Chapter 9
Artificial Intelligence for Disease Identification and Diagnosis

A. Lakshmi Muddana, Krishna Keerthi Chennam, and V. Revathi

9.1 Introduction to Medical Data Processing

Data is crucial in finding a solution to any problem. It plays a fundamental role in identifying diseases, finding causes of diseases, and treatment. Increased computational power, low storage costs, and availability of internet have driven the health center to maintain electronic health records. Advancements in medical devices and advances in data analytics led to the application of AI techniques in healthcare in detection and prognosis of diseases. Different types of cancer, heart-related diseases, and epidemics like Covid-19 [1], are leading causes of patient suffering and death. Early diagnosis and detection are crucial to prevent deterioration of patient health and mortality.

AI has driven advances in many fields including finance, agriculture, computer vision, e-commerce, driver less cars, voice-activated personal assistants, and healthcare. Medical data are in the form of medical notes, electronic health records, data from medical devices, lab test results, and images [2]. Data are available in both structured form like images, gene expression, and unstructured form like clinical notes. Application of AI techniques on medical data processing can (1) Uncover clinically relevant information hidden in massive amounts of data that can give health risk alerts and health outcome predictions, (2) Can reduce errors that are inevitable in human clinical practice, (3) More accurate and reliable information to doctors in disease identification and treatment, (4) Reduces manual work and subjectiveness,

A. L. Muddana (✉) · V. Revathi (✉)
GITAM University, Hyderabad, India
e-mail: amuddana@gitam.edu; rvankaya@gitam.edu

K. K. Chennam
Muffakham Jah College of Engineering and Technology, Hyderabad, India
e-mail: krishna.keerthi@mjcollege.ac.in

P. Siarry et al. (eds.), *The Fusion of Internet of Things, Artificial Intelligence, and Cloud Computing in Health Care*, Internet of Things,
https://doi.org/10.1007/978-3-030-75220-0_9

(5) Can find patterns from large scale data to predict the outbreak of pandemics, and (6) AI can combine data from different sources like medical records, radiology images, genome sequence, fitness band data to create personalized treatment plans.

Challenges of medical data processing include (1) Small datasets because data are not stored or due to privacy reasons, (2) Unbalanced datasets in case of cancer and rare diseases, (3) Unavailability of labeled data as labeling is time-consuming and requires doctors with a specialized skill set, (4) Subjectivity in identifying the diseases that hinders decision making, (5) Variability in patients environment and genes.

Medical data can use:

1. Supervised learning methods for building predictive models.
2. Unsupervised learning methods as preprocessing steps for feature extraction, dimensionality reduction, and to identify subgroups before sending to predictive models.
3. Deep learning methods require large amounts of labeled data [3]. But unlabeled data are available in abundance. Semisupervised approach uses a combination of supervised and unsupervised methods when labels or outcomes are missing in the instances of the datasets [4].

AI techniques for medical data processing can be categorized into

1. Machine learning methods like Support vector machines, K-Nearest Neighbors, Ensemble methods that take patients disease history, gene expressions, diagnostic results, clinical symptoms, medication, disease indicators in building the models for disease identification, and diagnosis [5].
2. Deep learning methods that build neural networks to capture nonlinear relationships. It can uncover nonlinear patterns in the data. Popular neural networks are Convolution neural networks for medical Images analysis and LSTM models for sequence data processing.

AI in healthcare is used to support decision making in disease prevention, control, and personalized treatment. It is critical in ensuring that doctors focus on cases that truly matters and leaving the routine ones to the machine. Physicians cannot be replaced by machines but can assist them to make better clinical decisions with more accuracy.

Some of the popular healthcare solutions using AI are IBM Watson Health, Google DeepMind that help in cancer diagnosis, predicting patient outcomes, averting blindness, etc. Ancora Medical that helps in cancer treatment, CloudMedx Health to extract data from electronic health records and outputs clinical insights for healthcare professionals [6].

9.2 Preprocessing Techniques

The dataset needs some sort of initial processing before giving as input to the model, called preprocessing. It is a crucial step in model building so that meaningful insights are drawn from the data. Different preprocessing methods are normalizing data, handling categorical features, handling missing data, handling label noise, elimination of outliers, etc.

Feature scaling: Some of the machine learning methods use gradient descent algorithm as an optimization technique. This algorithm requires the feature ranges to be on a similar scale for fast convergence. Also distance-based methods like K-Nearest Neighbors, K-Means use distance between the data points to measure the similarity. Hence, feature values are to be brought to similar scale. Popular feature scaling techniques are:

Normalization: Feature values are scaled down to the range of 0–1. This is a preferred method if the machine learning algorithm makes no assumption about the data distribution.

Standardization: Feature values are scaled such that the mean is 0 and standard deviation is 1. This method is helpful when the underlying data is normally distributed and also not affected by outliers.

Feature clipping: Sometimes dataset may have outliers. Specify, minimum and maximum values for the features so that values outside minimum and maximum specified are clipped to specified minimum and maximum. Another preprocessing step required in neural networks is to convert categorical data into numerical data since strings will not be converted to float by neural network and may generate error during model fitting.

Batch normalization normalizes inputs to each layer of the neural network.

This results in faster convergence and also provides a bit of regularization.

9.2.1 Handling Missing Data

Some values of the features may not be available in the dataset called missing data or null values. Missing data arise due to data corruption, communication errors, malfunctioning of devices, accidental clicks, and in some cases, data may not be specified deliberately like religion or age of a person [7].

Missing data can be categorized into [8, 9].

1. Missing Completely At Random (MCAR): Where missingness is not related to any characteristics of the dataset like material loss.
2. Missing At Random (MAR): The data are dependent on another feature, which is missing, e.g., Creatine value is dependent on a urine sample, which was missing.
3. Missing Not At Random (MNAR): Data value is missing and is related to the reason for its missing intentionally, e.g., age of female, income of person deliberately not specified.

Missing values will weaken the model by introducing bias. Handling missing data plays an important role in medical datasets since complete datasets produce robust models. Hence, missing values are handled as a preprocessing step before giving the dataset as input to the model.

Methods for handling missing data are [10].

1. **Deletion:** Remove the data based on the proportion of missing values. Deletion can be applied on a row or column, or both. Applying on row deletes the entire observation that has one or more missing values. This may be done when missing data is limited to a small number of observations but may result in producing biased estimates. Dropping the variable or attribute is preferred when more than 60% of values are missing, and the variable is insignificant.
2. **Imputation:**Imputation is estimating the missing value. This can be done by
   (a) Using a measure of central tendency like mean, median for continuous data, and mode for categorical data.
   (b) Use machine learning algorithms like KNN, XGBoost, Random forests to impute the missing values. KNN is widely used which can predict both discrete and continuous values.

9.2.2 *Handling Noisy Labels*

Data labeling is expensive and time-consuming as knowledge experts are required for labeling. In real-world scenarios, labels may also go wrong sometimes and the reasons may be.

1. The available information is insufficient due to poor quality data.
2. Experts often make mistakes during labeling.
3. Incorrect labels may come from communication or encoding problems. Real-world databases are estimated to contain around 5% of encoding errors.
4. Mistakes made during data entry.
5. Annotators may give incorrect labels when part of disease symptoms are given.
6. Variability in interpretation by experts.

   Label noise is different from outliers and feature noise. Label noise in training data decreases performance and increases complexity in learning.

   Methods to handle label noise are:

1. Use label noise-robust models like AdaBoost, naive Bayes, and random forest rather than decision tress and support vector machine.
2. Use data cleansing methods to remove mislabeled samples by outlier detection, anomaly detection, and voting filtering.
3. Use Label noise-tolerant learning algorithms that use prior information to detect like Bayesian prior, beta priors, Hidden Markov, Graphical methods, and probabilistic models.
4. Reduce the label noise in training data. Instead of giving large data with label noise for training, select small data with correct labels, and then apply predictions on it. Then voting of ensemble classifiers can be applied.

Carla E. Brodley [11] used filtering of data for wrong label identification before training the model. Aritra Ghosh [12] introduced noise tolerance risk minimization procedures with different loss functions like sigmoid loss, ramp loss, etc. to deal with label noise. Identification of mislabeled samples is done by Jeremy Speth [13], introducing multilabel identification. Wang et al [14] introduced an iterative learning framework for addressing the label noise issue by three steps including iterative label noise detection, discriminative feature learning and reweighting. Nithika Nigam [15] conducted a survey on different methods to deal with label noise in deep learning algorithms and statistical methods used in nondeep learning methods like bagging and voting mechanism.

Perona et al. [16] discussed Image denoising methods related to partial differential equations (PDEs), Rudin et al. [17] proposed variation-based image restoration with free local constraints, Domain transformations such as wavelets by Coifman [18], DCT method by Yaroslavsky [19], BLS-GSM by Portilla [20]. L Gondara [21] discussed nonlocal techniques including NL-means [22] reviewed different denoising algorithms. Dabov [23] proposed different domain transformations like BM3D. Models exploiting sparse coding techniques are mentioned in [24–26]. Vincent [27] discussed different ways for Extracting and composing robust features with denoising autoencoders.

9.3 Methods to Handle Unbalanced Datasets

The majority of medical datasets are unbalanced. Most of the Machine learning algorithms are designed to perform well when the number of samples in each class are nearly equal. Popular algorithms for balancing numerical datasets are SMOTE, MSMOTE.

SMOTE algorithm is applied on two cancer datasets [28] having features characterizing cell nuclei of the tumor. Dataset1 (Wisconsin) has 30 features computed from digitized image of fine needle aspirate of a breast mass, Dataset2 with 9 features describing the breast tumor characteristics and labels that indicate benign or malignancy of the tumor. Both the datasets have imbalanced classes. Datasets contained 16 null values in bare. Nuclei feature and was replaced by mean value of that feature. As the range of values of the features varies, the features were normalized in both the datasets. Datasets also exhibit class imbalance as shown in Table 9.1. Different methods are available for balancing the imbalanced datasets like Up-sampling and Down-sampling. As the dataset size is small, up-sampling is applied using the SMOTE algorithm. The algorithm synthesizes the samples by taking k-nearest neighbors of the randomly picked minority samples. The resultant datasets after applying SMOTE algorithm are shown in Table 9.2.

Table 9.1 Breast cancer datasets

DataSets	No of features	No. of samples with malignancy	No. of normal samples	Total number of samples
DataSet1	30	212	357	569
DataSet2	9	241	458	699

Table 9.2 Dataset after up-sampling

DataSets	No. of samples with malignancy	No. of normal samples	Total number of samples
DataSet1	357	357	714
DataSet2	458	458	916

9.4 Handling Small Datasets

Typically, real-world medical datasets are small in size and may have a large number of features. This could be due to the nonrecording of patient information, few instances in rare diseases, privacy issues, etc [29, 30]. This challenge needs to be addressed for the model to be accurate and robust. This challenge can be handled using different machine learning techniques like regularization, data augmentation, transfer learning, etc.

9.4.1 Regularization Techniques

One of the factors for poor model performance overfitting. It occurs when the model performs well on training data but performs poorly on the unseen test data.

Methods to combat the overfitting problem are

[font = red!50!black] Reduce the features of the dataset. But it may not be the right choice as it may result in the loss of useful information. Collect more data to increase the dataset size to train the model. But it may not always be possible to collect more data. Perform **Data Augmentation** to create new examples from existing examples of the dataset so as to in increasing dataset size. During the training process, as the number of iterations in training increase, train loss and validation loss decrease. But after some point, validation performance decreases. Stop training the model at this point called **Early stopping**.

Regularization penalizes the parameters taking large values and avoids overfitting. Different regularization techniques are

[font = red!50!black] **L1 regularization** is also called Lasso regression, where the sum of absolute values of a coefficient is added as a penalty term to the loss function. It shrinks the coefficient towards zero and discourages learning complex models to avoid overfitting. **L2** regularization is also called ridge regression, where sum of squares of the coefficient is added to the loss function. **Dropout**

regularization is used in deep learning. In each iteration of training, some nodes of the neural network are randomly made inactive. This results in having a different set of nodes in each iteration giving different outputs. It penalizes the weight matrices of nodes. Smaller weight matrices lead to simpler models and reduce overfitting.

9.4.2 Data Augmentation

In deep learning models, the neural network contains many layers to model complex relations in the data. Deep networks have more neurons in hidden layers creating a large number of trainable parameters, which require large datasets. Medical datasets are small in size due to the unavailability of recorded information. To handle the small size datasets, apply transformations on the available data to synthesize new data points, called data augmentation.

For the model to generalize well, the dataset size should be big enough and have variations in the data. Train the model with synthetically modified data to get better performance. Data augmentation can address the issues of diversity of data, amount of data, and also solve class imbalance issues. It is applied as a preprocessing step before applying the learning algorithm called off-line data augmentation, which is preferred for small datasets. Online data augmentation performs translation on a mini-batch, which is preferred for large datasets.

Data augmentation can be applied to different data forms like numerical, image, and text. Popular numerical data augmentation techniques are SMOTE and MSMOTE, which are already discussed in Sect. 9.3.

Image data augmentation: In real-world scenarios, images might have been taken under different conditions like different locations, orientations, scale, and brightness. Image data augmentation can be done by applying transformations on images like geometric transformations, color space transformations. Geometric transformations include rotation, flipping, scaling, and cropping. Flipping should be done carefully on medical data sets. For example, in chest X-ray if we perform a flip, then heart position will change from left to right which can yield to the case of dextrocardia. Rotation of image may result in a change of dimensions and need to be resized. Color space transformations like color casting, varying brightness, noise injection, etc. used when challenges are connected to the lighting of images [31, 32]. Image data augmentation is useful in computer vision tasks like object detection, image classification, image segmentation, etc. (Figs. 9.1, 9.2, 9.3, and 9.4).

These transformations may lead to changes in the image geometry and the image may lose its original features. This can be overcome with modern techniques like Generative Adversarial networks (GAN), neural style transfer that perform more realistic transformations.

GAN is a deep learning-based generative modeling approach to generate new images from available images. The model consists of two submodels. Generators submodel learns patterns in input and generate new images. A random vector is

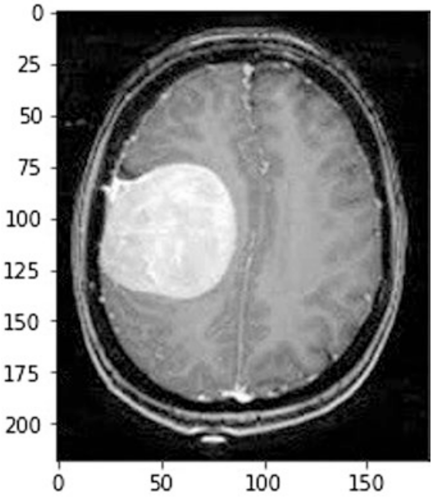

Fig. 9.1 BrainTumor sample image

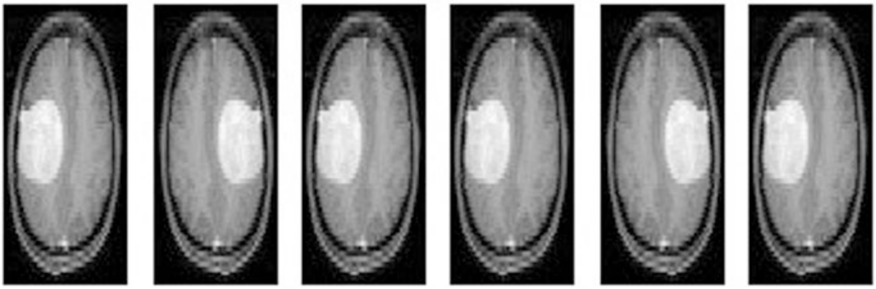

Fig. 9.2 After horizontal shift

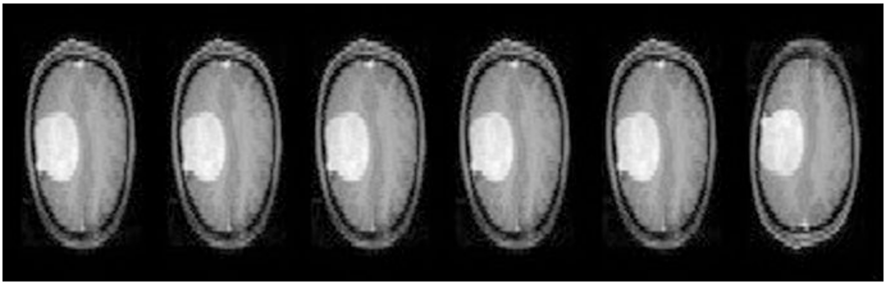

Fig. 9.3 After vertical shift

drawn from Gaussian distribution and is used as a seed in the generative process. The generated images look very similar to real images from the domain. The discriminative submodel classifies whether a given image is a real or generated one [33–35]. Popular use cases of GAN are filling images from the outline, converting

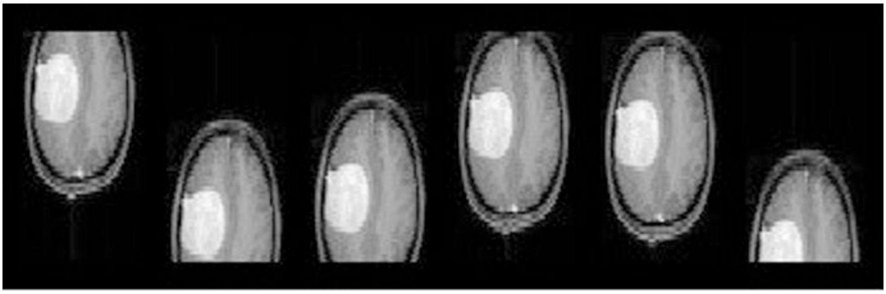

Fig. 9.4 After height shift

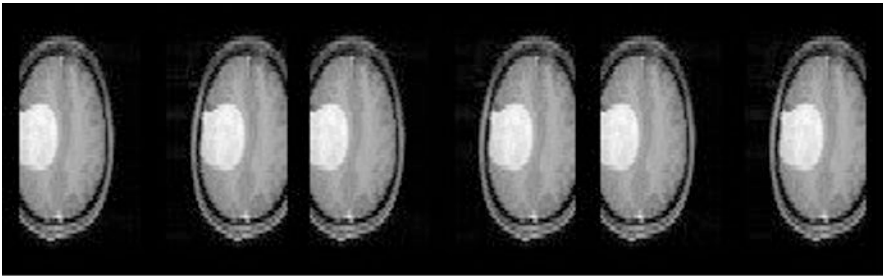

Fig. 9.5 After width shift

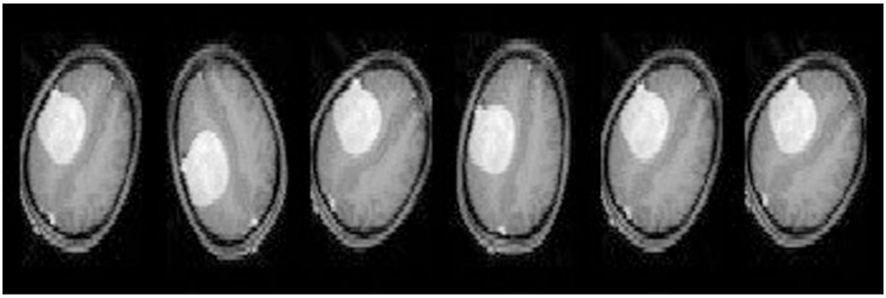

Fig. 9.6 After rotation

black and white images to color, and photo realistic depictions of product prototypes. In medical images, discriminator is used as regularizer or discriminator for abnormal images.

Neural Style Transfer (NST): New image is generated by taking the content of one image (content image) and style of another image (style image). The generated image looks like a new image. The image looks more artistic than realistic (Figs. 9.5, 9.6, 9.7, and 9.8).

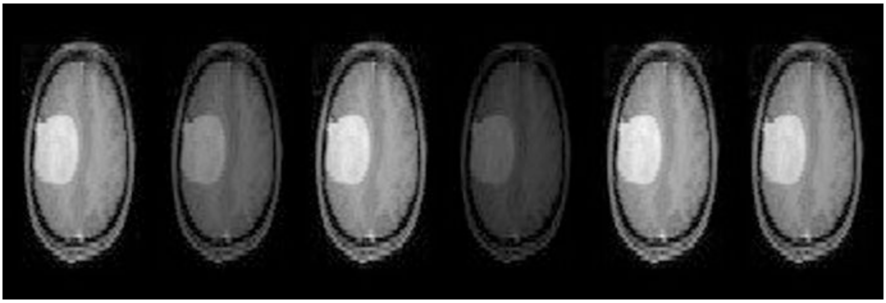

Fig. 9.7 After brightness effect

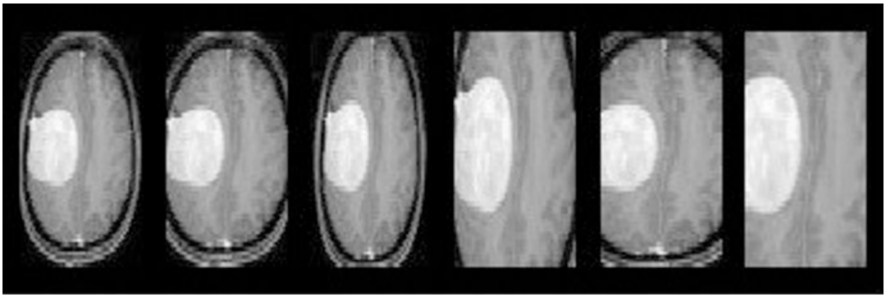

Fig. 9.8 After applying zoom effect

9.4.3 Transfer Learning

CNN can learn complex mappings when trained on enough data. Medical datasets are typically small in size. Training deep neural networks on small datasets results in overfitting. When we train deep neural networks on an image dataset, the first few layers of convnet recognize horizontal and vertical lines and colors. The next few layers learn simple shapes and colors using the features learned in previous layers. Subsequent layers try to learn parts of an object. The last layers recognize whole objects and perform classification. In any convolutional neural networks, other than the last few layers, the layers learn basic features. Using the pretrained model and replacing only the last few layers result in saving training time and computational power required. Deep learning networks take long training time on large datasets. Models like VGG, ResNet, InceptionNet are trained on benchmark datasets with millions of examples and thousands of classes. These top-performing models are made available and platforms like Keras provide libraries to reuse them [36]. Goal of Transfer Learning is to learn from related classification tasks for relevant data by identifying various types of abnormalities [37]. Transfer learning in Medical Imaging can be done by using two types (1) Same domain different task: The easiest method is to use learning from various tasks in similar domains. (2) Different

Table 9.3 Model performance using transfer learning

DataSet	Model	Train accuracy	Validation accuracy	Train loss	Validation loss
Covid-19 Chest X-ray with transfer learning	VGG16 (classifier)	1	1	0.0042	0.0013
Covid-19 Chest X-ray with transfer learning	VGG16 (freez 10 layers)	0.9889	1	0.022	0.0018
Brain tumor with transfer learning	VGG16 (classifier)	0.9802	0.9841	0.1003	0.0924
Brain tumor with transfer learning	VGG16 (freez 10 layers)	0.8063	0.8181	0.4213	0.4047

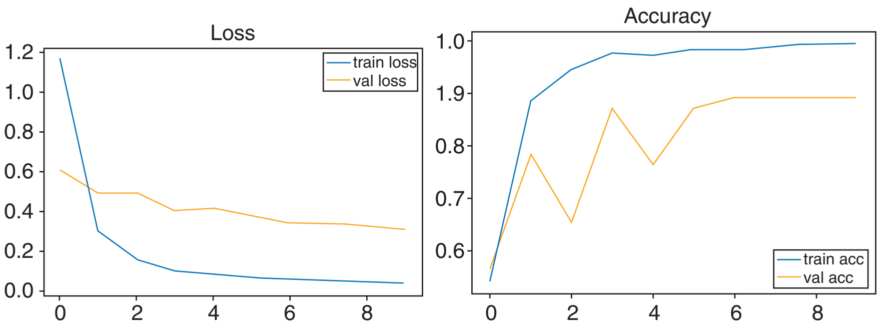

Fig. 9.9 Loss and accuracy curves for Covid−19 chest x-ray

domains same task: Initial start point is identified and then network tuned for the final task [38, 39]. Using pretrained models brings the benefit of decreased training time and results in lower generalization error.

Transfer learning can be used as:

(a) Classifier where pretrained model is downloaded and new image is input to predict the class.
(b) Can be used as a feature extractor. Layers prior to the output layer can be used as input to the layers of the new model. Take layers of pretrained models, freeze them and add new layers on top of these to train on the new small dataset. Pretrained weights are used as initial weights to the new model and continue learning on the new dataset.

Table 9.3 shows the model performance after applying transfer learning (a) as a classifier, (b) freezing 10 layers of VGG16 model, on two datasets Covid-19 chest X-ray images and Brain tumor images. Figures 9.9 and 9.10 show Loss and Accuracy curves, on Covid-19 Chest X-ray dataset, before and after transfer learning.

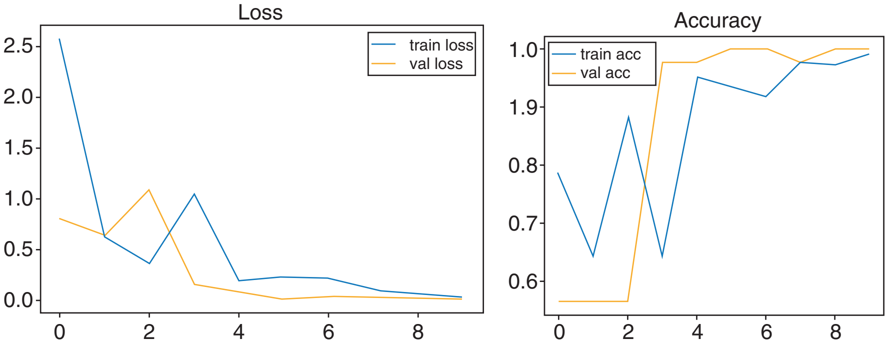

Fig. 9.10 Loss and accuracy curves for Covid−19 chest X-ray after transfer learning

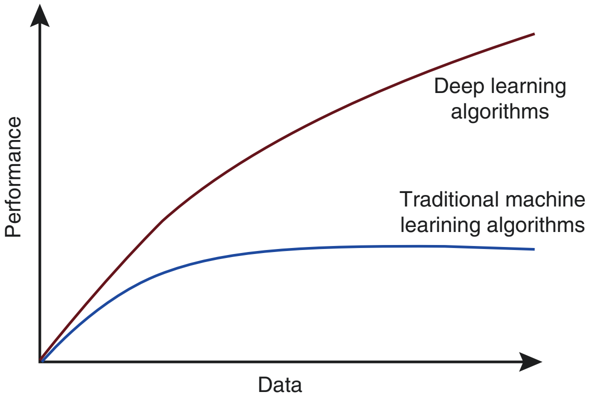

Fig. 9.11 Performance of machine learning and deep learning algorithms

9.5 Deep Learning Techniques

Traditional machine learning algorithms like Logistic Regression, Decision Trees, K-NN, and SVM can make use of the volume of data to some extent. Their performance will not improve further even if more data are available, as depicted in Fig. 9.11.

Deep learning techniques can make use of voluminous data by building complex model to learn nonlinear relationship among data. With lot of activities being digitized, like electronic health records, data are recorded and made available. This large amount of data can be utilized using deep learning methods to give more accuracy compared to machine learning methods. Deep learning algorithms are inspired by the structure and functioning of the human brain called artificial neural networks. Different Deep learning applications in healthcare are, detecting and diagnosing cancer cells, disease prediction and treatment, drug discovery, precision medicine, identifying health insurance fraud, etc [40].

Different neural network architectures are

[font = red!50!black]Feed forward neural networks (FNN), Convolutional neural networks (CNN), for image input like MRI images, X-rays, CT-scans. Recurrent neural networks (RNN), for sequence data like text, audio, time series data.

9.5.1 Autoencoders

Autoencoder is a type of neural network used for unsupervised learning where the dataset contains few labels or no labels. It encodes input data to some hidden representation and then decodes backward to original form. Autoencoder consists of three parts:

[font = red!50!black]**Encoder** that maps input data to hidden or compressed representation. **Bottleneck** layer represents compressed representation of input. Decoder that maps hidden representation back to original data as losslessly as possible by minimizing Reconstruction loss function.

Autoencoder performs nonlinear transformation to learn abstract features using neural networks. Classification or regression can then be applied on latent features.

Autoencoder architectures may include:

[font = red!50!black]Simple Feed Forward Networks Convolutional autoencoders that contain convolutional encoding and decoding layers to process image input. It is better suited for image processing for Image reconstruction, Image colorization, Latent space clustering, and Generating high resolution images. LSTM networks for sequence data.

Use cases of autoencoders are data compression, image denoising, dimensionality reduction and feature selection, and extraction ignoring noise. So it works well for correlated input features.

Autoencoders are built, with the following architecture, on Covid-19 chest X-ray dataset having 181 train images and 56 test images as shown in Table 9.4.

Table 9.4 Covid-19 chest X-ray dataset having 181 train images and 56 test images

Layer (type)	Outshape	Param
input5(*InputLayer*)	[(None, 240, 240, 3)]	0
conv2d18(*Conv*2*D*)	(None, 240, 240, 32)	896
max*pooling*2*d*8(*MaxPooling*2)	(None, 120, 120, 32)	0
(Conv2D)19	(None, 120, 120, 32)	9248
up*sampling*2*d*8(*UpSampling*2)	(None, 120, 120, 32)	0
conv2d21(*Conv*2*D*)	(None, 120, 120, 32)	9248
up*sampling*2*d*9(*UpSampling*2)	(None, 240, 240, 32)	0
conv2d22(*Conv*2*D*)	(None, 240, 240, 3)	867

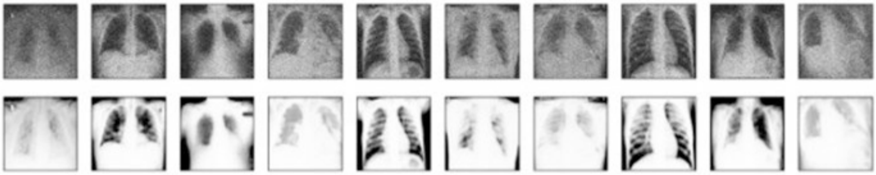

Fig. 9.12 Eliminated denoised output image of autoencoder

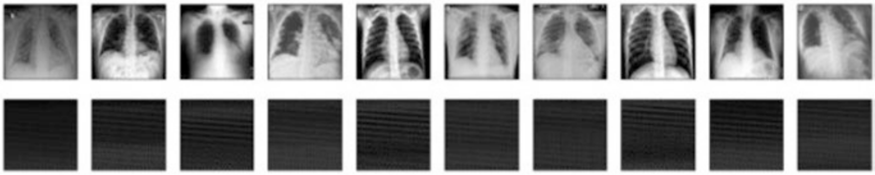

Fig. 9.13 Latent feature learning using autoencoder

Table 9.5 Model performance on breast cancer datasets using Machine learning algorithms

DataSets	Logistic regression	Logistic regression	SVM	SVM
	Train accuracy	Test accuracy	Train accuracy	Test accuracy
DataSet1	97.51	98.39	97.89	98.39
DataSet2	96.96	97.29	98.15	97.34

Total params: 29,507
Trainable params: 29,507
Nontrainable params: 0

Following is the output when autoencoder is used for denoising in Fig. 9.12 and latent feature learning using autoencoder as shown in Fig. 9.13

9.5.2 Neural Networks for Medical Datasets

Logistic regression and SVM models are rebuilt on the two up-sampled Breast cancer datasets described in Sect. 9.3, which shows the performance of the machine learning algorithms as shown in Table 9.5.

To improve the performance, a semisupervised learning technique can be adopted. Feature learning was applied using autoencoders to determine latent features. Autoencoder was tuned with different optimizers and mini batch sizes. Low loss was obtained with RMSPROP optimizer with batch size of 16 and trained for 100 epochs.

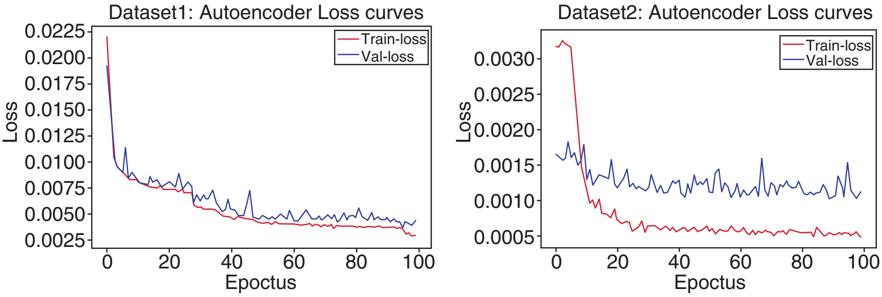

Fig. 9.14 Loss curves of autoencoder on breast cancer datasets

Table 9.6 Model performance on breast cancer using neural networks

DataSets	Feed forward neural network	Feed forward neural network
DataSets	Train accuracy	Test accuracy
DataSet1	99.81	98.88
DataSet2	98.98	99.12

Loss curves as depicted in Fig. 9.14, show good performance of the autoencoder. On the learned features of up-sampled data, the Feed forward Neural network classifier model was built. Neural network was tuned with different optimizers, batch sizes and number of hidden layers. Good train accuracy, test accuracy, and low variance were achieved with ADAM optimizer, nine hidden layers, and mini batch size of 16. The accuracy measures on train and test data are shown in Table 9.6.

Semisupervised learning using autoencoders for latent feature learning on Up-sampled data and neural network model for binary classifier has shown good performance. This gives good train, test accuracy, and could reduce the variance to less than 1% in both the datasets.

9.5.3 Convolutional Neural Networks (CNN)

Deep learning architectures are popular for image tasks. CNN is a type of deep neural network used for image input to perform feature extraction, classification, finding patterns, and in other computer vision tasks [37]. Applications include object detection, object classification, driverless vehicles, etc. Convolutional networks eliminate manual feature extraction and automatically detect important features of an image. CNN has a sequence of layers where each layer of the network detects different features of the image. The output of each layer is input to the next layer. It performs a series of convolution and pooling operations followed by fully connected layers. Convolution layer performs convolution operation that merges two sets of information like image and convolution filter, which produces a feature map. The input image is put through filters that activate certain features of images. The convolution operation is followed by pooling to reduce the dimension and number of parameters. This reduces training time and avoids overfitting. Commonly used pooling methods are maximum or average pooling.

Each neuron in the network takes inputs from previous layer neurons, applies activation function to produce output, which then becomes an input to next layer neurons. The activation function introduces nonlinearity into the output of neurons. Nonlinear activations perform transformations on an input to learn complex relationships. Popular Activation functions are Relu, Sigmoid, Tanh. Visualizing intermediate layers output on Covid-19 Chest X-Ray dataset is shown in Fig. 9.15.

Two Convolutional networks were built to classify Covid-19 Chest X-Ray dataset. Model 1 is a simple network with one convolution and max pooling layer. Model 2 is a deep network with three blocks of convolution and pooling layers. The model was fine-tuned on different optimizers. SGD / RMSProp found to be performing well with less variance is shown in Table 9.7. Loss and accuracy curves are shown in Figs. 9.16 and 9.17.

9.6 Open Research Problems

Methods to integrate complete data of patients like clinical notes, test values, disease indicators of patients, gene expression data, and medical images are required, to develop a comprehensive model. Such models can accurately predict diseases and help in personalized treatment.

Not many datasets are available on various diseases. There is a need for developing datasets on different diseases and make them available for research. Researchers need to build Generative models to perform more realistic transformations on medical images to increase the dataset size than simple data augmentation methods. This helps in developing complex models on small datasets and to avoid overfitting.

Many medical datasets are small in size. Use of pretrained models helps when the dataset size is small and new models can be built on these for faster training on new problems. Open source high-performing models like VGG, InceptionNet, ResNet on medical image datasets are needed which can be used for transfer learning.

Label noise in medical datasets significantly impacts the predictions and in supporting decision making. Focus is required in developing the methods to identify and handle label noise in medical datasets.

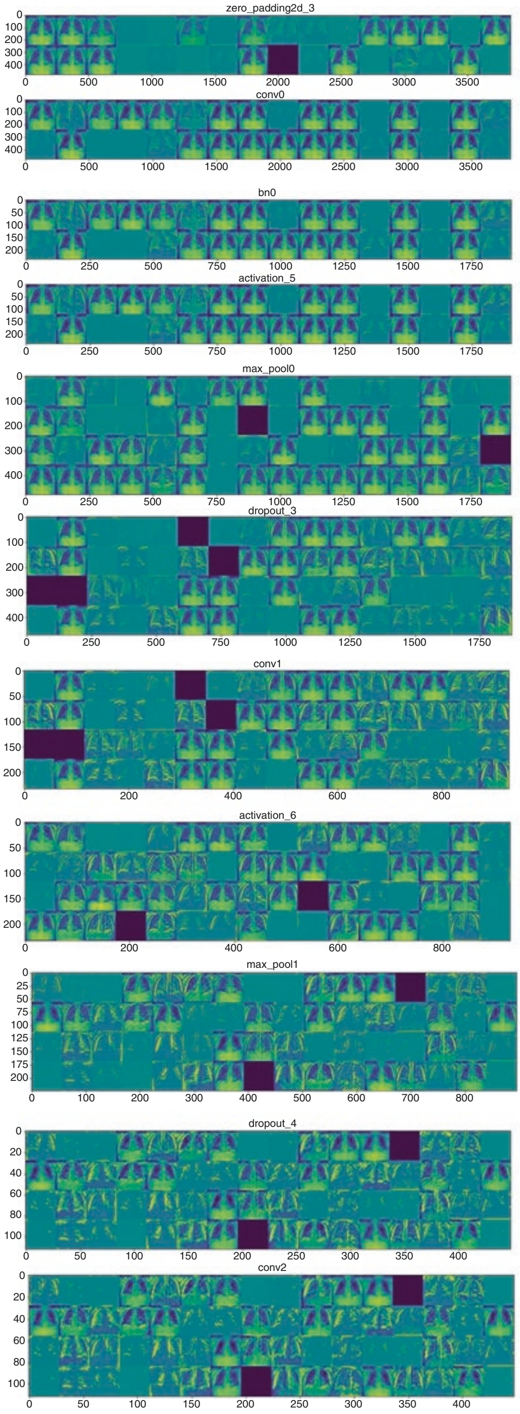

Fig. 9.15 Visualization of intermediate layers of covid-19 chest X-ray

Table 9.7 Model performance on covid-19 chest X-ray using CNN

DataSet	Model	Optimizer	Train accuracy	Validation accuracy	Train loss	Validation loss
Covid-19 Chest X-ray	Model 1 (one layer)	adam	0.9944	0.8913	0.0357	0.3068
	Model 2 (multiple blocks of conv and pooling layers)	adam	0.9723	0.913	0.1121	0.245
		sgd	0.9889	0.9782	0.0629	0.1791
		sgd	0.9889	0.9782	0.0629	0.1791
		adagrad	0.9944	0.8478	0.032	0.30144

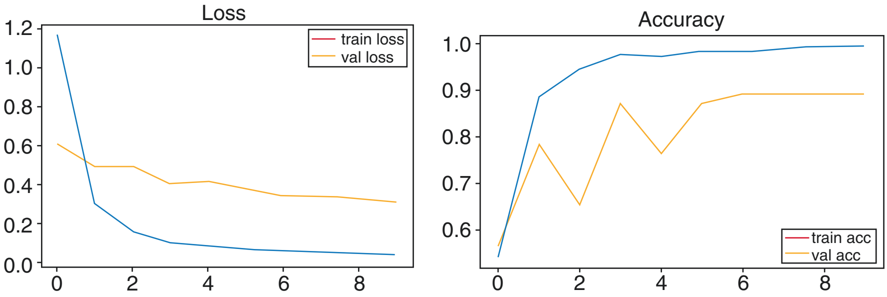

Fig. 9.16 Loss and accuracy curves of model 1

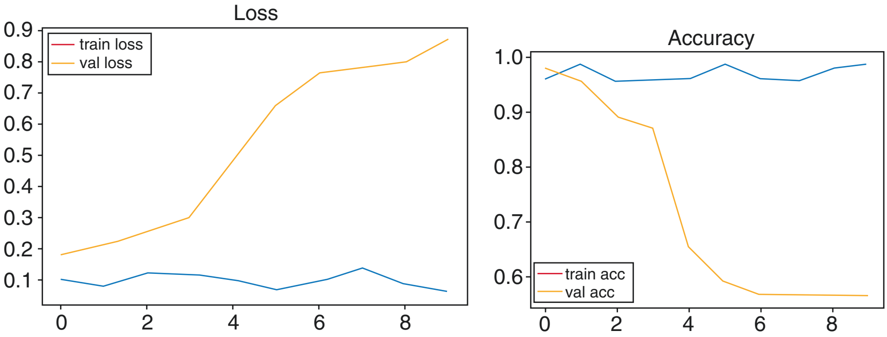

Fig. 9.17 Model 2 loss and accuracy curves with "SGD" optimizer

9.7 Future Scope

Deep learning models results in more accuracy when trained on large datasets. As real-world medical datasets are typically small, datasets can be augmented with more realistic images using generative models. GAN architecture and its performance on medical datasets can be discussed in the future work. Reinforcement learning techniques can be explored on medical datasets, in progressive decision making of disease diagnosis.

9.8 Conclusion

The chapter discusses the challenges in medical data processing for detecting and diagnosing diseases. The challenges include small datasets, missing data, and unbalanced datasets. Various methods to deal with the challenges like imputing missing data, increasing the dataset size using data augmentation techniques, Transfer learning using predefined models like VGG, ResNet, InceptioNet, and regularization methods are discussed. These methods are applied on medical datasets and the results are presented. Neural network models on two cancer datasets are built and the results are presented. Convolution network architectures for classifying medical image datasets to predict diseases like Covid-19 and Brain Tumor are presented. Autoencoders are built for image denoising, dimensionality reduction, and feature extractions to improve the model performance are presented on cancer datasets. The chapter concludes with open research problems and future scope to be explored in utilizing AI to provide robust healthcare solutions.

References

1. Amini, A., Chen, W., Fortino, G., Li, Y., Pan, Y. & Wang, M. D. (2020). Editorial: Special issue on AI-driven informatics, sensing, imaging and big data analytics for fighting the covid-19 pandemic. *IEEE Journal of Biomedical and Health Informatics, 24*(10).
2. Iyengar, S., Bonda, F. T., Gravina, R., Guerrieri, A., Fortino, G., & Sangiovanni Vincentelli, A., 2008, March. A framework for creating healthcare monitoring applications using wireless body sensor networks. In Proceedings of the ICST 3rd international conference on body area networks (pp. 1–2).
3. Piccialli, F., Di Somma, V., Giampaolo, F., Cuomo, S., & Fortino, G. (2021). A survey on deep learning in medicine: Why, how and when? *Information Fusion, 66*, 111–137.
4. Wang, Z., Wu, D., Gravina, R., Fortino, G., Jiang, Y., & Tang, K. (2017). Kernel fusion-based extreme learning machine for cross-location activity recognition. *Information Fusion, 37*, 1–9.
5. Chatterjee, R., Maitra, T., Islam, S. H., Hassan, M. M., Alamri, A., & Fortino, G. (2019). A novel machine learning-based feature selection for motor imagery EEG signal classification in internet of medical things environment. *Future Generation Computer Systems, 98*, 419–434.

6. Aluvalu, R., & Muddana, L. (2016). A dynamic attribute-based risk aware access control model (DA-RAAC) for cloud computing. 2016 IEEE international conference on computational intelligence and computing research (ICCIC). IEEE.
7. Köse, T., Özgür, S., Coşgun, E., Keskinoğlu, A., & Keskinoğlu, P. (2013). The prevention and handling of the missing data. *Korean Journal of Anesthesiology, 64*(5), 402–406.
8. Leke, C. A., & Marwala, T. (2019). Introduction to missing data estimation. In *Deep learning and missing data in engineering systems of studies in big data* (Vol. 48, pp. 1–20). Springer.
9. Sterne, J. A. C., White, I. R., Carlin, J. B., et al. (2009). Multiple imputation for missing data in epidemiological and clinical research: Potential and pitfalls. *BMJ, 338*(1), b2393.
10. Wells, B. J., Nowacki, A. S., Chagin, K., & Kattan, M. W. (2017). Strategies for handling missing data in electronic health record derived data. *eGEMs (Generating Evidence & Methods to Improve Patient Outcomes), 1*(3).
11. Brodley, C. E., & Fried, M. A. (1999). Identifying mislabeled training data. *Journal of Artificial Intelligence Research, 11*, 131–167.
12. Ghosh, A., Manwani, N., & Sastry, P. S. (2015). Making risk minimization tolerant to label noise. *Elsevier Neurocomputing, 160*, 93–107.
13. Speth, J., & Hand, E. M. (2019). Automated label noise identification for facial attribute recognition. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops (pp. 25–28).
14. Wang, Y., & Liu, W., et al. (2018). Iterative learning with open-set noisy labels. 2018 IEEE/CVF conference on computer vision and pattern recognition. https://doi.org/10.1109/CVPR.2018.00906.
15. Nigam, N., Dutta, T., & Hari, P. G. (2019). Impact of noisy labels in learning techniques: A survey. In Conference: Advances in data and information sciences.
16. Perona, P., & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*(7), 629–639.
17. Rudin, L. I., & Osher, S.. Total variation-based image restoration with free local constraints. Image Processing, 1994. Proceedings. ICIP-94. IEEE international conference (Vol. 1). IEEE.
18. Coifman, R. R., & Donoho, D. L. (1995). *Translation-invariant denoising*. Springer.
19. Yaroslavsky, L. P., Egiazarian, K. O., & Astola, J. T.. (2001). Transform domain image restoration methods: Review, comparison, and interpretation. Photonics west 2001-electronic imaging. *International Society for Optics and Photonics*.
20. Portilla, J., et al. (2003). Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Transactions on Image Processing, 12*(11), 1338–1351.
21. Gondara, L. (2016). Medical image denoising using convolutional denoising autoencoders. In 2016 IEEE 16th international conference on data mining workshops (ICDMW), Barcelona, Spain (pp. 241–246).
22. Buades, A., Coll, B., & Morel, J.-M. (2005). A review of image denoising algorithms, with a new one. *Multiscale Modeling and Simulation, 4*(2), 490–530.
23. Dabov, K., et al. (2007). Image denoising by sparse 3-D transform domain collaborative filtering. *IEEE Transactions on Image Processing, 16*(8), 2080–2095.
24. Olshausen, B. A., & Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research, 37*(23), 3311–3325.
25. Elad, M., & Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing, 15*(12), 3736–3745.
26. Mairal, J., et al. (2009). Online dictionary learning for sparse coding. Proceedings of the 26th annual international conference on machine learning. ACM.
27. Vincent, P., et al. (2008). Extracting and composing robust features with denoising autoencoders. Proceedings of the 25th international conference on machine learning. ACM.
28. Akkalakshmi, M., Riyazuddin, Y. M., Revathi, V., & Pal, A. (2020). Autoencoder-based feature learning and up-sampling to enhance cancer prediction. *International Journal of Future Generation Communication and Networking, 13*(1), 1453–1459.
29. Payan, A., & Montana, G. (2015). Predicting Alzheimer's disease: A neuroimaging study with 3D convolutional neural networks. arXiv preprint arXiv:1502.02506.

30. Mansoor, A., Cerrolaza, J., Idrees, R., Biggs, E., Alsharid, M., Avery, R., & Linguraru, M. G. (2016). Deep learning-guided partitioned shape model for anterior visual pathway segmentation. *IEEE Transactions on Medical Imaging, 35*(8), 1856–1865. https://doi.org/10.1109/TMI.2016.2535222
31. Wong, K. K., Fortino, G., & Abbott, D. (2020). Deep learning-based cardiovascular image diagnosis: A promising challenge. *Future Generation Computer Systems, 110*, 802–811.
32. Khan, S. R., Sikandar, M., Almogren, A., Din, I. U., Guerrieri, A., & Fortino, G. (2020). IoMT-based computational approach for detecting brain tumor. *Future Generation Computer Systems, 109*, 360–367.
33. Jabbar, M. A., Aluvalu, R., & Reddy, S. (2017). Cluster-based ensemble classification for intrusion detection system. In Proceedings of the 9th international conference on machine learning and computing (pp. 253–257). ACM.
34. Xu, J., Xiang, L., Liu, Q., Gilmore, H., Wu, J., Tang, J., & Madabhushi, A. (2016). Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images. *IEEE Transactions on Medical Imaging, 35*, 119–130.
35. Cheng, J.-Z., Ni, D., Chou, Y.-H., et al. (2016). Computer-aided diagnosis with deep learning architecture: Applications to breast lesions in US images and pulmonary nodules in CT scans. *Scientific Reports, 6*, 24454. https://doi.org/10.1038/srep24454
36. Ismail, W. N., Hassan, M. M., Alsalamah, H. A., & Fortino, G. (2020). CNN-based health model for regular health factors analysis in internet-of-medical things environment. *IEEE Access, 8*, 52541–52549.
37. Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering, 22*(10), 1345–1359.
38. Zhang, R., Zheng, Y., Mak, T. W. C., Yu, R., Wong, S. H., Lau, J. Y., & Poon, C. C. (2017). Automatic detection and classification of colorectal polyps by transferring low-level CNN features from nonmedical domain. *IEEE Journal of Biomedical and Health Informatics, 21*(1), 41–47.
39. Menegola, A., Fornaciali, M., Pires, R., Bittencourt, F. V., Avila, S., & Valle, E. (2017). Knowledge transfer for melanoma screening with deep learning. In International symposium on biomedical imaging (ISBI) (pp. 297–300). IEEE.
40. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., Van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis, 42*, 60–88.