparameters used were determined as a result of the experimental process.

Input Shape; Input layer size in deep neural networks. Epoch; The number of cycles in training deep neural networks. Batch size; is the number of subsamples used simultaneously in forward/backward propagation of deep neural networks during training. It may vary depending

Table 5 Parameters Used in Deep Learning Models

Hyper parameters	Value
Input Shape	224x224x3
Epoch	50/100
Batch size	64
Patience	2
Factor	0.5
Save Best Only	true
Mode	'min'
Monitor	'val_loss'
Metrics	'accuracy'
Loss	'categorical_crossentropy'
Optimizer	'adam'
Initial learn rate	1e-06

on the ram capacity. Patience; the number of cycles in which the loss value does not decrease (training stops). After the patience value, the learning rate value decreases with the factor value. Factor; The amount that the learning rate will decrease. Metrics; list of metrics evaluated by models. Monitor_SaveBestOnly_Mode; deep neural networks generate a weight vector at the end of each cycle. Here, the weight vector with the lowest (mode: "min") validation loss (monitor: 'val_loss') value according to the Monitor_SaveBestOnly_Mode parameters was recorded (save best only) and evaluated empirically with the test dataset. Learning rate; rate of convergence of backpropagation. Loss Function; It is called minimizing the objective function value. Within the scope of the study, categorical cross-entropy was used. The mathematical output of the categorical cross-entropy function is:

$$L_{CCE}\left(y,\widehat{y}\right) = -\frac{1}{N}\sum_{i=0}^{m}\sum_{j=0}^{n}\left(y_{ij} * \log\left(\widehat{y}_{ij}\right)\right) \tag{1}$$

where; ŷ prediction and y real value.

In this study, Kingma et al. (2014) proposed the Adam (Adaptive Momentum) optimization function [28]. The Adam function is a Gradient Descent algorithm used to reduce the cross-entropy loss. The Adam optimization function has been developed with a hybrid approach with momentum and Adagrad methods. As a result of the use of the Adagrad function, a learning problem has arisen. Here, the Root Mean Square Propagation (RMSprop) method has been applied to solve the problem. The pseudocode of the Adam function is given in Table 6.

3.5 Transfer learning and pre-training

Transfer learning study is a method that enables the use of pre-training models in a new problem. In this study, the MediNet database was created to be used in transfer learning studies. The aim of the study is to analyze the success of pre-training models with the MediNet dataset. The transfer learning and classification process: Deep learning models were trained

Table 6 Pseudocode for Adam algorithm

α: step size
$\beta_1 ve\ \beta_2$: exponential decay rates
$_0$: *i*nitial parameter vector
$m_0 \leftarrow 0$ (initialize 1st moment vector)
$v_0 \leftarrow 0$ (initialize 2st moment vector)
$t \leftarrow 0$ (initialize timestep)
while $_t$ not converged **do**
$t \leftarrow t+1$
$g_t \leftarrow \nabla f_t\ (_{t-1})$ (compute gradients)
$m_t \leftarrow \beta_1 m_{t-1} + \ (1-\beta_1) g_t$ (update biased first moment estimate)
$v_t \leftarrow \beta_2 v_{t-1} + \ (1-\beta_2) g_t^2$ (update biased second moment estimate)
$m_t^{'} \leftarrow \frac{m_t}{1-\beta_1^t}$ (correct bias in first moment estimate)
$v_t^{'} \leftarrow \frac{v_t}{1-\beta_2^t}$ (correct bias in first moment estimate)
$_{t+1} \leftarrow\ _t - \frac{\alpha}{\sqrt{v_t^{'}}+\varepsilon} m_t^{'}$ (update parameters)
end while
return $_t$ (resulting parameters)

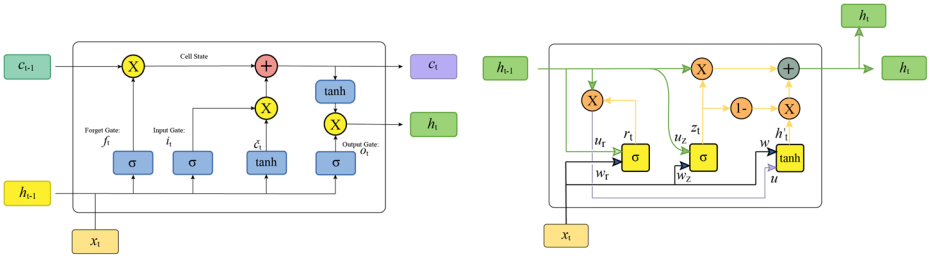
using the MediNet dataset. The weight vectors of the pre-training model were retrained with new datasets prepared for binary classification with transfer learning, and then performance analysis was performed for binary classification. The weight vector of the pre-training models and the number of parameters used in each deep convolutional neural network are given in Table 8.

3.6 Hybrid models

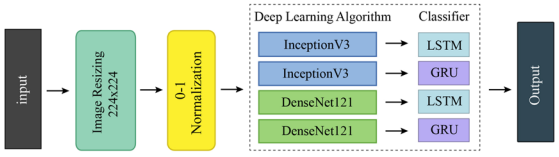
Deep neural networks and deep neural network RNN (LSTM, GRU) methods and hybrid methods have been proposed in the performance evaluation of the transfer learning application proposed with the MediNet dataset. Experimental results of hybrid method studies are given comparatively before pre-trained and after pre-trained. The flowchart of LSTM, GRU, and hybrid methods is given in Fig. 6.

3.6.1 Hybrid application of deep neural networks and LSTM, GRU

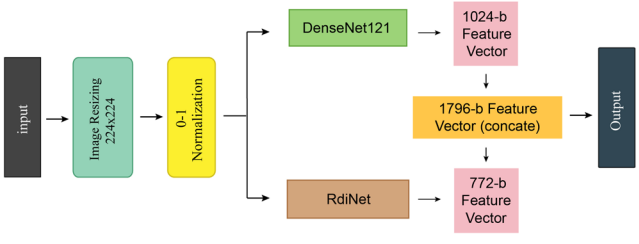
LSTM [20] architecture is one of the RNN architectures that provides unidirectional information flow and can store all helpful information thanks to the powerful memories in their architecture.



(a) LSTM Layer (b) GRU Layer



(c) Deep Learning Algorithms with LSTM and GRU



(d) Hybrid deep learning algorithm with DenseNet121 and RdiNet model

Fig. 6 LSTM, GRU, Hybrid method

GRU [10] is proposed to prevent memory loss in RNN architectures. The study applied hybrid methods with LSTM, GRU layer, and DenseNet121, InceptionV3 deep neural networks.

LSTM Hochreiter et al. It is an RNN architecture proposed by (1997). It is proposed to solve the gradient disappearance problem, a significant problem for RNN systems. They used memory cells to solve the gradient problem. Memory cells form the decision mechanisms of the LSTM architecture.

Gates consisting of the sigmoid function and the dot product are the decision mechanisms of the memory cells. Each memory cell consists of 3 gates: input, forget and output. The gateway decides which of the current or current information to keep. The unit responsible for information transport is updated at the cell state entrance gate. Forget the door; the unit responsible for the selection of login information.

The output gate determines what information will pass to the following memory cells or the output layer.

The mathematical equations in LSTM architecture:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{3}$$

$$\check{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \check{c}_t \tag{5}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{6}$$

$$h_t = o_t \odot tanh\,(c_t) \tag{7}$$

where; $i_t\, f_t\, o_t$; respectively; input/update, forget and output gate W_i, W_f, W_o, W_c; input, forget, output gate and cell state weight matrices, b_i, b_f, b_o, b_c input, forget, output and cell state bias value, σ sigmoid function, x_t; input vector, U; m x m matrix, $\check{c}_t$; cell input activation vector, c_t, c_{t-1}; t and *t-1* cell states, $\odot$ vector multiply and h_t is the hidden state value are shown.

GRU It is a variant of LSTM that combines GRU input port, forget port, hidden state, and memory state [61].

GRU architecture consists of reset and update ports. The reset gate decides which information to forget.

The update gate is similar to the forget and input gates in LSTM architectures. This gate decides which information can and cannot be deleted.

Equations of architecture;

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{8}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{9}$$

$$\widehat{\mathrm{h}}_t = \phi(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \tag{10}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widehat{\mathrm{h}}_t \tag{11}$$

where;

z_t, r_t,update and reset gate, $\widehat{h}_t$, h_t expected activation and output vector, σ, ϕ sigmoid and hyperbolic tangent activation function, W weight vector, x_t input vector, U m x m vector, b bias value, $\odot$ vector multiplication.

3.6.2 Hybrid application of deep neural networks

This method combines feature vectors obtained from deep neural networks and applies them to the classification problem. In this study, InceptionV3 + RdiNet, InceptionV3 + DenseNet121, InceptionV3 + AlexNet, DenseNet121 + RdiNet hybrid methods are proposed.

3.7 Fine tuning/frozen

It is frequently used to take advantage of the latest improvements in deep convolutional neural networks trained with large datasets. Frozen operation is performed according to the similarity between the datasets used in the pre-trained models and the datasets used in the studies [13, 34].

3.8 Cross-validation

Cross-validation is a vital model validation technique in revealing the accuracy and classification success of the model. Cross-validation is a crucial method in identifying under-learning and overlearning during the development of a model. In this method, the original clusters are divided into k parts.

While the "k-1" part is used in the training of the model, the remaining amount is used in the testing process of the model. The process here repeats k number of times. The flowchart of the cross-validation method applied in this study is given in Fig. 7.

3.9 Performance metrics for binary classification

In binary classification studies, CM is 2-dimensional (Table 7). The CM is determined by comparing the actual and predicted class labels. In Table 7, the rows show actual and columns the predicted class examples. According to Table 7, in the problem of diagnosis of pneumonia;

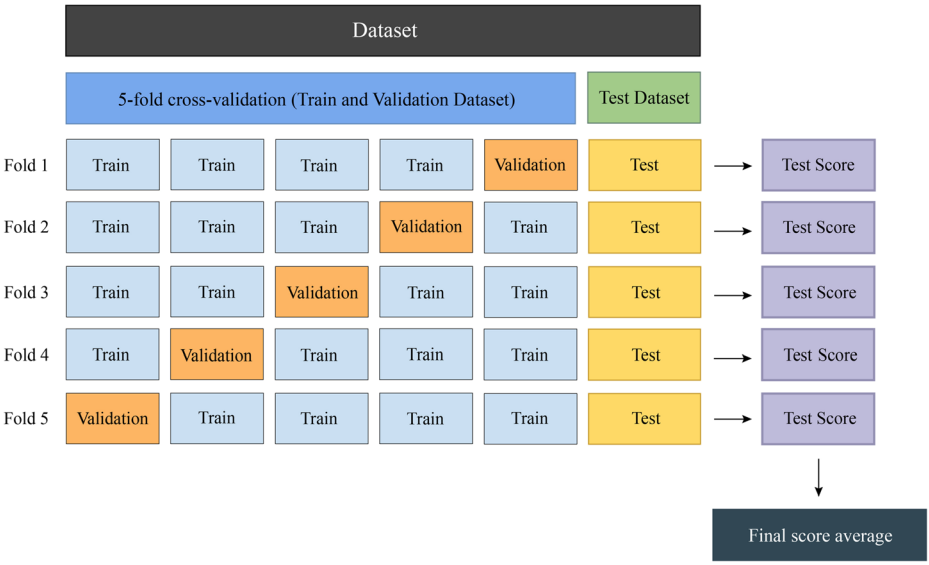


Fig. 7 5 fold cross validation

in the estimation made with a data set with normal and pneumonia tags; Determining that the data sample with TP: accurate pneumonia class label is pneumonia as a result of the estimation, TN: accurate class label is normal, predicted class label is normal, FP: The real class label is normal, but the prediction result is pneumonia, FN: Detection that the true class label is pneumonia, but the prediction result is normal.

According to Table 8, ACC: is the ratio of the number of correct predictions to the total number of samples; PPV: is the ratio of the prediction result to the number of positive samples correctly predicted to the number of all samples predicted positively. SN: ratio of the prediction result to the number of positive predicted samples to the number of negatively and positively predicted samples. F1: a hybrid performance proposed combining the PPV and SN metrics criterion.

4 Experimental results

In this section, the experimental results of the deep learning algorithms trained with the MediNet dataset, the experimental results of the classification studies with CT, CXR medical images (for example, before and after the pre-trained networks, applied hybrid methods, fine-

Table 7 Binary classification confusion matrix

		Predicted Class	
		Negative (**N**)	Positive (**P**)
Actual Class	Negative (**N**)	True Negative (**TN**)	False Positive (**FP**)
	Positive (**P**)	False Negative (**FN**)	True Positive (**TP**)

Table 8 The performance metrics of the binary classification

Metric	Formula
Accuracy	$ACC = \frac{TP+TN}{TP+TN+FP+FN}$
Positive Predictive Value (Precision)	$PPV = \frac{TP}{TP+FP}$
Sensitivity (Recall)	$SN = \frac{TP}{TP+FN}$
F_1-Score	$F_1 = 2\ x\ \frac{PPV\ x\ TPR}{PPV+TPR}$
Cohen's Kappa	$Kappa = \kappa = \frac{p_o - p_e}{1 - p_e};\quad p_o = ACC$
	$p_e = \frac{\sum_{c=1}^{N} (TP_c+FP_c)\ x\ (TP_c+FN_c)}{(TP+TN+FP+FN)^2}$

tuning studies, etc.) are given. Classification datasets with CT and CXR medical images were split into 9:1 (training, testing). The validation set consists of 20% of the training set.

4.1 Results of transfer learning study with MediNet data set

MediNet dataset consists of 4358 images. Deep neural networks and my hybrid methods were trained with AlexNet, VGG19-BN, InceptionV3, InceptionV3 + LSTM, InceptionV3 + GRU, DenseNet121, DenseNet121 + LSTM, DenseNet121 + GRU, RdiNet, ResNet101, EfficientNetB0, Nested-LSTM+CNN algorithms. During the training phase, the data was divided into 9:1 (train: validation). The training dataset consists of 3922; the validation dataset consists of 436 images. The size of the weight vectors of deep neural networks and the number of parameters are given in Table 9.

In Fig. 8, the artificial neural network for transfer learning in the AlexNet model was trained with a MediNet dataset with ten categories. Classification has been made for Chest/X-ray, Covid-19 and Diabetic Retinopathy images, which are a new dataset of the trained artificial neural network. The loss/validation loss and accuracy/validation accuracy metric results of the weight vectors of the lowest training loss values of the deep neural networks trained with the

Table 9 Deep Convolutional Neural Network Features

Model	Size	Total params
AlexNet	39.6 MB	3.461.002
VGG19-BN	231 MB	20.213.450
InceptionV3	250 MB	21,823,274
InceptionV3+LSTM	312 MB	27.162.538
InceptionV3+GRU	296.7 MB	25.823.530
DenseNet121	81 MB	7.047.754
DenseNet121+LSTM	123.5 MB	10.824.394
DenseNet121+GRU	114.5 MB	9.878.602
RdiNet	51.2 MB	4.442.038
ResNet101	488 MB	42.678.666
EfficientNetB0	46.7 MB	4.062.374
Nested-LSTM+CNN	47.5 MB	4.145.738

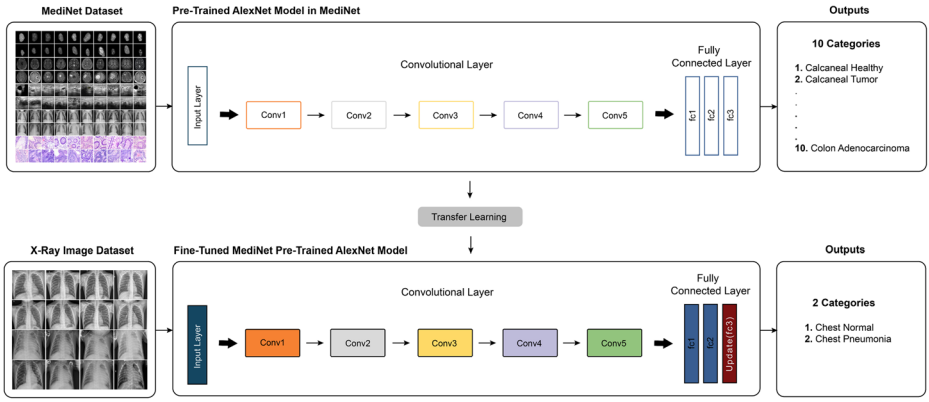


Fig. 8 Transfer Learning Flowchart for AlexNet model

MediNet dataset are given in Table 10. Loss graphs of MediNet training models are shown in Fig. 9.

4.2 Classification performance on chest X-ray images

In order to test the success of the transfer learning study conducted with the MediNet dataset in classification, first, a binary classification was made with the Chest X-ray dataset. The chest X-ray dataset consists of two categories as normal and as pneumonia. There are 1547 images, 683 in the normal images and 864 in the Pneumonia images. In the binary classification study conducted with the Chest dataset, 90% of the total data was used for training, while 10% was used for testing (Table 11). Applications were made with AlexNet, VGG19-BN, InceptionV3, DenseNet121, ResNet101, EfficientNetB0 and Nested-LSTM + CNN deep learning algorithms. In all studies performed with a Chest X-ray dataset (for example, w/o pre-training, w/o pre-training, fine-tuning) 50 epochs were used.

The success of the model in deep learning algorithms is directly proportional to the number of data used in the training dataset. Using the Chest dataset before pre-training, a different number of training sets/test sets were used with the AlexNet model. Training set / test set 90% / 10% accuracy when using data at 92.9%, when 70% / 30% data is used, the accuracy is 90.5% and when using 50% / 50% data, the accuracy was 88.8% (Table 12).

Classification study without transfer learning is presented for the Chest dataset. Before pre-training, the accuracy values were achieved in AlexNet, VGG19-BN, InceptionV3,

Table 10 Loss and ACC Values of Models

Model	Training Loss	Validation Loss	Training Accuracy	Validation Accuracy
AlexNet	0.0562	0.1631	0.9867	0.9586
VGG19-BN	0.1369	0.1587	0.9456	0.9494
InceptionV3	0.0568	0.1861	0.9811	0.9448
DenseNet121	0.1076	0.1912	0.9619	0.9448
RdiNet	0.0172	0.1054	0.9992	0.9609
ResNet101	0.0886	0.1885	0.9663	0.9448
EfficientNetB0	0.0785	0.2384	0.9711	0.9379
NestedLSTM+CNN	0.0227	0.0769	0.9990	0.9747

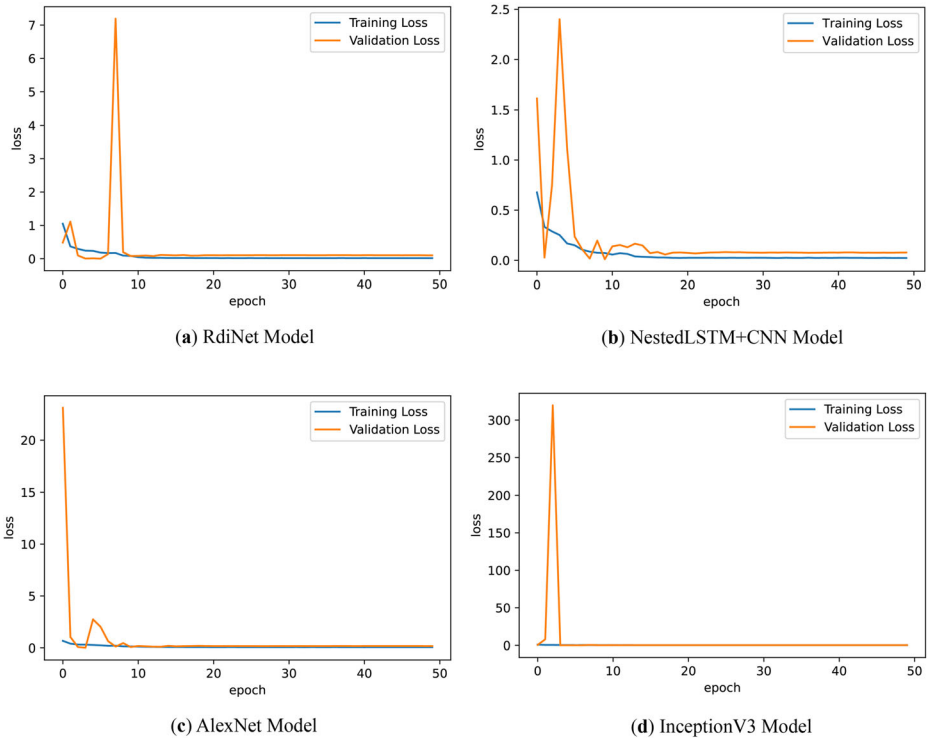


Fig. 9 Pre-training loss graphs for models: (**a**) RdiNet, (**b**) NestedLSTM+CNN, (**c**) AlexNet, (**d**) InceptionV3

DenseNet121, RdiNet, ResNet101, EfficientNetB0 and Nested-LSTM + CNN algorithms, respectively, while the accuracy values were 92.9%, 92.9%, 94.8%, 91.6%, 91.6%, 89.7%, 85.2%, 92.3%, respectively. The pre-training weight vectors with the MediNet dataset were used in the breast dataset with the transfer learning application. After pre-training, the accuracy values were 95.5%, 95.5%, 98.7%, 96.1%, 91.6%, 91.0%, 92.9%, respectively (Table 13).

Table 11 Image Distribution in Dataset

Category	Training Set	Test Set
Normal	614	69
Pneumonia	778	86
Total	1392	155

Table 12 Sample Training Set and Test Set Result for the AlexNet Model

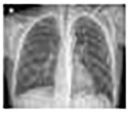
Sample Image	Training Set / Test Set	ACC	PPV	SN	Kappa
	90%–10%	0.9290	0.9213	0.9535	0.8557
	70%–30%	0.9054	0.8836	0.9529	0.8073
	50%–50%	0.8876	0.8404	0.9811	0.7690

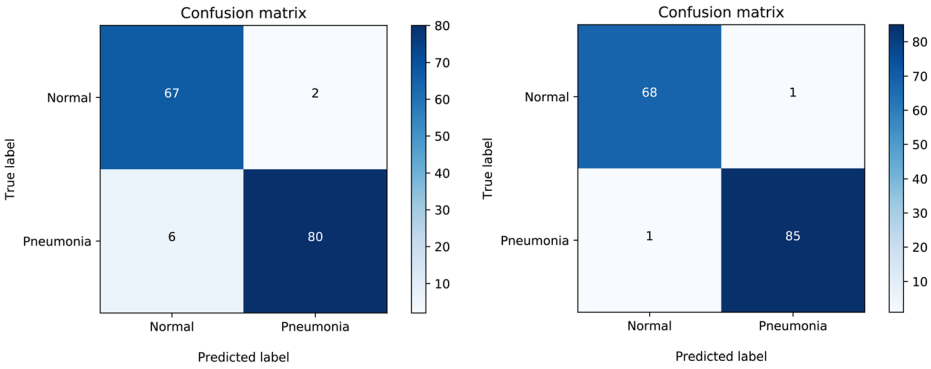
Table 13 Classification Performance Before and After Pre-Training

Network	Class	ACC	Kappa	AUC	PPV	SN	F_1
Random Initialization (w/o pre training)							
AlexNet	NORMAL	0.9290	0.8557	0.9260	0.9400	0.9000	0.9200
	PNEUMONIA				0.9200	0.9500	0.9400
VGG19-BN	NORMAL	0.9290	0.8557	0.9260	0.9400	0.9000	0.9200
	PNEUMONIA				0.9200	0.9500	0.9400
InceptionV3	NORMAL	0.9484	0.8961	0.9506	0.9200	0.9700	0.9400
	PNEUMONIA				0.9800	0.9300	0.9500
DenseNet121	NORMAL	0.9161	0.8309	0.9173	0.8900	0.9300	0.9100
	PNEUMONIA				0.9400	0.9100	0.9200
RdiNet	NORMAL	0.9161	0.8295	0.9172	0.8800	0.9200	0.9000
	PNEUMONIA				0.9400	0.9100	0.9300
ResNet101	NORMAL	0.8968	0.7916	0.8969	0.8700	0.9000	0.8900
	PNEUMONIA				0.9200	0.9000	0.9100
EfficientNetB0	NORMAL	0.8516	0.7000	0.8505	0.8300	0.8400	0.8300
	PNEUMONIA				0.8700	0.8600	0.8700
NestedLSTM + CNN	NORMAL	0.9226	0.8428	0.9202	0.9300	0.9000	0.9100
	PNEUMONIA				0.9200	0.9400	0.9300
MediNet-based model (w/ pre training)							
AlexNet	NORMAL	0.9548	0.9084	0.9536	0.9600	0.9400	0.9500
	PNEUMONIA				0.9500	0.9700	0.9600
VGG19-BN	NORMAL	0.9548	0.9087	0.9550	0.9400	0.9600	0.9500
	PNEUMONIA				0.9600	0.9500	0.9600
InceptionV3	NORMAL	0.9871	0.9739	0.9869	0.9900	0.9900	0.9900
	PNEUMONIA				0.9900	0.9900	0.9900
DenseNet121	NORMAL	0.9613	0.9216	0.9608	0.9600	0.9600	0.9600
	PNEUMONIA				0.9700	0.9700	0.9700
RdiNet	NORMAL	0.9290	0.8557	0.9304	0.9000	0.9400	0.9200
	PNEUMONIA				0.9500	0.9200	0.9400
ResNet101	NORMAL	0.9161	0.8290	0.9115	0.9400	0.8700	0.9000
	PNEUMONIA				0.9000	0.9500	0.9300
EfficientNetB0	NORMAL	0.9097	0.8161	0.9057	0.9200	0.8700	0.9000
	PNEUMONIA				0.9000	0.9400	0.9200
NestedLSTM + CNN	NORMAL	0.9290	0.8565	0.9289	0.9100	0.9300	0.9200
	PNEUMONIA				0.9400	0.9300	0.9400

InceptionV3 algorithm has been the most successful model in the chest-X-ray dataset with an accuracy rate of 94.8% before pre-training and 98.7% after pre-training.

The comparison of accuracy values in the pre-and post-transfer learning studies during the testing process of deep neural networks is given in Fig. 12. The Roc graphs of the algorithms are shown in Fig. 13.

In Fig. 10a, before transfer learning, the InceptionV3 architecture correctly diagnosed 147 data out of 155 test data and misdiagnosed 8 data. Accordingly, while the F1-score produced a value of 94% in the predictions made for 69 normal patients, the F1-score produced a value of 95% from the predictions of 86 pneumonia patients. Finally, 95% accuracy was obtained on 155 test data. In Fig. 10b, after transfer learning, the InceptionV3 architecture made correct diagnoses for 153 data in 155 test data and misdiagnosed for 2 data. Accordingly, while the f1 score produced a value of 99% in the predictions made for 69 normal patients, the f1 score produced a value of 99% from the predictions of 86 pneumonia patients. Finally, 99% accuracy was obtained on 155 test data. In Fig. 11, pre-trained and post-pre-trained loss graphs of the InceptionV3 algorithm are given.

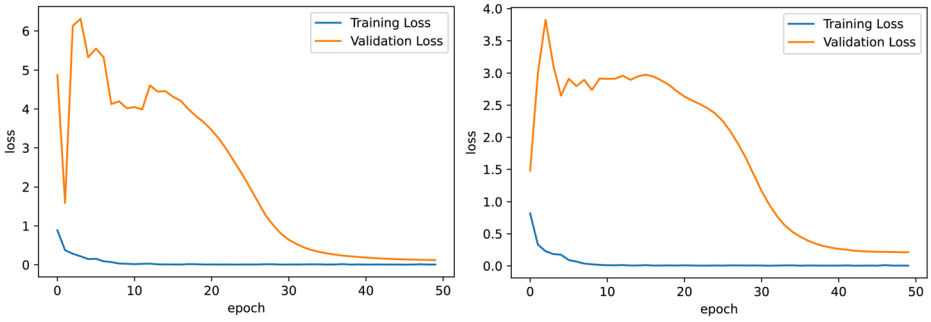


(**a**) Before pre-training (**b**) After pre-training

Fig. 10 Confusion matrix of InceptionV3 architecture before (a) and after (b) pre-training

According to Fig. 12, the Inception architecture before pre-trained was the most successful model with an accuracy of 94.8%, and after being pre-trained, the Inception architecture with an accuracy of 98.7%. EfficientNetB0 architecture had the lowest accuracy with 85.2% and 91.0% accuracy before and after pre-trained, respectively.

Retraining the entire training model is not always good in terms of process, time, and efficiency. Big dataset of data used in classification has a small size-dataset and the similarity of the data used in the classification to the dataset used in the training model, or fine-tuning work can be done according to different situations [13]. In this study, fine-tuning work was performed by training certain layers of the architectures used in the study after pre-training (Tables 14, 15, 16). In the fine-tuning work done in AlexNet architecture after pre-training in Table 14, the training made with the convolutional layer generally performed better than the training made with the fully connected layer (fcc). 68.4% accuracy was achieved when only the output layer of the previous training neural network was trained. When part of the classification layer (f6, f7, f8) is trained, the accuracy results are between 60% - and 68.4%. Accuracy values between 56.1% - and 91.0% were obtained in studies performed with the convolutional layers. In the Conv4-f8 run, the performance of the model turned out to be very low. In the Conv1-conv3-f8 study, the model showed the best performance.



(**a**) Before pre-training (**b**) After pre-training

Fig. 11 Loss plots of Inception V3 architecture before (a) and after (b) pre-training

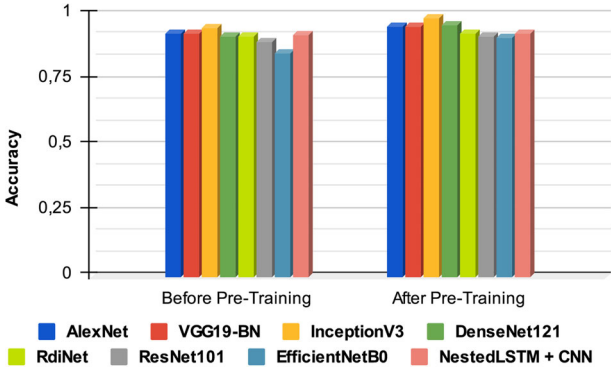


Fig. 12 Chest X-ray dataset pre and after pre-training accuracy values (Bar Chart)

In the first four results in Table 15, the learning level during the model's training is deficient. The model performed relatively well in the works conv5-conv3-f8 and conv5-conv2-f8. In the conv5-conv1-f8 layer, learning is again at a low level.

The first 30% of the InceptionV3, DenseNet121, RdiNet, ResNet101, and EfficientNetB0 models launched with MediNet weights were frozen. The experimental results of the study performed with the Nested-LSTM layer in the Nested-LSTM + CNN model and the results of the classification study using MediNet weights (unfrozen) are given in Table 16. The results of unfrozenda, ACC, PPV, SN metrics with DenseNet121 architecture were 96.1%, 96.5%,

Table 14 Fine Tuning for Alexnet Model After Pre -Training

Fine-tuning	ACC	PPV	SN	Kappa
Shallow-tuning: only f8	0.6839	0.6370	1.0000	0.3117
FT: f7-f8	0.6000	0.5811	1.0000	0.1113
FT: f6-f8	0.6581	0.6187	1.0000	0.2509
FT: conv5-f8	0.7032	0.6515	1.0000	0.3568
FT: conv4-f8	0.5613	0.5584	1.0000	0.0161
FT: conv2-conv5-f8	0.8452	0.8780	0.8372	0.6883
FT: conv2-conv4-f8	0.6710	1.0000	0.4070	0.3793
FT: conv2-conv3-f8	0.7290	0.6746	0.9884	0.4181
FT: conv1-conv5-f8	0.7677	0.7717	0.8256	0.5258
FT: conv1-conv4-f8	0.8903	0.9600	0.8372	0.7814
FT: conv1-conv3-f8	0.9097	0.9390	0.8953	0.8182

Table 15 Fine Tuning for VGG19-BN Model After Pre -Training

Fine-tuning	ACC	PPV	SN	Kappa
Shallow-tuning: only f8	0.5548	0.5548	1.0000	0.0000
FT: f7-f8	0.5548	0.5548	1.0000	0.0000
FT: f6-f8	0.5548	0.5548	1.0000	0.0000
FT: conv5-conv4-f8	0.5548	0.5548	1.0000	0.0000
FT: conv5-conv3-f8	0.7613	0.7168	0.9419	0.4973
FT: conv5-conv2-f8	0.6968	0.7241	0.7326	0.3853
FT: conv5-conv1-f8	0.5613	0.5584	1.0000	0.0161

Table 16 Deep Learning Models After Pre-Training; Fine Tuning Application for InceptionV3, DenseNet121, RdiNet, ResNet101, EfficientNetB0, Nested-LSTM+CNN (Only FCC Layer)

Model	Unfrozen			Frozen		
	ACC	PPV	SN	ACC	PPV	SN
InceptionV3	0.9871	0.9884	0.9884	0.9226	0.9535	0.9111
DenseNet121	0.9613	0.9651	0.9651	0.9419	0.9651	0.9326
RdiNet	0.9290	0.9535	0.9213	0.9032	0.9302	0.8989
ResNet101	0.9161	0.9011	0.9535	0.8387	1.0000	0.7748
EfficientNetB0	0.9097	0.9000	0.9419	0.5548	1.0000	0.5548
Nested-LSTM+CNN (only Nested-LSTM layer)	0.9290	0.9412	0.9302	0.7419	0.6855	0.9884

96.5%, respectively, and the results of ACC, PPV, SN metrics in the frozen study performed with DenseNet121 architecture were 94.2%, 96.5%, and 93.3%, respectively.

According to Fig. 13, Inception architecture before pre-trained was the most successful model with a roc value of 95.1%, and after pre-trained, Inception architecture with a roc value of 98.7%.

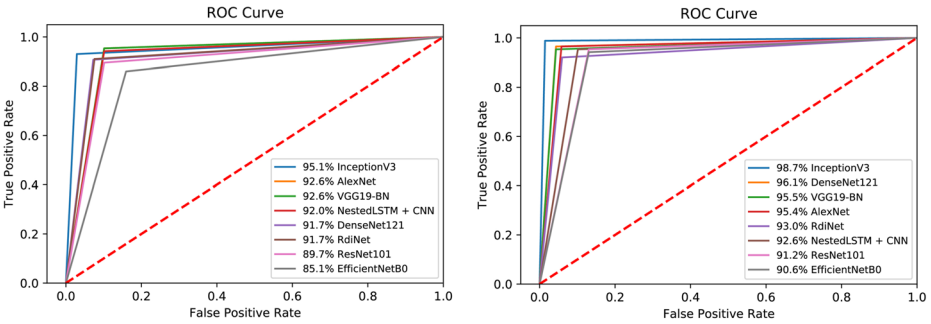
EfficientNetB0 architecture had the lowest roc value with 85.1% and 90.6% roc values before and after pre-trained.

4.2.1 Hybrid working with deep learning algorithms LSTM and GRU

In this method, feature extraction is done with InceptionV3 and DenseNet121 algorithms. LSTM and GRU algorithms are used for binary classification in the "Chest X-Ray Images dataset" dataset, the feature vectors obtained from the algorithms. The experimental results of the applied hybrid study are given in Table 17 in the form of pre-trained and pre-trained MediNet weights.

In this study, ACC, PPV, and SN values were 93.5%, 96.5%, and 92.2%, respectively, in the hybrid application made with the DenseNet121 model, which was started with random weights (before pre-training) and the GRU algorithm.

In the hybrid application made with the DenseNet121 model, which was started with MediNet weights (after pre-training) and the GRU algorithm, the ACC, PPV, and SN values were 95.5%, 96.5%, 95.4%, respectively.



(**a**) Before pre-training (**b**) After pre-training

Fig. 13 Comparison of the roc curves obtained from the binary classification study with the before (a) and after (b) pre-training deep neural networks

Table 17 Deep Learning Algorithms with LSTM and GRU

Model	Before			After		
	ACC	PPV	SN	ACC	PPV	SN
InceptionV3 + LSTM	0.9226	0.9419	0.9205	0.9419	0.9651	0.9326
InceptionV3 + GRU	0.9097	0.9419	0.9000	0.9226	0.9419	0.9205
DenseNet121 + LSTM	0.9355	0.9535	0.9318	0.9419	0.9419	0.9529
DenseNet121 + GRU	0.9355	0.9651	0.9222	0.9548	0.9651	0.9540

4.2.2 Hybrid working with deep learning algorithms

In this method, InceptionV3-RdiNet, InceptionV3-DenseNet121, InceptionV3-ResNet101, and DenseNet121-RdiNet hybrid methods were applied for binary classification in the "Chest X-Ray Images dataset" dataset. The experimental results of the applied hybrid study are given in Table 18 in the form of pre-trained and pre-trained MediNet weights.

In the hybrid study performed with the InceptionV3 and DenseNet121 model, which was started with random weights (before pre-training), the ACC, PPV, and SN values were 94.2%, 96.5%, 93.3%, respectively.

In the hybrid study with the InceptionV3 and DenseNet121 model initiated with MediNet weights (after pre-training), the ACC, PPV, and SN values were 95.5%, 96.5%, 95.4%, respectively.

Fold cross-validation performance The 5-fold cross-validation (CV) method was applied to evaluate the classification success of the transfer learning study proposed with the MediNet medical visual database. Experimental results of 5-fold CV application are given before pre-trained (Random Initialization (w/o pre-training)) and after pre-trained with MediNet weights (MediNet-based model (w/ pre-training)) (Table 19). The loss and accuracy graphs of the InceptionV3 model on 2-class using the Fold-1-5 "Chest X-Ray Images dataset" train and validation dataset are given in Fig. 14.

While the InceptionV3 model showed 93.7% (average) success before being pre-trained, it showed 93.9% (average) success after being pre-trained.

4.2.3 Hyperparameter Selection

In this section, experimental results of hyperparameters used in deep neural networks are given (Table 20).

Table 18 Hybrid Working with Deep Learning Algorithms

Model	Before			After		
	ACC	PPV	SN	ACC	PPV	SN
InceptionV3 + RdiNet	0.9290	0.9186	0.9518	0.9355	0.9535	0.9318
InceptionV3 + DenseNet121	0.9419	0.9651	0.9326	0.9548	0.9651	0.9540
InceptionV3 + AlexNet	0.9290	0.9186	0.9518	0.9484	0.9535	0.9535
DenseNet121 + RdiNet	0.9355	0.9535	0.9318	0.9548	0.9767	0.9438

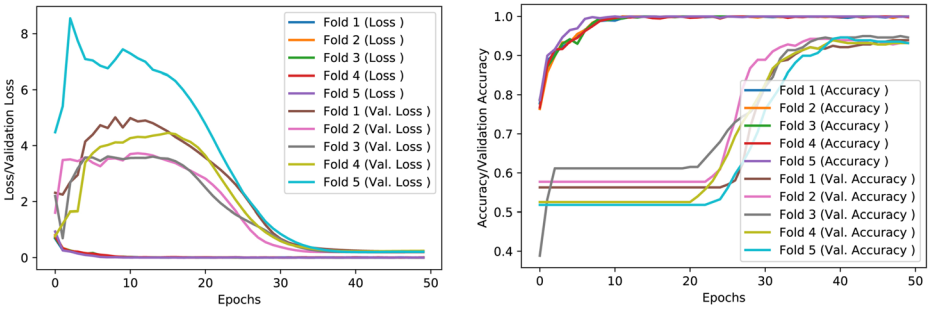
Table 19 Performance of the proposed InceptionV3 on 5-fold cross-validation using three and two class categories

Network	Transfer Learning	Fold	ACC	PPV	SN	Kappa
InceptionV3	Random Initialization (w/o pre training)	**Fold 1**	0.9484	0.9535	0.9535	0.8955
		Fold 2	0.9290	0.9213	0.9535	0.8557
		Fold 3	0.9355	0.9318	0.9535	0.8690
		Fold 4	0.9355	0.9318	0.9535	0.8690
		Fold 5	0.9355	0.9419	0.9419	0.8694
		Average	0.9368 ± 0.0063	0.9361±0.0109	0.9512±0.0047	0.8717±0.0130
InceptionV3	MediNet-based model (w/ pre training)	**Fold 1**	0.9355	0.9318	0.9535	0.8690
		Fold 2	0.9484	0.9432	0.9651	0.8952
		Fold 3	0.9419	0.9231	0.9767	0.8816
		Fold 4	0.9419	0.9326	0.9651	0.8819
		Fold 5	0.9290	0.9121	0.9651	0.8553
		Average	0.9394 ± 0.0066	0.9285 ± 0.0104	0.9651 ± 0.0074	0.8766 ± 0.0135

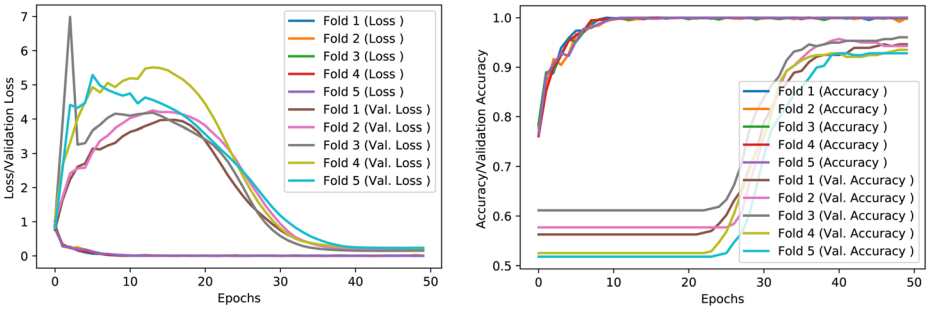
Epoc: 50, batch size: 64, Learning Rate Update: factor = 0.5, patience = 2 ve factor = 0.2, patience = 2, Optimizer: Adam metrics have been the most successful hyperparameters.

4.2.4 Convergence speed analysis

The proposed RdiNet architecture has been experimentally tested with different training test rates and different optimization algorithms and the loss values obtained during the test phase



(**a**) Before pre-training



(**b**) After pre-training

Fig. 14 The loss/validation loss and accuracy/validation accuracy graphs of the InceptionV3 model on 2-class using the Fold-1-5 (with before (a) and after (b) pre-training)

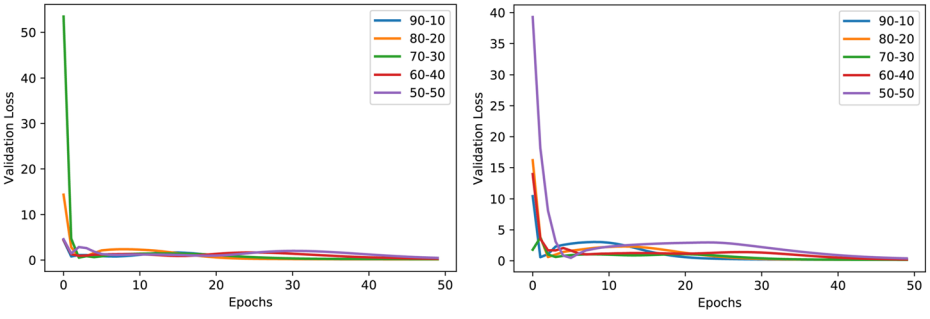
Table 20 Hyperparameter selection of the AlexNet model

Epoch	ACC	PPV	SN	Kappa
10 Epoch	0.6129	1.0000	0.5890	0.1427
30 Epoch	0.9226	0.9419	0.9205	0.8428
50 Epoch	0.9290	0.9213	0.9535	0.8557
100 Epoch	0.9290	0.9186	0.9518	0.8570
Batch Size				
32	0.9226	0.9070	0.9512	0.8442
64	0.9290	0.9213	0.9535	0.8557
128	0.8710	0.9767	0.8235	0.7327
256	0.8065	0.8488	0.8111	0.6059
512	0.7548	0.9186	0.7182	0.4860
Learning Rate Update (lr: 0.000001)				
factor=0.7, patience=10	0.9226	0.9186	0.9405	0.8437
factor=0.7, patience=5	0.9226	0.9535	0.9111	0.8424
factor=0.7, patience=2	0.9226	0.9070	0.9512	0.8442
factor=0.5, patience=10	0.8903	0.8721	0.9259	0.7796
factor=0.5, patience=5	0.9161	0.9186	0.9294	0.8305
factor=0.5, patience=2	0.9290	0.9213	0.9535	0.8557
factor=0.2, patience=10	0.9097	0.8837	0.9500	0.8187
factor=0.2, patience=5	0.9097	0.9302	0.9091	0.8166
factor=0.2, patience=2	0.9290	0.9186	0.9518	0.8570
Optimizer				
Adam	**0.9290**	**0.9213**	**0.9535**	**0.8557**
RMSprop	0.9226	0.9651	0.9022	0.8419
AdaDelta	0.7935	0.9884	0.7328	0.5633
AdaGrad	0.9161	0.9535	0.9011	0.8290
SGD	0.9226	0.9419	0.9205	0.8428

(validation loss values) during the training and the convergence analysis of the RdiNet model to the optimum loss point are given in Figs. 15 and 16. In Table 21, the experimental results of the analysis study with different training test rates and different optimization algorithms are given.

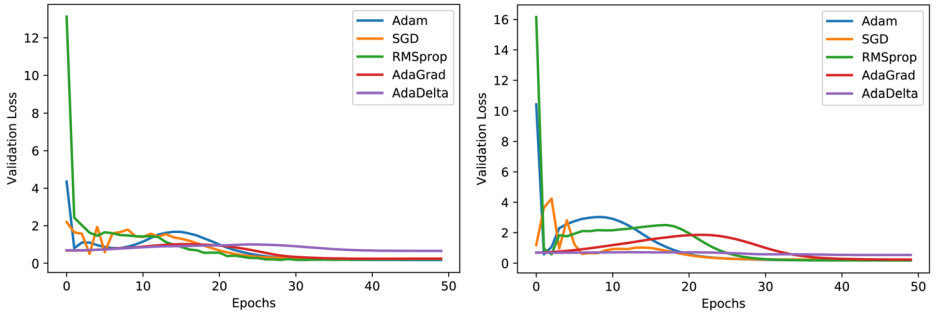
In Fig. 15, a comparison of the convergence speed with the MediNet dataset before and after pre-trained in terms of train/test data numbers in the RdiNet model and the Chest Dataset is given.

Model Adam optimization algorithm was trained with Chest Dataset for 50 cycles using 64 batch size values. While converging towards the optimum loss value (the loss value is 0) in every five studies before transfer learning, In the study where the training test ratio was 70/30,



(**a**) Before pre-training (**b**) After pre-training

Fig. 15 Convergence speed comparison in terms of number data on Chest Dataset of RdiNet model (with the before (a) and after (b) pre-training)



(**a**) Before pre-training (**b**) After pre-training

Fig. 16 Comparison of the convergence of Adam, RMSProp, AdaDelta, AdaGrad and SGD on binary classification problems (with before (a) and after (b) pre-training)

the RdiNet model converged much faster when compared to other training/test ratios. However, during the training, oscillations towards the optimum were observed in the RdiNet model. At the end of the training, the RdiNet model converged towards the exact optimum in studies where the training test ratio was 90/10 and 80/20. After transfer learning, it converged towards the optimum loss value in every five studies; In the study where the training test ratio was 50/50, the RdiNet model converged much faster when compared to other training/test ratios. However, during the training, oscillations towards the optimum were observed in the RdiNet model.

In Fig. 16, the convergence speed comparison of the RdiNet model and the MediNet dataset in the Chest Dataset and the optimization algorithms before and after pre-trained is given. The model was trained with the Chest Dataset for 50 cycles using the training test ratio with 90/10, 64 batch size values. Before transfer learning, Adam, RMSProp, AdaGrad, and SGD converged towards the optimum loss (the loss value being 0); In the AdaDelta algorithm, oscillations towards the optimum were observed during the training. The RdiNet model trained with the RMSprop algorithm before transfer learning converged much faster compared to other optimization algorithms. At the end of the training, the RdiNet model trained on Adam, RMSProp, and SGD algorithms converged towards the same optimum.

Table 21 RdiNet modeli hiperparamters (Rndm: Random Initialization (w/o pre training), TL: MediNet-based model (w/ pre training))

Epoch	**Optimizar**	**ACC**		**PPV**		**SN**		**Kappa**	
		Rndm	TL	Rndm	TL	Rndm	TL	Rndm	TL
90–10	Adam	0.9161	0.9290	0.9419	0.9535	0.9101	0.9213	0.8295	0.8557
80–20	Adam	0.9129	0.9258	0.9101	0.9270	0.9364	0.9429	0.8228	0.8487
70–30	Adam	0.8989	0.9140	0.9059	0.9098	0.9094	0.9317	0.7960	0.8268
60–40	Adam	0.8401	0.8901	0.9622	0.9593	0.7938	0.8594	0.6672	0.7742
50–50	Adam	0.7442	0.7842	0.9858	0.9953	0.6852	0.7189	0.4585	0.5460
Optimizer	Train/Test	**ACC**		**PPV**		**SN**		**Kappa**	
		Rndm	TL	Rndm	TL	Rndm	TL	Rndm	TL
Adam	90–10	0.9161	0.9290	0.9419	0.9535	0.9101	0.9213	0.8295	0.8557
RMSprop	90–10	0.9097	0.9226	0.9419	0.9419	0.9000	0.9205	0.8161	0.8428
AdaDelta	90–10	0.6774	0.7484	0.8140	0.7791	0.6731	0.7701	0.3297	0.4899
AdaGrad	90–10	0.9097	0.9161	0.9419	0.9535	0.9000	0.9011	0.8161	0.8290
SGD	90–10	0.8903	0.9032	0.9302	0.9186	0.8791	0.9080	0.7764	0.8038

After transfer learning, Adam, RMSProp, AdaGrad, and SGD converge towards the optimum loss (the loss value is 0); In the AdaDelta algorithm, oscillations towards the optimum were observed during and at the end of the training. After transfer learning, the RdiNet model trained with the RMSprop algorithm converged much faster compared to other optimization algorithms. At the end of the training, the RdiNet model trained on Adam, RMSProp, and SGD algorithms converged towards the same optimum.

In addition, it was observed that the RdiNet model, which was trained with the AdaGrad algorithm before and after the transfer, converged towards the optimum at the end of the training.

4.3 Classification performance on COVID19-CT dataset

In order to test the success of the transfer learning study conducted with the MediNet dataset in classification, a binary classification was made with another dataset, the Covid-19 dataset. Covid-19 dataset consists of two categories, Covid-19 and nonCovid-19.

In a binary classification study, 90% of the total data is used for training, while 10% is used for testing (Table 22). Applications were done with AlexNet, VGG19-BN, InceptionV3, DenseNet121, RdiNet, ResNet101, EfficientNetB0 and Nested-LSTM + CNN deep learning algorithms. In all applications with Covid-19 datasets (for example, w / o pre-training, w / pre-training, fine-tuning), 100 epochs were used.

In the pre-training study, AlexNet, VGG19-BN, InceptionV3, DenseNet121, RdiNet, ResNet101, EfficientNetB0 and Nested-LSTM + CNN algorithms achieved 74.7%, 77.3%, 80.0%, 88.0%, 77.3%, 58.7%, 66.7%, 78.7%, respectively, after pre-training, the accuracy values were again 80.0%, 78.7%, 89.3%, 92.0%, 82.7%, 77.3%, 78.7%, 80.0%, respectively (Table 23). DenseNet121 algorithm was the most successful model in the covid-19 study with an accuracy rate of 88.0% before pre-training and 92.0% after pre-training. The comparison of accuracy values in the pre-and post-transfer learning studies during the testing process of deep neural networks is given in Fig. 19. The Roc graphs of the algorithms are shown in Fig. 20.

In Fig. 17a, before transfer learning, the DenseNet121 architecture correctly diagnosed 66 data out of 75 test data and misdiagnosed 9 data. Accordingly, while the F1-score produced a value of 90% in the predictions made for 44 Non-COVID-19 patients, the F1-score produced a value of 85% from the predictions of 31 COVID-19 patients. Finally, 88% accuracy was obtained on 75 test data. In Fig. 17b, after transfer learning, the DenseNet121 architecture made correct diagnoses for 69 data in 75 test data and misdiagnosed for 6 data. Accordingly, while the f1 score produced a value of 93% in the predictions made for 44 Non-COVID-19 patients, the f1 score produced a value of 90% from the predictions of 31 COVID-19 patients. Finally, 92% accuracy was obtained on 75 test data. In Fig. 18, the pre-trained and post-pre-trained loss graphs of the DenseNet121 algorithm are given.

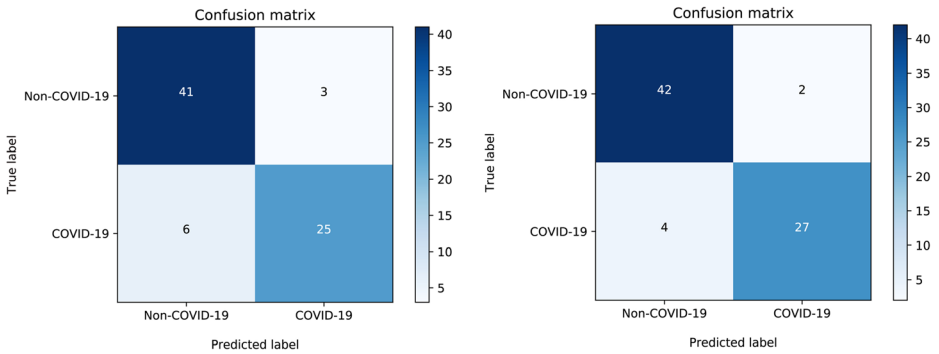
In Fig. 19, the DenseNet121 architecture, before pre-trained, was the most successful model with an accuracy of 88.0%, and after pre-trained, the DenseNet121 architecture was with an

Table 22 Image Distribution in The Dataset

Category	Training Set	Test Set
NonCOVID	353	44
COVID	318	31
Total	671	75

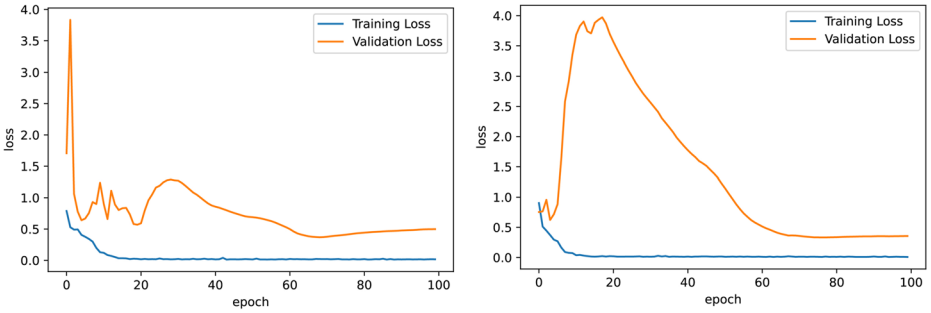
Table 23 Classification Performance Before and After Pre-Training

Network	Class	ACC	Kappa	AUC	PPV	SN	F_1
Random Initialization (w/o pre training)							
AlexNet	Non-COVID-19	0.7467	0.4751	0.7364	0.7800	0.8000	0.7900
	COVID-19				0.7000	0.6800	0.6900
VGG19-BN	Non-COVID-19	0.7733	0.5392	0.7735	0.8300	0.7700	0.8000
	COVID-19				0.7100	0.7700	0.7400
InceptionV3	Non-COVID-19	0.8000	0.5816	0.7867	0.8100	0.8600	0.8400
	COVID-19				0.7900	0.7100	0.7500
DenseNet121	Non-COVID-19	0.8800	0.7490	0.8691	0.8700	0.9300	0.9000
	COVID-19				0.8900	0.8100	0.8500
RdiNet	Non-COVID-19	0.7733	0.5212	0.7724	0.8600	0.7800	0.8200
	COVID-19				0.6500	0.7700	0.7000
ResNet101	Non-COVID-19	0.5867	0.2185	0.6191	0.7600	0.4300	0.5500
	COVID-19				0.5000	0.8100	0.6200
EfficientNetB0	Non-COVID-19	0.6667	0.2747	0.6301	0.6700	0.8400	0.7500
	COVID-19				0.6500	0.4200	0.5100
NestedLSTM + CNN	Non-COVID-19	0.7867	0.5601	0.7801	0.8200	0.8200	0.8200
	COVID-19				0.7400	0.7400	0.7400
MediNet-based model (w/ pre training)							
AlexNet	Non-COVID-19	0.8000	0.5816	0.7867	0.8100	0.8600	0.8400
	COVID-19				0.7900	0.7100	0.7500
VGG19-BN	Non-COVID-19	0.7867	0.5559	0.7753	0.8000	0.8400	0.8200
	COVID-19				0.7600	0.7100	0.7300
InceptionV3	Non-COVID-19	0.8933	0.7821	0.8948	0.9300	0.8900	0.9100
	COVID-19				0.8500	0.9000	0.8800
DenseNet121	Non-COVID-19	0.9200	0.8335	0.9128	0.9100	0.9500	0.9300
	COVID-19				0.9300	0.8700	0.9000
RdiNet	Non-COVID-19	0.8267	0.6339	0.8312	0.9100	0.8200	0.8600
	COVID-19				0.7100	0.8500	0.7700
ResNet101	Non-COVID-19	0.7733	0.5477	0.7830	0.8600	0.7300	0.7900
	COVID-19				0.6800	0.8400	0.7500
EfficientNetB0	Non-COVID-19	0.7867	0.5381	0.7562	0.7600	0.9300	0.8400
	COVID-19				0.8600	0.5800	0.6900
NestedLSTM + CNN	Non-COVID-19	0.8000	0.5934	0.8010	0.8500	0.8000	0.8200
	COVID-19				0.7400	0.8100	0.7700



(**a**) Before pre-training (**b**) After pre-training

Fig. 17 Confusion matrix of DenseNet121 architecture before (a) and after (b) pre-training



(a) Before pre-training (b) After pre-training

Fig. 18 Loss plots of DenseNet121 architecture before (a) and after (b) pre-training

accuracy of 92.0%. On the other hand, the ResNet101 architecture had the lowest accuracy with 58.7% and 77.3% accuracy before and after pre-trained. MediNet weights were started by freezing the first 30% of AlexNet, VGG19-BN, InceptionV3, DenseNet121, RdiNet, ResNet101, and EfficientNetB0 models. The results of the classification study (unfrozen) are given in Table 24. In the unfrozen study with AlexNet architecture, ACC, PPV, and SN metrics were 80.0%, 78.6%, and 71.0%, respectively. In the frozen study performed with

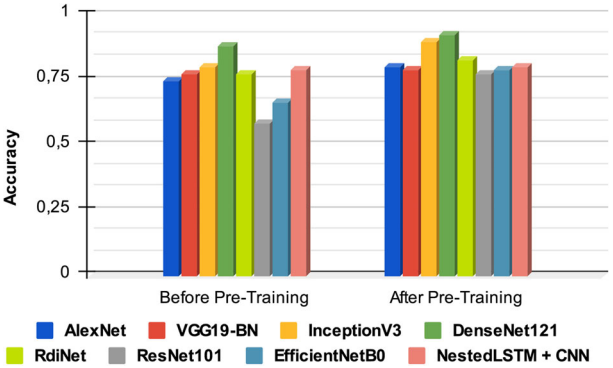


Fig. 19 Covid-19 dataset pre and after pre-training accuracy values (Bar Chart)

Table 24 Deep Learning Models After Pre-Training; Fine Tuning Application for AlexNet, VGG1—BN, InceptionV3, DenseNet121, RdiNet, ResNet101, EfficientNetB0, Nested-LSTM+CNN (Only FCC Layer)

Model	Unfrozen			Frozen		
	ACC	PPV	SN	ACC	PPV	SN
AlexNet	0.8000	0.7857	0.7097	0.7867	0.7742	0.7273
VGG19-BN	0.7867	0.7586	0.7097	0.6933	0.6452	0.6250
Inceptionv3	0.8933	0.8485	0.9032	0.5467	0.3226	0.4348
Densenet121	0.9200	0.9310	0.8710	0.8267	0.8065	0.7812
RdiNet	0.8267	0.7097	0.8462	0.8133	0.8065	0.7576
Resnet101	0.7733	0.6842	0.8387	0.6933	0.6129	0.6333
EfficientNetB0	0.7867	0.8571	0.5806	0.4133	1.0000	0.4133
Nested-LSTM+CNN (only Nested-LSTM layer)	0.8000	0.7353	0.8065	0.6400	0.5909	0.4194

AlexNet architecture, ACC, PPV, and SN metrics were 78.7%, 77.4%, and 72.7%, respectively.

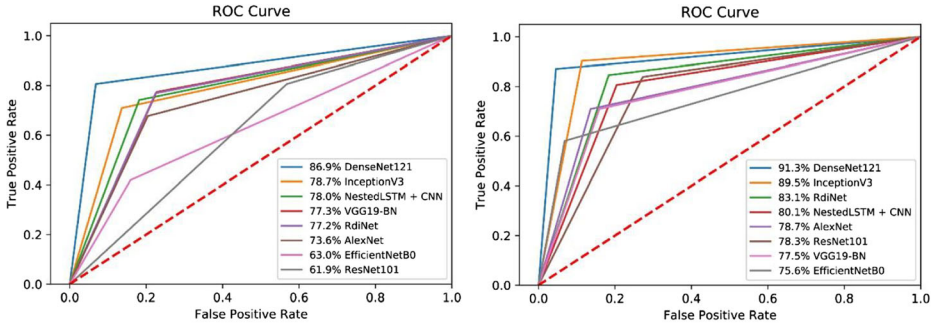
According to Fig. 20, the DenseNet121 architecture before pre-trained was the most successful model with a roc AUC value of 86.9%, and after pre-trained, the DenseNet121 architecture with a ROC- AUC value of 91.3%. The DenseNet121 architecture after pre-trained was the least unsuccessful model, with a ROC-AUC of 61.9% and a post-pre-trained EfficientNetB0 architecture of 75.6%.

4.4 Classification performance on diabetic retinopathy dataset

To test the transfer learning study's success with the MediNet dataset in classification, a binary classification was made with the last dataset, the Diabetic Retinopathy dataset. The Diabetic Retinopathy dataset consists of two categories: no symptoms and symptoms. There are 915 images, 320 in the No symptoms dataset and 595 in the Symptoms dataset. In the binary classification study conducted with the Diabetic Retinopathy dataset, 90% of the total data was used for training, while 10% was used for testing (Table 25). Applications were made with AlexNet, VGG19-BN, InceptionV3, DenseNet121, RdiNet, ResNet101, EfficientNetB0 and Nested-LSTM + CNN deep learning algorithms. 100 epochs were used in all processes (w/o pre-training, w/o pre-training, fine-tuning) performed with the Diabetic Retinopathy dataset.

Before pre-training, AlexNet, VGG19-BN, InceptionV3, DenseNet121, RdiNet, ResNet101, EfficientNetB0 and Nested-LSTM + CNN algorithms achieved 72.8%, 73.9%, 66.3%, 75.0%, 78.3%, 75.0%, 72.8%, 79.4%, respectively, after pre-training, the values were 78.3%, 75.0%, 70.7%, 77.2%, 79.4%, 77.2%, 73.9%, 81.5%, respectively (Table 26). NestedLSTM + CNN algorithm has been the most successful diabetic retinopathy study with an accuracy rate of 79.4% before pre-training and 81.5% after. The comparison of accuracy values in the pre-and post-transfer learning studies during the testing process of deep neural networks is given in Fig. 23. The Roc graphs of the algorithms are shown in Fig. 24.

In Fig. 21, before transfer learning, the Nested-LSTM + CNN architecture correctly diagnosed 73 data out of 92 test data and misdiagnosed 19 data. Accordingly, while the F1-score produced a value of 71% in the predictions made for 31 No symptoms patients, the F1-score produced a value of 84% from the predictions of 61 Symptoms patients. Finally, 79.3% accuracy was obtained on 91 test data. In Fig. 21b, after transfer learning, the Nested-LSTM + CNN architecture made correct diagnoses for 75 data in 92 test data and misdiagnosed for 17



(a) Before pre-training (b) After pre-training

Fig. 20 Comparison of the roc curves obtained from the binary classification study with the before (a) and after (b) pre-training deep neural networks

Table 25 Dataset Distribution Representation

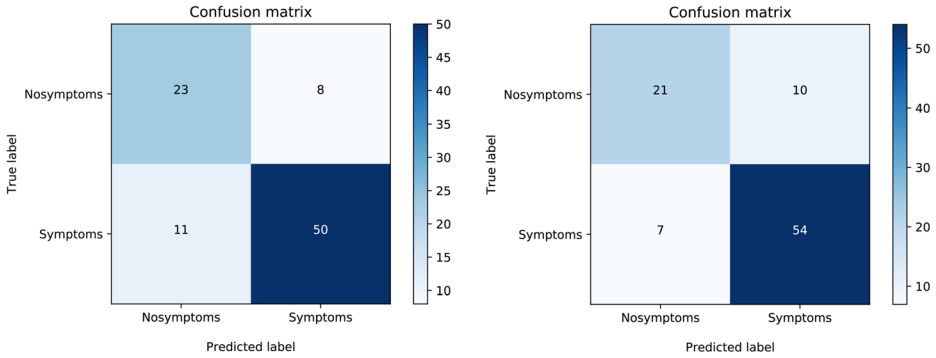
Category	Training Set	Test Set
Nosymptoms	289	31
Symptoms	534	61
Total	823	92

data. Accordingly, while the f1 score produced a value of 71% in the predictions made for 31 No symptoms patients, the f1 score produced a value of 96% from the predictions of 61 Symptoms patients. Finally, 81.5% accuracy was obtained on 91 test data. In Fig. 22, before pre-trained and post-pre-trained loss, the Nested-LSTM + CNN algorithm graphs are given.

In Fig. 23, the NestedLSTM + CNN architecture before pre-trained was the most successful model, with an accuracy value of 79.4%. After pre-trained the NestedLSTM + CNN architecture with an accuracy of 81.5%. InceptionV3 architecture had the lowest accuracy with 66.0% and 70.6% accuracy before and after pre-trained, respectively. MediNet weights were

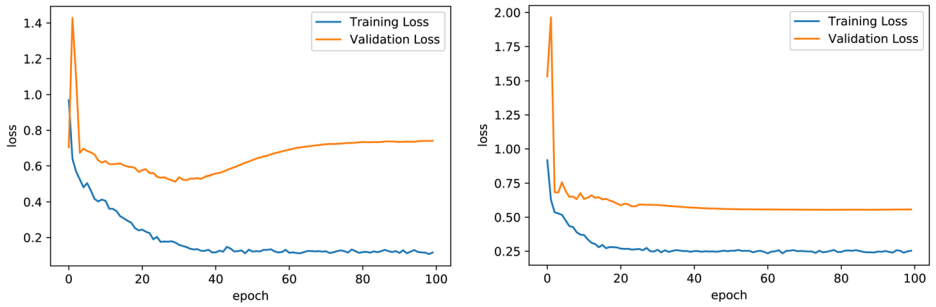
Table 26 Classification Performance Before and After Pre-Training

Network	Class	ACC	Kappa	AUC	PPV	SN	F_1
Random Initialization (w/o pre training)							
AlexNet	Nosymptoms	0.7283	0.4151	0.7158	0.5800	0.6800	0.6300
	Symptoms				0.8200	0.7500	0.7900
VGG19-BN	Nosymptoms	0.7391	0.3659	0.6684	0.6700	0.4500	0.5400
	Symptoms				0.7600	0.8900	0.8200
InceptionV3	Nosymptoms	0.6630	0.0000	0.5000	0.0000	0.0000	0.0000
	Symptoms				0.6600	1.0000	0.8000
DenseNet121	Nosymptoms	0.7500	0.4076	0.6925	0.6700	0.5200	0.5800
	Symptoms				0.7800	0.8700	0.8200
RdiNet	Nosymptoms	0.7826	0.4805	0.7681	0.5500	0.7400	0.6300
	Symptoms				0.9000	0.8000	0.8500
ResNet101	Nosymptoms	0.7500	0.3870	0.6766	0.7000	0.4500	0.5500
	Symptoms				0.7600	0.9000	0.8300
EfficientNetB0	Nosymptoms	0.7283	0.2971	0.6285	0.7100	0.3200	0.4400
	Symptoms				0.7300	0.9300	0.8200
NestedLSTM + CNN	Nosymptoms	0.7935	0.5486	0.7808	0.6800	0.7400	0.7100
	Symptoms				0.8600	0.8200	0.8400
MediNet-based model (w/ pre training)							
AlexNet	Nosymptoms	0.7826	0.5135	0.7567	0.6800	0.6800	0.6800
	Symptoms				0.8400	0.8400	0.8400
VGG19-BN	Nosymptoms	0.7500	0.4699	0.7480	0.6100	0.7400	0.6700
	Symptoms				0.8500	0.7500	0.8000
InceptionV3	Nosymptoms	0.7065	0.2927	0.6359	0.5900	0.4200	0.4900
	Symptoms				0.7400	0.8500	0.7900
DenseNet121	Nosymptoms	0.7717	0.4591	0.7168	0.7100	0.5500	0.6200
	Symptoms				0.7900	0.8900	0.8400
RdiNet	Nosymptoms	0.7935	0.5265	0.7712	0.6500	0.7100	0.6800
	Symptoms				0.8700	0.8300	0.8500
ResNet101	Nosymptoms	0.7717	0.4591	0.7168	0.7100	0.5500	0.6200
	Symptoms				0.7900	0.8900	0.8400
EfficientNetB0	Nosymptoms	0.7391	0.3548	0.6605	0.6800	0.4200	0.5200
	Symptoms				0.7500	0.9000	0.8200
NestedLSTM + CNN	Nosymptoms	0.8152	0.5764	0.7813	0.7500	0.6800	0.7100
	Symptoms				0.8400	0.8900	0.8600



(**a**) Before pre-training (**b**) After pre-training

Fig. 21 Confusion matrix of Nested-LSTM + CNN architecture before (a) and after (b) pre-training



(**a**) Before pre-training (**b**) After pre-training

Fig. 22 Loss plots of Nested-LSTM + CNN architecture before (a) and after (b) pre-training

started by freezing the first 30% of AlexNet, VGG19-BN, InceptionV3, DenseNet121, RdiNet, ResNet101, and EfficientNetB0 models, and the experimental results of the study with the Nested-LSTM layer in the Nested-LSTM + CNN model and using MediNet weights. The results of the classification study (unfrozen) are given in Table 27. In the unfrozen with

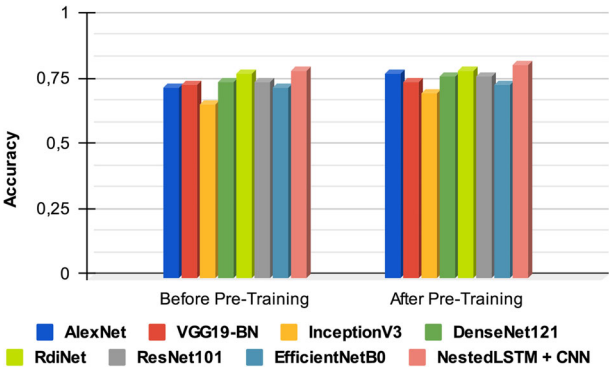


Fig. 23 Accuracy values of Diabetic Retinopathy dataset before and after pre-training (Bar Chart)

Table 27 Deep Learning Models After Pre-Training; Fine Tuning Application for AlexNet, VGG1—BN, InceptionV3, DenseNet121, RdiNet, ResNet101, EfficientNetB0, Nested-LSTM+CNN (Only FCC Layer)

Model	Unfrozen			Frozen		
	ACC	PPV	SN	ACC	PPV	SN
AlexNet	0.7826	0.8361	0.8361	0.8043	0.9016	0.8209
VGG19-BN	0.7500	0.8519	0.7541	0.7283	0.8852	0.7500
Inceptionv3	0.7065	0.7429	0.8525	0.6630	1.0000	0.6630
Densenet121	0.7717	0.7941	0.8852	0.7500	0.8689	0.7794
RdiNet	0.7935	0.8689	0.8281	0.8043	0.9180	0.8116
Resnet101	0.7717	0.7941	0.8852	0.6630	1.0000	0.6630
EfficientNetB0	0.7391	0.7534	0.9016	0.6630	1.0000	0.6630
Nested-LSTM+CNN (only Nested-LSTM layer)	0.8152	0.8438	0.8852	0.6630	0.6630	1.0000

RdiNet architecture, ACC, PPV, and SN metrics were 79.3%, 86.9%, and 82.8%, respectively. In the frozen performed with RdiNet architecture, ACC, PPV, and SN metrics were 80.4%, 91.8%, and 81.2%, respectively.

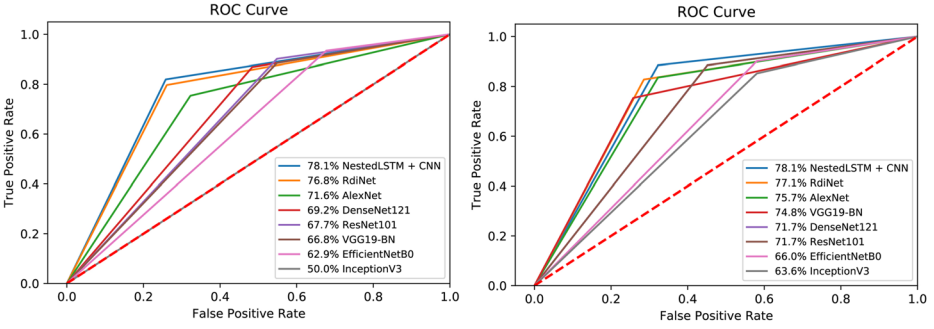
According to Fig. 24, NestedLSTM + CNN architecture was the most successful model, with a ROC-AUC value of 78.1% before pre-trained and 78.1% after pre-trained NestedLSTM + CNN architecture.

The InceptionV3 architecture after pre-trained was the most unsuccessful model, with a ROC-AUC of 50.0% and a ROC value of 63.6% after pre-trained.

5 Limitations, future research directions

This study has some limitations:

(i) Due to the lack of technical equipment, the MediNet data set consists of a limited number of data samples and ten categories. To improve the classification success of the transfer learning method proposed with MediNet, the number of data samples and the number of categories in the MediNet dataset can be increased.



(**a**) Before pre-training (**b**) After pre-training

Fig. 24 Comparison of the roc curves obtained from the binary classification study with the before (a) and after (b) pre-training deep neural networks

(ii) The medical images used in the MediNet dataset and the medical images used in the classification studies consist of raw data. Here, preprocessing techniques can be applied to separate the heart, text areas, and bone regions from the lung images in the CXR dataset. As a result, the performance of the proposed transfer learning application can be improved by preprocessing the medical dataset samples.

(iii) The proposed transfer learning method has been developed to be applied only to the binary classification problem. Here, the performance of the transfer learning method proposed by expanding the MediNet visual database can be investigated in binary and multiple classification problems.

(iv) The RdiNet model proposed in the study was successful according to the experimental results applied to the binary classification problem, but its effectiveness against modern architectures remained low. Here, the depth of the architecture and the use of additional feature vectors can be increased to increase efficiency in the classification applications of the architecture.

6 Discussion

Deep neural networks from artificial intelligence technologies are essential in detecting various diseases that cause severe complications in humans, such as cancer, lung diseases, and Alzheimer's [12].

Extensive data are generally needed for deep neural networks with complex structures and high parameters to be successful [43]. However, obtaining labeled data is challenging in many areas, such as image processing. Deep learning algorithms trained from scratch face current problems such as overfitting, underfitting, and gradient disappearance. Here, (i) data augmentation method can be applied to solve the problems experienced [3], (ii) deep neural networks can be developed with residual networks or additional feature vectors (iii) transfer learning technique can be applied [4]. Transfer learning practice between solutions draws attention as an essential solution technique. The transfer learning process is a method that generally focuses on the reuse of neural networks that have been previously trained with large amounts of datasets. ImageNet dataset is frequently used in the transfer learning process [5]. The literature has observed that transfer learning studies proposed with ImageNet generally produce high performance [5]. The ImageNet dataset consists of heterogeneous data samples. Therefore, in the transfer learning process proposed with ImageNet in image processing, models focus on learning with heterogeneous data. In medical image processing applications, the disease diagnosis/diagnosis process is one of the applications that require high precision. Therefore, transfer learning applications made with datasets of very different categories, such as ImageNet, can be disadvantageous in the disease detection process. A transfer learning application is proposed with the MediNet dataset, which consists of only medical images from the main contributions of this study. Therefore, the study focuses only on the performance of the MediNet dataset, which consists of medical images, in the transfer learning process. The proposed transfer learning technique is given a comparative analysis before and after pre-trained with AlexNet, VGG19-BN, InceptionV3, DenseNet121, RdiNet, ResNet101, EfficientNetB0, Nested LSTM + CNN models from deep neural networks. In the light of experimental results, it has been observed that the proposed method produces successful results. However, the proposed database needs to be developed with big data in different categories. Therefore, another contribution of this work, deep neural

network-based RdiNet architecture is proposed. The RdiNet architecture is a practical network that data scientists can benefit from. It can provide an alternative view to the gradient disappearance problem in deep neural networks with residual networks and additional feature vectors.

In this research, we present MediNet, a visual database consisting of different medical images for transfer learning applications. MediNet dataset was trained one by one with deep learning algorithms, which are AlexNet, VGG19-BN, InceptionV3, DenseNet121, ResNet101, EfficientNetB0, Nested LSTM + CNN, RdiNet models. Classified medical big data in real life is almost non-existent. With the MediNet project, we presented a new visual database consisting of medical data used in transfer learning research. With the MediNet study, we achieved good results with the small dataset. As demonstrated by his experimental results, it is shown to achieve more successful results using transfer learning technic in the medical field. Two significant obstacles that limited us in this study; are the small-data size and lack of technical equipment (the study was done with a personal computer).

7 Conclusion and future work

This research proposes a transfer learning method with a MediNet visual dataset consisting of different medical images. The proposed transfer learning method has been applied in detecting pneumonia, COVID-19, and eye diabetes from datasets consisting of CXR and CT images. AlexNet, VGG19-BN, InceptionV3, DenseNet121, ResNet101, EfficientNetB0, Nested LSTM + CNN architectures, and hybrid methods are proposed to be used together with the proposed CNN-based RdiNet deep neural network in the transfer learning process and solving the classification problem. Within the scope of the study, firstly, deep neural networks were trained with the MediNet dataset. In the second stage, the trained neural networks were applied to detect pneumonia, COVID-19, and eye-diabetes from datasets consisting of CXR and CT images. In the disease detection process, hybrid methods were used using deep neural networks and LSTM, GRU, and deep neural networks within the scope of the study. Experimental results are given comparatively before and after transfer learning. Our research findings; showed that the transfer learning application proposed with MediNet was successful in the disease detection process. In the first stage of the study, NestedLSTM+CNN and RdiNet architectures trained with the MediNet dataset were the most successful models, with validation accuracy values of 97.5% and 96.1%, respectively. In the study of pneumonia detection from CXR images before transfer learning, the InceptionV3 model was the most successful, with an accuracy value of 94.8% during the test phase.

In the post-transfer-learning pneumonia detection study, the InceptionV3 model produced an accuracy value of 98.7% during the testing phase. In detecting COVID-19 from CT images, the ResNet101 architecture produced an accuracy of 58.7% during the test phase before transfer learning, while it produced 77.33% accuracy after transfer learning. In detecting eye diabetes from CT images, AleNet architecture produced 72.8% accuracy during the test phase before transfer learning, while it produced 78.3% accuracy after transfer learning. MediNet database will continue to be developed and expanded for future studies. For this, we need classified data (especially .dicom format), and we need the help of scientists/researchers.

Appendix

Details of AlexNet, VGG19-BN, and Nested-LSTM+CNN architectures are given in Tables 28, 29 and 30.

Table 28 AlexNet Architecture

Stages	Layers	Output size
Input Image	Input Layer	224x224x3
Conv Layers	Conv2d (96) @ 11×11 & 5×5+BN+ReLU	45x45x96
	MaxPooling2d @ 3×3	15x15x96
	Conv2d (256) @ 3×3+BN+ReLU	15×15×256
	MaxPooling2d @ 3×3	5x5x256
	Conv2d (384) @ 3×3+BN+ReLU	5×5×384
	Conv2d (384) @ 3×3+BN+ReLU	5x5x384
	Conv2d (256) @ 3×3+BN+ReLU	5×5x256
	MaxPooling2d @ 3×3	1x1x256
Classification Block	Flatten	256
	FullyConnected (256)+BN+ReLU	256
	Dropout (rate=0.2)	256
	FullyConnected (128)+BN+ReLU	128
	Dropout (rate=0.2)	128
	FullyConnected (2)+Softmax	2

The value after "@" is the kernel_size value. In Max Pooling, it is the pool_size value. The value after the "&" is the strides value. BN: BatchNormalization.

Table 29 VGG19-BN Architecture

Stages	Layers	Output size
Input Image	Input Layer	224x224x3
Conv Block	Conv2d (64) @ 3×3 & 3×3+BN+ReLU	75x75x64
	Conv2d (64) @ 3×3 & 2×2+BN+ReLU	38x38x64
	MaxPooling2d @ 2×2 & 2×2	19x19x64
	2 x Conv2d (128) @ 3×3+BN+ReLU	19x19x128
	MaxPooling2d @ 2×2 & 2×2	9x9x128
	4 x Conv2d (256) @ 3×3+BN+ReLU	9x9x256
	MaxPooling2d @ 2×2 & 2×2	4x4x256
	4 x Conv2d (512) @ 3×3+BN+ReLU	4x4x512
	MaxPooling2d @ 2×2 & 2×2	2x2×512
	4 x Conv2d (512) @ 3×3+BN+ReLU	2×2x512
	MaxPooling2d @ 2×2 & 2×2	1x1x512
Classification Block	Flatten	512
	FullyConnected (256)+BN+ReLU	256
	Dropout (rate=0.2)	256
	FullyConnected (128)+BN+ReLU	128
	Dropout (rate=0.2)	128
	FullyConnected (2)	2

The value after "@" is the kernel_size value. In Max Pooling, it is the pool_size value. The value after the "&" is the strides value. BN: BatchNormalization.

Table 30 Nested-LSTM+CNN Architecture

Stages	Layers	Output size
Input Image	Input Layer	224x224x3
Conv Layers	Conv2d (96) @ 11×11 & 5×5+BN+ReLU	45x45x96
	MaxPooling2d @ 3×3	15x15x96
	Conv2d (192) @ 3×3+BN+ReLU	15×15×192
	Conv2d (256) @ 3×3+BN+ReLU	15x15×256
	Conv2d (128) @ 3×3+BN+ReLU	15x15x128
	MaxPooling2d @ 3×3	5x5x128
	GlobalMaxPooling2D	128
Classification Block	Flatten	15×1440
	NestedLSTM (256)+BN+ReLU	15×256
	Dropout (rate=0.2)	15×256
	NestedLSTM (192)+BN+ReLU	15×192
	Dropout (rate=0.2)	15×192
	NestedLSTM (128)+BN+ReLU	128
	Dropout (rate=0.2)	128
	Concatenate	256
	FullyConnected (2)+Softmax	2

The value after "@" is the kernel_size value. In Max Pooling, it is the pool_size value. The value after the "&" is the strides value. BN: BatchNormalization.

Declarations

Conflict of interest The authors declare no potential conflict of interests.

References

1. Abbas A, Abdelsamea MM, Gaber MM (2020) Detrac: transfer learning of class decomposed medical images in convolutional neural networks. IEEE Access 8:74901–74913. https://doi.org/10.1109/ACCESS.2020.2989273
2. Al-Dhabyani W, Gomaa M, Khaled H, Fahmy A (2020) Dataset of breast ultrasound images. Data in brief 28:104863. https://doi.org/10.1016/j.dib.2019.104863
3. Alok N, Krishan K, Chauhan P (2021) Deep learning-based image classifier for malaria cell detection. In: Machine learning for healthcare applications, pp 187–197. https://doi.org/10.1002/9781119792611.ch12
4. An G, Akiba M, Omodaka K, Nakazawa T, Yokota H (2021) Hierarchical deep learning models using transfer learning for disease detection and classification based on small number of medical images. Sci Rep 11(1):1–9. https://doi.org/10.1038/s41598-021-83503-7
5. Ayana G, Park J, Choe SW (2022) Patchless multi-stage transfer learning for improved mammographic breast mass classification. Cancers 14(5):1280. https://doi.org/10.3390/cancers14051280
6. Borkowski AA, Bui MM, Thomas LB, Wilson CP, DeLand LA, Mastorides SM 2019 Lung and colon cancer histopathological image dataset (lc25000). arXiv preprint arXiv:1912.12142
7. Bukhari SUK, Asmara S, Bokhari SKA, Hussain SS, Armaghan SU, Shah SSH (2020) The histological diagnosis of colonic adenocarcinoma by applying partial self supervised learning. medRxiv. https://doi.org/10.1101/2020.08.15.20175760
8. Chaitanya K et al (2019) Semi-supervised and task-driven data augmentation. In: International conference on information processing in medical imaging, pp 29–41. https://doi.org/10.1007/978-3-030-20351-1_3
9. Chakrabarty N, Brain MRI (n.d.) Images for Brain Tumor Detection. https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection. Accessed 1 March 2021

10. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. https://doi.org/10.48550/arXiv.1406.1078
11. Dabral I, Singh M, Kumar K (2019, December) Cancer detection using convolutional neural network. In: International conference on deep learning, artificial intelligence and robotics. Springer, Cham, pp. 290-298. https://doi.org/10.1007/978-3-030-67187-7_30
12. Darbari A, Kumar K, Darbari S, Patil PL (2021) Requirement of artificial intelligence technology awareness for thoracic surgeons. Cardiothorac Surg 29(1):1–10. https://doi.org/10.1186/s43057-021-00053-4
13. Darma IWAS, Suciati N, Siahaan D (2020) Balinese carving recognition using pre-trained convolutional neural network. In: 2020 4th international conference on informatics and computational sciences (ICICoS). IEEE. 1-5. https://doi.org/10.1109/ICICoS51170.2020.9299021
14. Deepak S, Ameer PM (2019) Brain tumor classification using deep CNN features via transfer learning. Comput Biol Med 111:103345. https://doi.org/10.1016/j.compbiomed.2019.103345
15. Enguehard J, O'Halloran P, Gholipour A (2019) Semi-supervised learning with deep embedded clustering for image classification and segmentation. IEEE Access 7:11093–11104. https://doi.org/10.1109/ACCESS.2019.2891970
16. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, Thrun S (2017) Dermatologist-level classification of skin cancer with deep neural networks. Nature 542(7639):115–118. https://doi.org/10.1038/nature21056
17. Github, (2017) Diabetic Retinopathy Detection. https://github.com/javathunderman/retinopathy-dataset. Accessed 1 March 2021
18. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770-778. https://doi.org/10.1109/CVPR.2016.90
19. He X et al (2020) Sample-efficient deep learning for COVID-19 diagnosis based on CT scans. Medrxiv. https://doi.org/10.1101/2020.04.13.20063941
20. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
21. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700-4708. https://doi.org/10.1109/CVPR.2017.243
22. Hwang JH, Seo JW, Kim JH, Park S, Kim YJ, Kim KG (2022) Comparison between deep learning and conventional machine learning in classifying Iliofemoral deep venous thrombosis upon CT venography. Diagnostics 12(2):274. https://doi.org/10.3390/diagnostics12020274
23. Jain R, Nagrath P, Kataria G, Kaushik VS, Hemanth DJ (2020) Pneumonia detection in chest X-ray images using convolutional neural networks and transfer learning. Measurement 165:108046. https://doi.org/10.1016/j.measurement.2020.108046
24. Ker J, Wang L, Rao J, Lim T (2017) Deep learning applications in medical image analysis. IEEE Access 6:9375–9389. https://doi.org/10.1109/ACCESS.2017.2788044
25. Kermany D, Zhang K, Goldbaum M (2018) Labeled optical coherence tomography (oct) and chest X-ray images for classification. Mendeley Data 2(2). https://doi.org/10.17632/rscbjbr9sj.2
26. Khan HA, Jue W, Mushtaq M, Mushtaq MU (2020) Brain tumor classification in MRI image using convolutional neural network. Math Biosci Eng 17:6203–6216. https://doi.org/10.3934/mbe.2020328
27. Kim YG et al (2020) Effectiveness of transfer learning for enhancing tumor classification with a convolutional neural network on frozen sections. Sci Rep 10(1):1–9
28. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980
29. Kogilavani SV, Prabhu J, Sandhiya R, Kumar MS, Subramaniam U, Karthick A, … Imam SBS (2022) COVID-19 detection based on lung CT scan using deep learning techniques. Comput Math Methods Med 2022:1–13. https://doi.org/10.1155/2022/7672196
30. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp 1097–1105
31. Kumari S, Singh M, Kumar K (2019) Prediction of liver disease using grouping of machine learning classifiers. In: International conference on deep learning, artificial intelligence and robotics. Springer, Cham, pp 339–349. https://doi.org/10.1007/978-3-030-67187-7_35
32. Kwon G, Ryu J, Oh J, Lim J, Kang BK, Ahn C, Bae J, Lee DK (2020) Deep learning algorithms for detecting and visualising intussusception on plain abdominal radiography in children: a retrospective multicenter study. Sci Rep 10(1):17582. https://doi.org/10.1038/s41598-020-74653-1

33. Li T, Gao Y, Wang K, Guo S, Liu H, Kang H (2019) Diagnostic assessment of deep learning algorithms for diabetic retinopathy screening. Inf Sci 501:511–522. https://doi.org/10.1016/j.ins.2019.06.011
34. Li X, Grandvalet Y, Davoine F, Cheng J, Cui Y, Zhang H, Belongie S, Tsai YH, Yang MH (2020) Transfer learning in computer vision tasks: remember where you come from. Image Vis Comput 93:103853. https://doi.org/10.1016/j.imavis.2019.103853
35. Moniz JRA, Krueger D (2017, November) Nested lstms. In: Asian conference on machine learning. PMLR. pp. 530-544. https://doi.org/10.48550/arXiv.1801.10308
36. Narayanan BN, Hardie RC, Krishnaraja V, Karam C, Davuluru VSP (2020) Transfer-to-transfer learning approach for computer aided detection of COVID-19 in chest radiographs. AI 1(4):539–557. https://doi.org/10.3390/ai1040032
37. Narayanan BN, Hardie RC, De Silva MS, Kueterman NK (2020) Hybrid machine learning architecture for automated detection and grading of retinal images for diabetic retinopathy. J Med Imaging 7(3):034501. https://doi.org/10.1117/1.JMI.7.3.034501
38. Negi A, Kumar K (2021) Face mask detection in real-time video stream using deep learning. In: Computational Intelligence and Healthcare Informatics, pp 255–268. https://doi.org/10.1002/9781119818717.ch14
39. Negi A, Kumar K (2022) AI-based implementation of decisive technology for prevention and fight with COVID-19. In: Cyber-physical systems. Academic press, pp 1–14. https://doi.org/10.1016/B978-0-12-824557-6.00008-X
40. Negi A, Chauhan P, Kumar K, Rajput RS (2020, December) Face mask detection classifier and model pruning with keras-surgeon. In: 2020 5th IEEE international conference on recent advances and innovations in engineering (ICRAIE). IEEE. pp. 1-6. https://doi.org/10.1109/ICRAIE51050.2020.9358337
41. Negi A, Kumar K, Chauhan P, Rajput RS (2021, February) Deep neural architecture for face mask detection on simulated masked face dataset against COVID-19 pandemic. In: 2021 international conference on computing, communication, and intelligent systems (ICCCIS). IEEE. pp. 595-600. https://doi.org/10.1109/ICCCIS51004.2021.9397196
42. Noreen N, Palaniappan S, Qayyum A, Ahmad I, Imran M, Shoaib M (2020) A deep learning model based on concatenation approach for the diagnosis of brain tumor. IEEE Access 8:55135–55144. https://doi.org/10.1109/ACCESS.2020.2978629
43. Rajpurkar P, Park A, Irvin J, Chute C, Bereket M, Mastrodicasa D, Langlotz CP, Lungren MP, Ng AY, Patel BN (2020) AppendiXNet: deep learning for diagnosis of appendicitis from a small dataset of CT exams using video pretraining. Sci Rep 10(1):1–7. https://doi.org/10.1038/s41598-020-61055-6
44. Rehman A, Naz S, Razzak MI, Akram F, Imran M (2020) A deep learning-based framework for automatic brain tumors classification using transfer learning. Circuits Syst Signal Process 39(2):757–775. https://doi.org/10.1007/s00034-019-01246-3
45. Reis HC (2017) Calcaneus benign tumor detection using canny edge detector. Int J Oncol Cancer Ther 2:20–23
46. Sandfort V, Yan K, Pickhardt PJ, Summers RM (2019) Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks. Sci Rep 9(1):1–9. https://doi.org/10.1038/s41598-019-52737-x
47. Simonyan K, Zisserman A (2014). Very deep convolutional networks for large-scale image recognition. https://doi.org/10.48550/arXiv.1409.1556
48. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818-2826. https://doi.org/10.1109/CVPR.2016.308
49. Tan M, Le Q (2019, May) Efficientnet: rethinking model scaling for convolutional neural networks. In: International conference on machine learning. PMLR. pp. 6105-6114
50. Wacker J, Ladeira M, Nascimento JEV 2020 Transfer learning for brain tumor segmentation. In: International MICCAI Brainlesion workshop. 241-251. https://doi.org/10.1007/978-3-030-72084-1_22
51. Wang L, Nie D, Li G, Puybareau E, Dolz J, Zhang Q, Wang F, Xia J, Wu Z, Chen JW, Thung KH, Bui TD, Shin J, Zeng G, Zheng G, Fonov VS, Doyle A, Xu Y, Moeskops P, … Shen D (2019) Benchmark on automatic six-month-old infant brain segmentation algorithms: the iSeg-2017 challenge. IEEE Trans Med Imaging 38(9):2219–2230. https://doi.org/10.1109/TMI.2019.2901712
52. Wu YH, Gao SH, Mei J, Xu J, Fan DP, Zhang RG, Cheng MM (2021) Jcs: An explainable covid-19 diagnosis system by joint classification and segmentation. IEEE Trans Image Process 30:3113–3126. https://doi.org/10.1109/TIP.2021.3058783

Hatice Catal Reis received her MSc in Geomatics Engineering from Selcuk University, Turkey, in 2010. She received her PhD degree in Remote Sensing and GIS from Yildiz Technical University, Turkey, in 2015. She is currently working in the Department of Geomatics Engineering (Medical Photogrammetry), Gumushane University, Turkey.

Veysel Turk received his MSc in Computer Engineering from Harran University, Turkey, in 2022.

Kourosh Khoshelham is an Associate Professor at the Department of Infrastructure Engineering of the University of Melbourne. He received his PhD in photogrammetry and laser scanning from the Hong Kong Polytechnic University in 2004, and held Assistant Professor positions at Delft University of Technology and University of Twente before joining the University of Melbourne in 2015.

Serhat Kaya Serhat Kaya is currently a master' student at Dicle University, Department of Mining Engineering.