SN Computer Science (2022) 3:292
https://doi.org/10.1007/s42979-022-01166-1

SURVEY ARTICLE

Survey of Supervised Learning for Medical Image Processing

Abeer Aljuaid[1] · **Mohd Anwar**[1]

Received: 14 January 2022 / Accepted: 20 April 2022 / Published online: 17 May 2022

Abstract
Medical image interpretation is an essential task for the correct diagnosis of many diseases. Pathologists, radiologists, physicians, and researchers rely heavily on medical images to perform diagnoses and develop new treatments. However, manual medical image analysis is tedious and time consuming, making it necessary to identify accurate automated methods. Deep learning—especially supervised deep learning—shows impressive performance in the classification, detection, and segmentation of medical images and has proven comparable in ability to humans. This survey aims to help researchers and practitioners of medical image analysis understand the key concepts and algorithms of supervised learning techniques. Specifically, this survey explains the performance metrics of supervised learning methods; summarizes the available medical datasets; studies the state-of-the-art supervised learning architectures for medical imaging processing, including convolutional neural networks (CNNs) and their corresponding algorithms, region-based CNNs and their variants, fully convolutional networks (FCN) and U-Net architecture; and discusses the trends and challenges in the application of supervised learning methods to medical image analysis. Supervised learning requires large labeled datasets to learn and achieve good performance, and data augmentation, transfer learning, and dropout techniques have widely been employed in medical image processing to overcome the lack of such datasets.

Keywords Deep learning · Convolutional neural network (CNN) · Fast R-CNN · Faster R-CNN · FCN · Mask R-CNN · Medical image processing · Supervised learning · U-Net

Introduction

In many cases, accurate diagnoses of diseases rely heavily on image acquisition systems and image interpretation. Image acquisition and reconstruction devices (e.g., computed tomography (CT) and magnetic resonance imaging (MRI) scanners) have been improved in recent years, and they now support the collection of higher resolution medical images, such as radiological images (e.g., X-ray, CT, and MRI scans) and microscopic images (e.g., histological photos). Medical image interpretation requires diligence and expertise to extract useful information from large amounts of data [1, 2] For example, to diagnose cancer, pathologists use microscopes to look for changes in the cytology and architecture of the cell structure, and one sample may contain a million cells [3]. If the cells are small, even an experienced pathologist may misclassify cancer [4], and delayed or inaccurate diagnosis increases the mortality rate [1, 5]. Therefore, there is a great need for accurate automated medical image analysis, which requires efficient, effective machine learning algorithms.

Automated medical image analysis can reduce the burden on pathologists and radiologists; furthermore, it provides precise diagnoses and accelerates the diagnostic process. Machine learning and deep learning methods are widely used for automated medical image processing. Machine learning models learn from data, identify patterns, and make appropriate predictions or decisions based on those data. Machine learning has significantly impacted medical research and healthcare delivery. However, the performance of machine learning algorithms for image processing relies

This article is part of the topical collection "Innovative AI in Medical Applications" guest edited by Lydia Bouzar-Benlabiod, Stuart H. Rubin and Edwige Pissaloux.

✉ Mohd Anwar
manwar@ncat.edu

Abeer Aljuaid
aaljuaid@aggies.ncat.edu

[1] Department of Computer Science, North Carolina A&T State University, 1601 E Market St, Greensboro, NC 27411, USA

heavily on feature extraction algorithms and requires an expert to select the most useful features for the task.

Machine learning algorithms process images in two stages. In the first stage, a hand-crafted feature extraction method extracts important features from the image. In the second stage, a classifier method is applied to classify the image further based on feature extraction. Thus, using machine learning algorithms in medical image analysis is tedious and time consuming [6, 7].

Deep learning algorithms have been proven to surpass machine learning algorithms in medical image analysis tasks [7–12]. Deep learning algorithms are capable of extracting image features automatically, which makes them more suitable for automated medical image analysis and able to provide accurate diagnoses [8, 11, 13, 14]. For image processing, deep learning algorithms can be used to train models for automatic identification of objects by analyzing millions of images.

Deep learning can be classified as supervised and unsupervised learning. The supervised learning has yielded exceptional results in medical image processing, with performance comparable to that of humans [5, 8]. The supervised learning requires a ground truth dataset and prior knowledge about the output of the dataset. The goal of supervised learning is to understand the relationship and structure of the input dataset to predict the output accurately.

Unlike supervised learning, unsupervised learning allows direct learning of a data pattern without the need for labels [15]. The unsupervised learning understands and determines the inherent structure of a set of data points using statistical methods such as clustering algorithms and density estimation [15]. Unsupervised learning algorithms can be used not only for classification, detection, and segmentation but also for other tasks such as compression, dimensionality reduction, denoising, super-resolution, and reconstruction of images.

To the best of our knowledge, no comprehensive survey on supervised learning techniques for medical image processing has been published to date. This article provides a survey of the supervised learning techniques used in medical image processing tasks, including the available medical image datasets, evaluation matrices, state-of the-art architectures, and applications of medical image processing.

We conducted an online search of peer-reviewed articles from IEEE, ACM, Science Direct, SpringerLink, Wiley, PubMed, and Scopus. We conducted the search using the following keywords: deep learning, medical image processing tasks, the state-of-the-art of supervised learning techniques, medical image dataset, performance metrics, and transfer learning. We included articles that were related to artificial intelligence in medical image analysis and supervised learning algorithms. We excluded articles that were: short (less than 3 pages), secondary or tertiary studies (such as literature reviews, surveys, and others), unavailable in full-text, or structured as a tutorial or editorial.

The rest of this paper is organized as follows. Section 2 provides an overview of the medical image processing tasks of classification, detection, and segmentation. Section 3 explores supervised learning architectures and quantitative evaluation metrics. Section 4 summarizes the available medical image datasets and the supervised deep learning application in medical images processing. Section 5 discusses the trends, accuracy, influencing factors, and challenges in applying supervised learning to medical image analysis. Finally, Sect. 6 concludes this paper.

Medical Image Processing Tasks

Medical image processing tasks can be grouped into classification, detection and segmentation tasks (Fig. 1), which are usually performed manually by clinicians [5]. Deep learning can be leveraged to automate time-consuming medical image processing and to perform predictive modelling for detection of disease such as cancer cells.

In classification, known as a computer-aided diagnosis (CAD) [5], objects are categorized into groups or types based on specific features. Classification can be binary (e.g., benign or malignant) or multi-class (e.g., classifying a lesion into multiple types or degrees) [17]. For example, in the presence of cancer cells, classification can be performed to differentiate between normal and abnormal cells or to categorize the cancer cells into multiple grades (e.g., mild, moderate, or severe).

Detection involves finding the region of objects in an image by drawing bounding boxes. For example, tissue

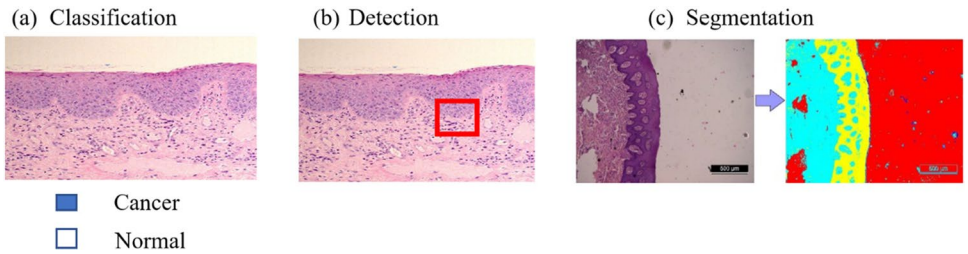

Fig. 1 Use of deep learning techniques for essential medical tasks: **a** classification, **b** detection, and **c** segmentation [16]

heterogeneities (anomalous lesions) are detected by drawing bounding boxes [5, 17].

Segmentation is important for precise delineation of organs or structures on medical images for accurate diagnosis, treatment, or surgical planning [5, 17]. Segmentation predicts pixel-wise masks of the objects in an image, labeling them by drawing precise contours. Segmentation can be semantic or instance-based. Semantic segmentation is a type of pixel-level classification that generates only one mask for the whole image. It treats objects belonging to one class as a single instance, but it cannot differentiate individual instances. Instance segmentation combines object detection and semantic segmentation [18]—object detection to identify a region of interest (ROI) and semantic segmentation to predict a segmentation mask for each ROI—so it can separate individual instances. For example, in the presence of cancer cells, the purpose of segmentation is to delineate the cell shapes.

Supervised Deep Learning Architectures and Performance Metrics

This section discusses the performance metrics and state-of-the-art supervised deep learning architectures.

Performance Metrics

When applying supervised learning algorithms for image processing tasks, performance metrics are needed to evaluate the designed models [19]. Each task has specific metrics.

For classification tasks, the performance metrics count the numbers of correct and incorrect predictions based on the true positive (TP, a model that correctly predicts the positive class), true negative (TN, a model that correctly predicts the negative class), false positive (FP, a model that incorrectly predicts the positive class), and false negative (FN, which incorrectly predicts the negative class) approach to calculate accuracy, precision, recall, specificity and F-score [19].

The accuracy refers to the proportion of all correct predictions to the total number of predictions [19]. Top-N accuracy represents the rate at which the model correctly predicts the positive class in the top-N highest probabilities.

The precision is the proportion of true positive to the total number of positive predictions (either correct or incorrect predictions) [19]. The precision compute the model's accuracy in classifying samples as positive. The precision concerns to correctly classify all positive class and avoid misclassifying negative samples as positive. For examples, the precision represents how many patients really have cancer from all the patients that are predicted as positive for cancer, and low precision means high false positives (classified many negative samples as positives).

The recall (known as sensitivity and true positive rate) is the ratio of true positive to the total number of positive samples [19]. The recall computes the model's capability to classify positive samples. In contrast to precision, the recall considers how accurately classifying all positive samples, but it does not consider if a negative sample is incorrectly labeled. For examples, the recall shows how many cancer patients were predicted as positive from all the cancer patients, and high recall means the model predicts most or all the positive samples.

In contrast to recall, specificity is the ratio of negative samples that were classified as negatives [19]. The specificity measures the model's ability to classify negative samples. For examples, the specificity shows how many non-cancer patients were correctly predicted from all the non-cancer patients, and high specificity means the model predicts most or all the negative samples.

The F-score is a harmonic mean of recall and precision. A high value of the F-score indicates the model has high recall and precision and low FN and FP rate [3]

Classification performance can be described visually using a table (confusion matrix) or graph [receiver-operating characteristic (ROC) and area under the curve (AUC)]. A confusion matrix provides more details about the model performance by counting TP, TN, FP, and FN for each class. The ROC is a probability curve that shows the trade-off between TP rate (recall) and FP rate (1-specificity) at various thresholds [19, 20].

The ROC draws the performance of a model without any consideration of distribution or class error costs [21]. A good model has a ROC curve that reaches the top left corner of the ROC curves, while a curve at the lower right corner or below the diagonal represents a poor classifier [22]. The area under the ROC graph is represented by the AUC, and it is used to compare different ROC curves. A good model has a large value of the AUC [21]. Figure 2 illustrates the ROC curve that compared different supervised deep learning algorithms' prediction. The inception-ResNetv2 [23] algorithm has the highest ROC curve of 76.60%, whereas the lowest ROC curve of 71.63% obtained from Inception-v3 [24]. This means the inception-ResNet-v2 algorithm outperforms other algorithms [22].

For object detection and segmentation tasks, the performance metrics are a measure of the difference between the proposed and predicted segmentation masks, i.e., between the ground truth and prediction bounding boxes. The intersection over union (IoU), average precision (AP), mean AP (mAP), Jaccard index, Dice coefficient, and Hausdorff distance are used for detection and segmentation evaluation.

The IoU is used to compute the number of TP, FP, and FN for object detection by calculating the overlap ratio between

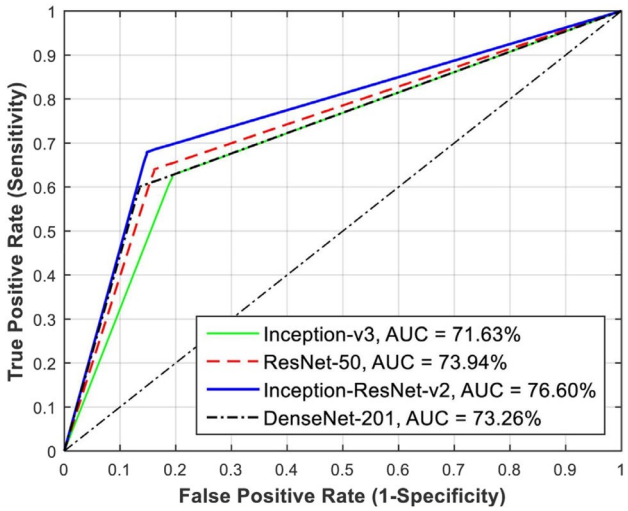

Fig. 2 ROC curves of different deep learning algorithms' predictions [25]

the ground truth and prediction bounding boxes based on a threshold [26, 27]. For instance, if the threshold is 0.5, any bounding boxes with an IoU > 0.5 are TP. Otherwise, they are counted as FP. However, the TN does not use in object detection due to the endless number of bounding boxes that should not be predicted [27].

Average Precision (AP) is the most common metrics used for evaluating object detection [27]. AP is calculated the area under the precision and recall curves (PRC) at various levels. PRC is similar to ROC, but it describes the trade-off between precision and recall [21, 27]. However, the PRC is a zigzag curve, making estimating the AUC more challenging [27]. To remove the zigzag curve, 11-point interpolation and all-point interpolation methods are used to calculate the AP [27].

11-point interpolation has summarized the curve of precision and recall by calculating the average of the maximum precision values at a set of 11 recall levels. Where the all-point interpolation would be considered all recall levels instead of just including 11 levels [27]. The equation of both interpolations is in Table 1.

AP is computed individually for every class. The mean AP (mAP) is an object detection metric that is used to calculate the mean of the AP over all classes [27]. mAP with high value means the best detection model.

The Jaccard index, Dice coefficient, and Hausdorff distance are the most applied evaluation metrics for medical image segmentation, and they are calculated from the IoU at the pixel level [28–30]. Jaccard Index and dice coefficient are used to measure the similarity between the ground truth and predicated segmentation mask with a value range between 0 and 1 [30].

Jaccard index is a proportion of the number of similarity pixels to the total number of both similarity and dissimilarity pixels [28]. The high value of Jaccard and dice coefficient indices represents a good segmentation model [30].

Table 1 Performance metrics for various image processing tasks, including classification, detection, and segmentation

Classification	Detection	Segmentation
$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)}$ [1] [19]	$\text{IoU} = \frac{\text{area}(B_p \cap B_{gt})}{\text{area}(B_p \cup B_{gt})}$ [2] [27]	$\text{Jaccardindex} = \frac{\lvert G \cap S \rvert}{\lvert G \cup S \rvert}$ [6] [9]
$\text{Precision} = \frac{TP}{(TP+FP)}$ [19]	AP_{11} [3] $= \frac{1}{11}\sum_{r\epsilon\{0,0.1,\ldots,1\}} \text{pinterp}(r)$ [27]	$\text{DiceCoefficient} = \frac{2*\lvert G \cap S \rvert}{\lvert G \rvert + \lvert S \rvert}$ [9]
$\text{Recall} = \frac{TP}{(TP+FN)}$ [19]	Where $\text{pinterp}(r) = \max_{\tilde{r} \geq r} p(\tilde{r})$ [27]	$\text{AverageHausdorffDistance} = \frac{GtoS+StoG}{G+S}/2$ [7] [31]
$\text{Specificity} = \frac{TN}{(TN+FP)}$ [19]	AP_{11} [4] $= \sum_n (r_{n+1} - r_n)\text{pinterp}(r_{n+1})$ [27]	
$\text{F} - \text{score} = \frac{2\times(\text{Precision}\times\text{Recall})}{(\text{Precision}+\text{Recall})}$ [19]	Where $\text{pinterp}(r_{n+1}) = \max_{\tilde{r}_{n+1} \geq r} p(\tilde{r})$ [27]	
	mAP [5] $= \frac{1}{N}\sum_{i=1}^{N} AP_i$ [27]	

[1] TP = True Positive, TN = True Negative, FP = False Positive, and FN = False Negative

[2] IoU = Intersection over Union, B_p = Prediction bounding box, and B_p = Ground truth bounding box

[3] AP_{11} = 11-point interpolation, $Pinterp(r)$ = precision replaced with the maximum precision whose recall value is $\geq r$ where $P(\tilde{r})$ is the precision at each recall level r

[4] All point interpolation where n is the number of all recall levels, and Pinterp(r_{n+1}) is the precision replaced with the maximum precision whose recall value is $\geq r_{n+1}$

[5] Mean Average Precision where N is the number of classes and (AP_i) is AP of i class

[6] $|G \cap S|$ is the area of intersection between a set of pixels for ground truth (G) and a set of pixel for prediction segmentation (S), and $|G \cup S|$ is the area of union between (G) and (S)

[7] GtoS is the average Hausdorff distance from ground truth to segmentation where StoG is the average Hausdorff distance from segmentation to ground truth

The Hausdorff distance is the most common metric applied for medical image segmentation [31]. It is compared the ground truth images with the predicated segmentation result, and it ranks different segmentation results from best to worst [31]. Table 1 shows the mathematical formulations of the performance metrics for classification, detection, and segmentation.

Supervised Deep Learning Architectures

Convolutional Neural Network

Convolutional neural networks (CNNs) are the most common architectures used in supervised deep learning techniques. CNNs are the state-of the-art architectures for classification, detection, and segmentation [2, 8, 11, 13, 32, 33] and surpass human performance in image classification [5, 8]. The reasons for the success of CNNs in various image processing tasks are its requirement for fewer parameters than needed in a dense network, its ability to extract features automatically from a large amount of data, and its characteristics of local connectivity and parameter sharing.

Local connectivity means that each hidden unit is connected to a patch (subregion) of an input image called a receptive field. Parameter sharing refers to a patch sharing a set of weights (filter or kernel). In contrast, a dense network requires a weight for each unit, and each weight in a layer is connected to each neuron in a sequential layer; in other words, it is fully connected. This configuration leads to a high number of parameters and high computational cost due to the calculations of the linear activations of the hidden layers. Thus, CNNs reduce both the memory storage and parameter requirements compared to dense networks, leading to increased network efficiency.

A CNN consists of a convolutional layer, rectified linear activation function, normalization unit, pooling layer, dropout unit, and fully connected layer (Fig. 3).

Convolutional Layer Unit The convolutional layer is the central component of a CNN and can automatically extract important features from an input image by applying a convolution operation. The convolution operation is a linear operation that computes the dot product of a set of weights (filter or kernel) and receptive fields to produce an output (feature map). The convolutional layer can have more than one filter and can produce more than one feature map.

In contrast to a linear neural network, the set of weights in a CNN is a multidimensional array (2D for grayscale images and 3D for color images) known as a filter or kernel, and each filter represents a specific feature. The filter is smaller than the input data for repetition on each overlapping filter-sized patch (receptive field).

The filter starts at the top left of the input data and shifts horizontally to the right by the stride length. When the filter reaches the top right of the input data, it moves down vertically by the same stride length and starts over from the left side of the input data. The process is repeated until the entire image is covered and the feature map is computed. Figure 4 illustrates a convolution operation.

Interaction of the filter with the input image creates a feature map that is smaller than the input image. For example, a filter of size 3×3 can reduce an input image of 64 pixels to a feature map with 36 pixels (Fig. 4). Padding is a means of increasing the size of the feature map; it adds extra pixels of value zero around the perimeter of the input image and, consequently, each pixel in the image gets a chance to be at the center of the filter.

A CNN can have more than one convolutional layer stacked together to produce hierarchical features. The output of the first convolutional layer is concatenated with low-level learned features extracted from a second convolutional layer. The combination of low-level features produces multifeatures that can express the shape. The process continues until the very deep layers are more class specific, such as faces or animals. Thus, the first convolutional layers extract generic functions such as lines, dots, corners, and so on, and

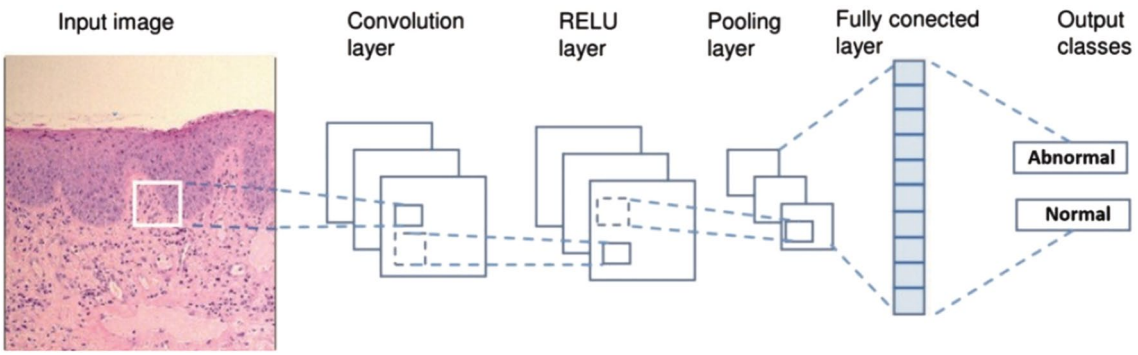

Fig. 3 CNN architecture

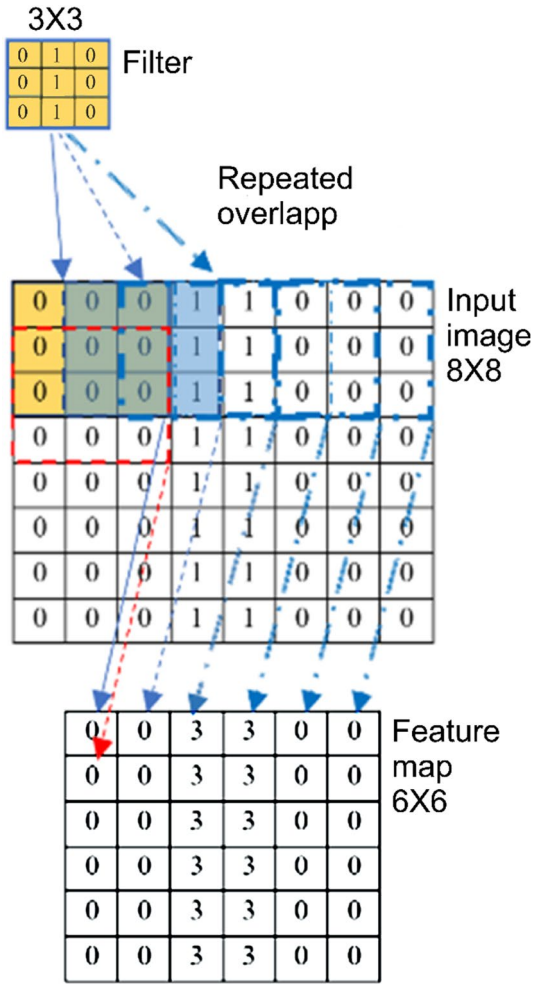

Fig. 4 Convolution layer extracts features from an input image by calculating the dot product of a 3×3 filter and an 8×8 input image to produce a 6×6 feature map. The filter represents a vertical line feature

the deeper layers extract high-level features such as shapes, faces, and the entire object [5, 8, 32, 34–38].

Rectified Linear Activation Function Unit Layer The rectified linear unit (ReLU) layer is the second CNN layer. It was introduced by Krizhevsky et al. [39] in 2012 and is an activation function that sets all negative values to zero. Mathematically, it is defined as

$$f(x) = \max(0, x). \tag{1}$$

The ReLU layer is used to avoid the vanishing gradient problem [39], in which a model is unable to propagate useful information from the final layer to the initial layer. An S-shaped activation function (e.g., sigmoid and tan$h$) transfers an input value into a range, e.g., (0, 1) for sigmoid or (−1, 1) for tan$h$. When the weight is updated in a deeper network with an S-shaped function, the derivation of the S-shaped function becomes quite small; thus, the network is unable to update and converge weights to the first layers.

Normalization Unit Normalization scales down the activation features in the limited range (e.g., 0 to 1). It is used to restrict the unbound activation functions (e.g., ReLU) from increasing the output layers value, and to accelerate the learning process of CNN. There are many normalization techniques such as local response normalization [39], batch normalization [40], weight normalization [41], Layer normalization [42], group normalization, and weight standardization [43]. The first two are the most adapted in deep learning [38, 39, 44] and in medical images processing [2, 6, 11, 13, 34, 35].

Local response normalization was inspired by the neurobiology concept of lateral inhabitant that means an excited neuron inhibits the activity of its neighbors. Local response normalization is applied for local contrast enhancement using the local maximum pixel value as excitation activation for the subsequent layers. Local response normalization can be applied after the activation function. Mathematically, it is defined as:

$$b^i_{x,y} = \frac{a^i_{x,y}}{\left(K + \alpha \sum_{j=\max\left(0,\frac{i-n}{2}\right)}^{\min\left(N-1,\frac{i+n}{2}\right)} \left(a^j_{x,y}\right)^2\right)^{\beta}}, \tag{2}$$

where i represents the output of filter (feature map). $a_{x,y}$ is the pixel value of the feature map before normalization at (x,y) position. N represents the total number of feature maps. n is the adjacent length. k, α, and β are hyperparameters.

Batch normalization reduces internal covariate shift by standardizing the internal layer's input for each mini-batch. Internal covariate shift refers to the change of the distribution of the input features with the weight updated in the prior layer during the training time. Internal covariate shift slows the convergency of deep learning by requiring small rate learning (a hyperparameter that determines the amount of weights that are updated during training time), cautious initialization, and difficult to train deep learning with saturating nonlinearities function [40].

Batch normalization scales down each minibatch based on the standard normal distribution, then two trainable parameters are applied for scaling and shifting the normalized value. The batch normalization algorithm is represented in Fig. 5. Even though the distributions of the input feature are changed during training time, they will be changed in the same mean and variance with batch normalization.

Batch normalization can be applied after each convolution layer and before activation function.

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Fig. 5 Batch Normalization's algorithm [40]

Pooling Layer After the ReLU layer, there comes pooling layer, which reduces the spatial size of the input representation [5, 45]. Feature maps record the precise locations of the input image features, so any shift in the location of a feature leads to a different feature representation. The pooling layer reduces the spatial size and allows the feature representation to be more invariant to translation, enabling the recognition of objects more than their precise locations [5, 45].

The pooling layer applies a pooling operation, which is like a convolution operation [45]. It computes the dot product of a pooling filter and a fixed-shape window of feature maps known as a pooling window. The pooling filter shifts from the top left of the feature map to the right and from top to bottom by the stride length, covering the entire feature map. Unlike a convolution filter, a pooling filter does not have parameters. Pooling operations can employ average, max, and global pooling.

Average pooling computes the average value of the elements in the pooling window, whereas max pooling calculates the maximum value of the elements in the pooling window. Global pooling summarizes an entire feature map in a single value (the strongest activation value) [5, 45].

Dropout Unit Dropout [46] is a regularization technique used to solve the overfitting problem by reducing the model's complexity. An overfitting problem occurs when training a complex supervised deep learning model (such as a CNN) with an unsuitable dataset [11, 34]. Overfitting affects model generalization and performance by learning noise from the training dataset. Dropout randomly removes some activation nodes at each training iteration based on the dropout ratio. These dropped out nodes are blocked in the forward pass and backpropagation.

Fully Connected Layer The last CNN layer is the fully connected layer, which is used to predict a label for an image. The fully connected layer is a linear layer [5, 47]. A linear layer consists of an input layer, one or more hidden layers, and an output layer. All linear layers are fully connected; specifically, each neuron in a layer is connected with each neuron in the subsequent layer. Each linear layer is calculated using the following formula:

$$g(wx + \text{bias}), \tag{3}$$

where g is an activation function (e.g., ReLU), w is a weight vector, and x is an input vector.

The output layer of the fully connected layer applies an activation function (e.g., softmax) to compute a probability score (a number ranging from 0 to 1) for each class label.

Pre-trained Convolutional Neural Network CNNs have widely been used in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [48], especially after the impressive results achieved by LeNet-5 [49] in handwriting recognition. The ILSVRC is an annual competition for computer vision tasks that uses a subset of ImageNet [50]—a large public dataset with more than 14 million images and 21,000 classes. The ILSVRC facilitates the development of different versions of CNNs known as pretrained CNNs. CNNs and their variations became state-of-the art technologies for image processing from 2012 to 2015 [48]. Consequently, these approaches have been adopted for many medical image processing tasks.

The *LeNet-5* architecture, which first demonstrated the feasibility of CNNs, was introduced by LeCun et al. [49] in 1998. LeNet-5 utilized the CNN architecture for handwriting recognition, achieving a 99.2% classification accuracy. The input of LeNet-5 consists of grayscale images. LeNet-5 has seven layers: the first four layers are for the feature extractor block, and the next three are for the classifier block.

The feature extractor block consists of a convolutional layer, a tan$h$ activation function, and an average pooling layer that are repeated twice. The convolutional layer has a 5×5 filter with a stride of 1. The number of filters starts from 6 and increases to 16 with the network depth. The average pooling is non-overlapping (the stride of the pooling operation is equal to the pooling filter, e.g., 2×2). The last three layers of LeNet-5 are fully connected layers for interpretation and final prediction.

AlexNet was designed by Krizhevsky et al. [39] in 2012 to solve the task of classifying 1.5 million images into 1000 classes at ILSVRC-2010. AlexNet follows the design of LeNet-5 with some differences: AlexNet increases the number of layers to eight, the first five being for feature extraction and the last three for classification.

AlexNet introduces some novel features that have become essential components in CNN architecture. It was the first architecture to apply a ReLU function after each convolutional layer instead of S-shaped functions such as tan$h$. AlexNet replaces the non-overlapping average pooling in

LeNet-5 with overlapping max pooling (the stride of the pooling operation is smaller than the pooling filter, e.g., 3×2). Overlapping max pooling reduces the top-one error rate (the rate at which the model gives the highest score to the correct class) by 0.4% and the top-five error rate (the rate at which the model correctly predicts the positive class to be among the classes with the five highest probabilities) by 0.3% compared to those obtained by non-overlapping pooling.

In addition, AlexNet introduces a pattern of stacking convolutional layers, where the output of one convolutional layer is used as input for the following convolutional layer with no pooling layer between them, to provide more distinctive feature maps by applying a ReLU layer after each convolutional layer. AlexNet increases the number of filters with the network depth, from 96 to 256, 384, 384, and 256. The filter size starts from 11×11 and decreases to 5×5 and 3×3 in the depth layer.

Furthermore, AlexNet applies local response normalization after ReLU in the first two convolutional layer, and as a result, the top-1 and top-5 were reduced by 1.4% and 1.2%, respectively. AlexNet uses dropout regularization and data augmentation to reduce the overfitting problem and improve the performance accuracy. Data augmentation techniques create a transformed version of the original images to expand the training dataset without the need to collect new data. AlexNet ranked first at ILSVRC-2012, achieving a top-five test error rate of 15.3%, whereas the second ranking approach yielded 26.2%.

ZFNet was proposed by Zeiler and Fergus [38] in 2014 for ImageNet classification. ZFNet is similar to AlexNet, the only difference is the size of the filter and stride. ZFNet decreases the filter size and stride of the two first convolutional layers from 11×11 with stride 4 to 7×7 with stride 2. Consequently, ZFNet can obtain more distinctive feature maps without aliasing, leading to classification accuracy improvement by 1.6% in terms of the top-five test error rate. ZFNet proved to be the state-of-the-art approach at ILSVRC 2013 for ImageNet classification, achieving a top-five test error rate of 14.8%.

The Visual Geometry Group (VGG) was designed by Simonyan and Zisserman [51] in 2014 for classification and localization tasks. The VGG stretches the depth of the CNN to 16 (VGG-16) and 19 (VGG-19) convolutional layers, approximately twice the number of layers in AlexNet.

The VGG applies a small receptive field (3×3 and 1×1 with a stride of 1 pixel). The filter number starts from 64 and then increases by a factor of 2. The VGG utilizes stacked convolutional layers. Unlike AlexNet, the VGG uses non-overlapping max pooling with a size of 2×2 and stride of 2. The VGG approach achieved a top-five test rate error of 7.32%. VGG is illustrated in Fig. 6.

Conv3-256 Conv3-256	Conv3-256 Conv3-256 Conv3-256
maxpool	
Conv3-512 Conv3-512 Conv3-512	Conv3-512 Conv3-512 Conv3-512 Conv3-512
maxpool	
Conv3-512 Conv3-512 Conv3-512	Conv3-512 Conv3-512 Conv3-512 Conv3-512
maxpool	
3-FC	
Soft-max	

Fig. 6 Architectures of VGG-16 and VGG-19 [51]

GoogleNet [37] was designed for classification and detection tasks. GoogleNet increases the depth and width of the CNN while keeping the computational budget stable (12× fewer parameters than AlexNet). The key innovation of GoogleNet is the inception module, which replaces the fully connected convolutional layer with a sparsely connected layer. GoogleNet has 22 convolutional layers, or 100 layers including the pooling layers, inception modules, and auxiliary classifiers.

A naïve inception module is a block of three parallel convolutional layers and a max pooling layer. The convolutional layers have various sizes corresponding to the receptive fields (1×1, 3×3, and 5×5), and the max pooling layer is 3×3. The outputs of each convolutional layer and the max pooling layer are concatenated to a single vector used as input for the next stage. However, the drawback of a naïve inception module is that convolution operation calculation becomes quite expensive with a larger filter size (e.g., 3 or 5), especially with stacked inception modules. Therefore, the 1×1 convolutional layer is applied before the 3×3 and 5×5 convolutional layers for dimension reduction. Figure 7 illustrates the inception module, where (a) illustrates the naïve inception module and (b) depicts the inception module with dimension reduction.

GoogleNet connects an auxiliary classifier to the intermediate layers. The auxiliary classifier is used to increase the discriminative power of the lower layers, solve the vanishing gradient problem, and provide extra regularization. The auxiliary classifier has five layers:

1. Non-overlapping average pooling layer (5×3).
2. 1×1 convolutional layer for dimensional reduction with 128 channels and ReLU.
3. Fully connected layer with 1024 units and ReLU.
4. Dropout layer with a ratio of 0.7.

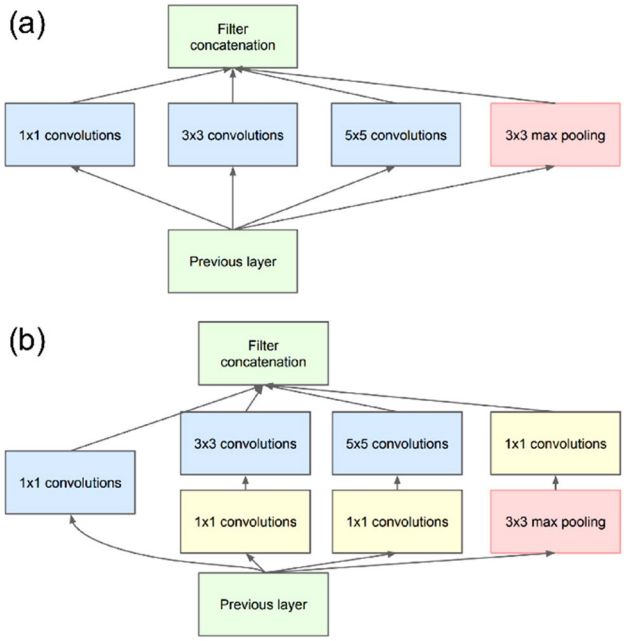

Fig. 7 GoogleNet inception module [37]. **a** Naïve inception model. **b** Inception module with dimension reduction

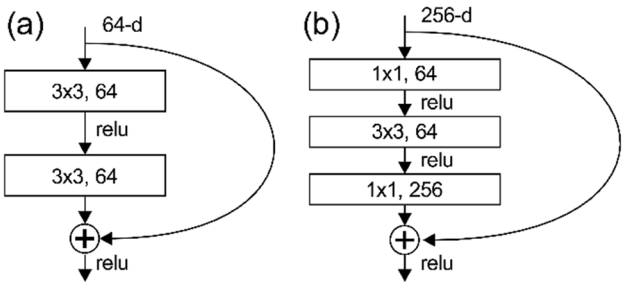

Fig. 8 Residual block architecture [44]. **a** Residual block. **b** Residual block with dimension reduction

5. Softmax layer to predict 1000 classes (the same number of classes that the main classifier predicts).

GoogleNet applies local response normalization after the first and third convolutional layer. Unlike the previous network architecture, GoogleNet replaces the fully connected layers with average pooling layers. Therefore, its top-one test error rate is improved by 0.6%. It surpassed the VGG at ILSVRC-14 in the classification tasks, with a top-five test error rate of 6.67%, whereas the VGG yielded 7.32%.

The deep residual network (ResNet) was proposed by He et al. [44] in 2016 for classification, detection, localization, and segmentation. ResNet introduces a residual block to solve the vanishing gradient problem that comes from increasing the network depth. The residual mapping does not require extra parameters; therefore, ResNet is eight times deeper than VGG and has lower complexity (fewer parameters).

The residual block (Fig. 8) is designed based on skip connection, where the input into the residual block is fused with its output. The residual block consists of two 3×3 convolutional layers, and each one is followed by a ReLU layer. Using a 3×3 convolution layer can increase the computational complexity by stacking more residual blocks. Consequently, ResNet adds 1×1 convolutional layers before and after the 3×3 convolutional layer for dimension reduction and restoration.

ResNet uses batch normalization after each convolutional layer and before ReLU.

ResNet secured first place at the ILSRVC 2015 competition for classification, detection, localization, and segmentation. It achieved a top-five test error rate of 3.57%, and it improved object detection on the COCO dataset [52] by 28%.

CNN with Transfer Learning Training a complex supervised deep learning model (such as a CNN) requires numerous labeled samples for good performance and generalization [2, 11, 32–34], and creating a high quality dataset with a massive number of samples is expensive and complex [37] especially if it requires human intervention for labeling as in the cases of the medical dataset.

Transfer learning method helps train a complex model with a small dataset. Transfer learning transfers knowledge from source tasks to target tasks. For example, knowledge obtained from a pretrained model developed to recognize an animal organ can be used to classify a human organ. Transfer learning initializes a new model by reusing the weights of pretrained models that were developed to solve related tasks. Figure 9 shows the differences between traditional deep learning and transfer learning.

If the two tasks are similar, then the deeper layer of the pretrained network can be used as the starting point for the new model. If the two tasks are different, then the weights of the early layers of the pretrained network can be fine-tuned by refreezing the deeper layers and re-training them for a new task. Initializing a model with the weights from the pretrained model would improve the performance and decrease the training time [7, 34, 35, 53].

Region-Based Convolutional Neural Network

Girshick et al. [54] proposed the R-CNN for object localization, detection, and segmentation.

An R-CNN combines the strength of a CNN, the proposed region method, and a support vector machine (SVM). The proposed region method—a selective search—generates candidate region boxes, and the CNN extracts features from each box. Then, the SVM predicts a class and draws a

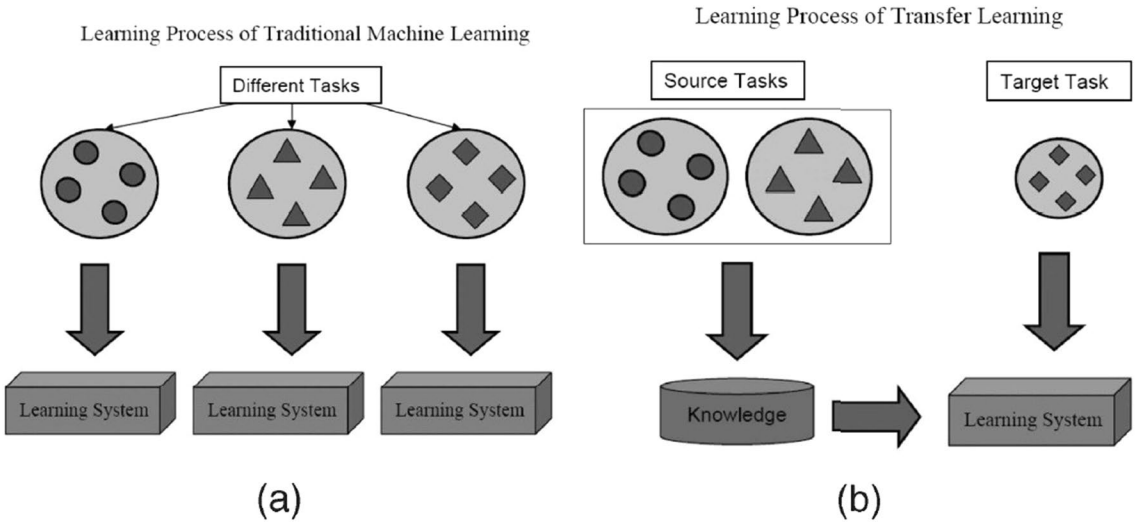

Fig. 9 The difference between: **a** traditional learning and **b** transfer learning [53]

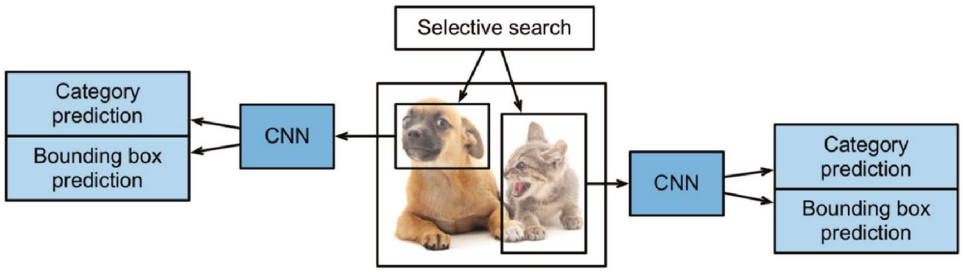

Fig. 10 R-CNN architecture [55]

bounding box for each candidate object. Figure 10 illustrates the R-CNN architecture.

The R-CNN approach achieved a mean average precision (mAP) of 53.3% at Pascal VOC 2012 [26] and 31.4% for the detection task at ILSVRC-2013. However, R-CNNs are slow and expensive because a CNN is run for each candidate region [56]. For example, if the region proposed method generates 2000 candidate boxes, then the CNN runs 2000 times to extract features of each box. As such, RCNNs require large amounts of memory to store feature maps.

Fast R-CNN Girshick [56] introduced the fast region-based convolutional neural network (Fast R-CNN) to improve the R-CNN speed. The input into Fast RCNN consists of an image and a set of proposed regions generated by a selective search (Fig. 11). Fast R-CNN applies one CNN to extract image features. After the CNN computes the feature maps, Fast R-CNN uses the ROI pooling layer to convert the size of each proposed region into a fixed length. The last layer of Fast R-CNN is a fully connected layer that divides into two branches for classification and bounding box predictions. Fast R-CNNs are faster than R-CNNs and decrease the training time from 83 to 9.5 h. Fast R-CNN improved the object detection by 16.7% in terms of the mAP at Pascal

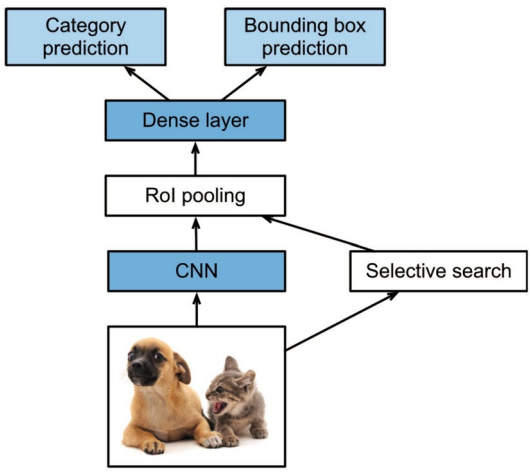

Fig. 11 Fast R-CNN architecture [55]

VOC 2012. However, Fast R-CNNs rely heavily on selective searching, which takes up most of the training time [57].

Faster R-CNN Ren et al. [57] proposed the faster region-based convolutional neural network (Faster R-CNN) to improve both the training speed and detection accuracy.

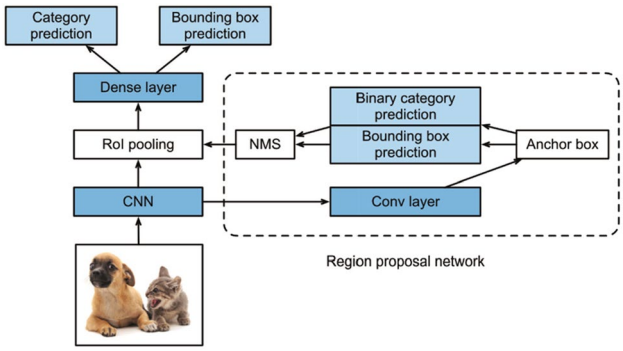

Fig. 12 Faster R-CNN architecture [55]

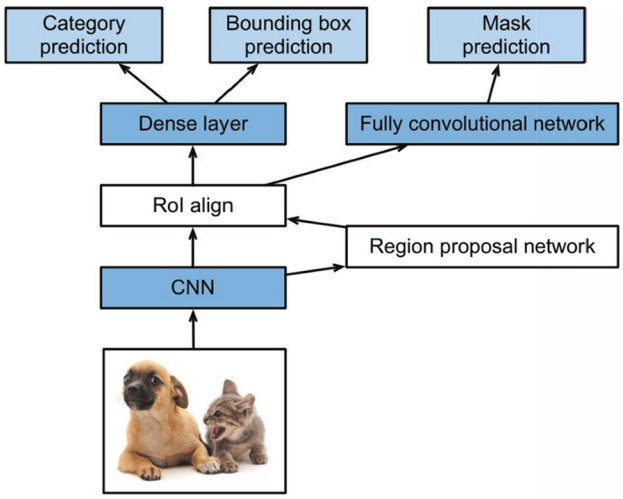

Fig. 13 Mask R-CNN architecture [55]

Faster R-CNN is similar to Fast RCNN but uses a region proposed network (RPN), which is a fully convolutional network (FCN) [58], instead of a selective search to generate proposed regions (Fig. 12). Unlike a selective search, an RPN is applied to feature maps and reduces the number of proposed regions by removing similar generated regions. An RPN generates anchor boxes (different sizes of boxes) and then predicts a binary classification (object or background) and bounding box for each anchor box. The RPN subsequently applies non-maximum suppression to remove similar bounding boxes. Thus, Faster RCNN achieved a mAP of 75% in the detection task at Pascal VOC 2012.

Mask R-CNN He et al. [59] introduced Mask R-CNN for instance segmentation, which is designed to detect an object while simultaneously generating a segmentation mask. Mask R-CNN has a mask branch added in parallel with Faster R-CNN (Fig. 13).

Fully Convolutional Network

Long et al. [58] proposed the FCN for semantic segmentation. An FCN is a CNN that replaces all dense layers with convolution layers and is divided into two parts: downsampling and upsampling paths. The downsampling path is a CNN (convolution layers, ReLU, and pooling layers) for feature extraction. The upsampling path contains transposed convolution layers (deconvolution) for recovering the spatial information of feature maps. The FCN utilizes the skip connection to preserve the spatial information in the early layers (Fig. 14). FCNs became state-of-the-art technology at PASCAL VOC2012 for segmentation tasks by achieving a mean IoU of 62.2%.

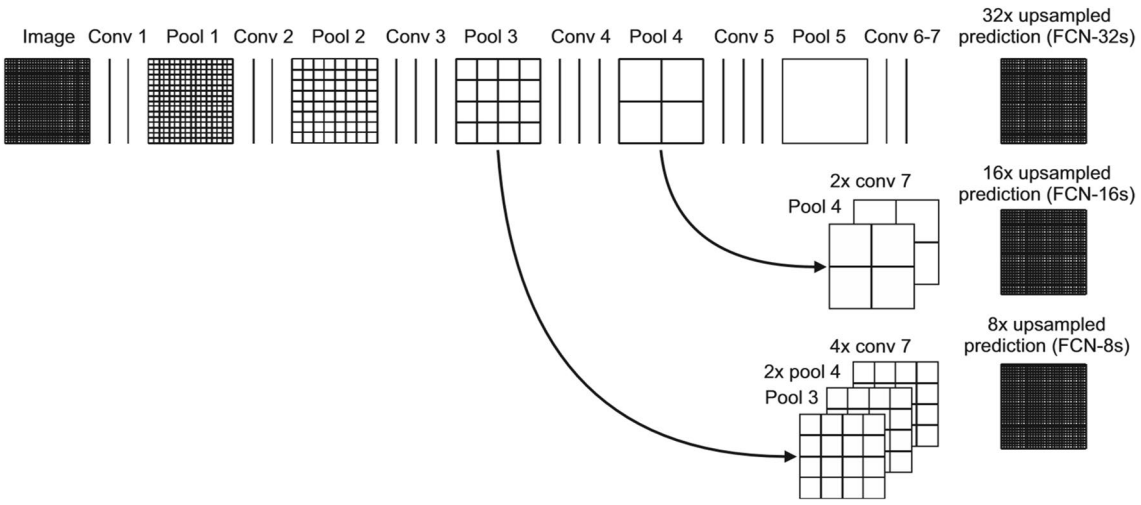

Fig. 14 FCN architecture [58]

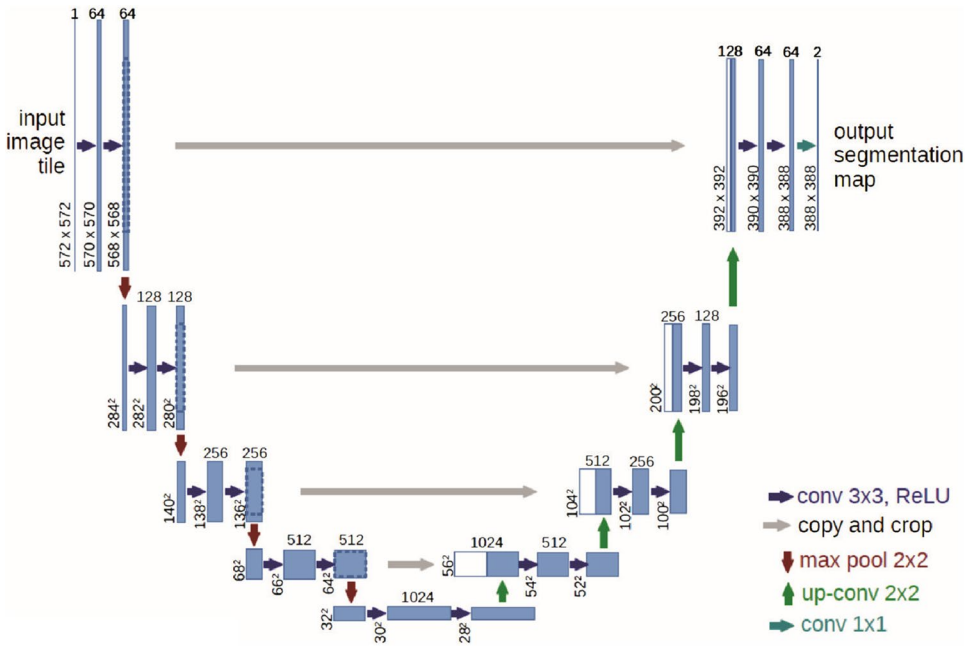

Fig. 15 U-Net architecture [60]

U-Net Ronneberger et al. [60] designed U-Net for biomedical semantic segmentation. U-Net has two symmetrical paths (contracting and expanding), creating a U-shape. Figure 15 shows the U-Net architecture. The contraction (downsampling) path is a traditional CNN for feature extraction. The expanding path (upsampling) is used to preserve spatial information. Both paths are connected by skip connection to preserve the spatial features from the early layers. U-Net is fast, taking less than 1 s to segment an image with dimensions of 512 × 512. U-Net proved to be a state-of-the-art method at the International Symposium on Biomedical Imaging (ISBI) 2015 [61] for 2D image segmentation by achieving 92% and 77% average IoUs on PhC-U373 and DIC-Hela datasets, respectively.

Medical Image Processing: Datasets and Applications

This section explores the available medical image datasets, and the supervised deep learning application in medical image processing.

Available Medical Image Datasets

A medical image dataset is the first component necessary to build an accurate automated diagnosis system using supervised learning. The dataset consists of inputs known as examples or instances. Each example has several attributes or features that are used to predict the desired output (target or label). For medical image datasets, the inputs are medical images, and the outputs are the diagnosis results (e.g., normal or abnormal). However, high-quality labeled medical image datasets are lack [5, 32, 34, 62–65], and the majority of publicly accessible medical image datasets have a small number of images [5].

Table 2 provides a list of the available medical image datasets for various diseases along with their download links, and some of these datasets were created for the medical tasks challenges. Table 3 summarizes the medical task challenges with their dataset.

Supervised Deep Learning Application in Medical Image Processing

Supervised Deep Learning Networks have been adopted to develop automated diagnosis systems for many medical image processing tasks for various diseases.

Brain

CNN and its versions were adopted for brain disease. Lu et al. [6] applied the AlexNet model for pathological brain detection in MRI images. They applied data augmentation and transfer learning to avoid overfitting. The model was trained for 2 min 17 s and achieved 100% accuracy. Toğaçar et al. [83] proposed a novel CNN with hypercolumn techniques and a feature selection approach for brain tumor MRI classification. First, they concatenated the features that are extracted by both pretrained AlexNet and VGG-16. They used hypercolumn techniques (instance segmentation) to retain the spatial information at the early layers of CNN, which improves the classification accuracy. They applied the recursive feature elimination (RFE) to select the most

Table 2 Medical image datasets for various diseases

Datasets	Type of image	Number of cases	Number of images	Download links
Brain datasets				
IXI dataset [66]	MRI	–	600	https://brain-development.org/ixi-dataset/
OASIS-3 [67]	MRI	1000 +	2000 +	https://www.oasis-brains.org/#data
Multimodal Brain Tumor Image Segmentation Benchmark [68]	MRI	2000	8000	Multimodal Brain Tumor Segmentation Challenge 2020: Data | CBICA | Perelman School of Medicine at the University of Pennsylvania (upenn.edu)
Breast cancer datasets				
BancoWeb LAPIMO database [69, 70]	Mammography	320	1473	http://lapimo.sel.eesc.usp.br/bancoweb/
Breast Cancer Histopathological Database (BreakHis) [70]	Microscopic	82	9109	https://web.inf.ufpr.br/vri/databases/breast-cancer-histopathological-database-breakhis/
Breast Cancer Wisconsin Dataset [70]	Mammography	–	569	https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)
INbreast dataset [69, 70]	Mammography	115	410	http://medicalresearch.inescporto.pt/breastresearch/index.php/Get_INbreast_Database
Mammographic Image Analysis Society (MIAS) [69, 71]	Mammography	161	322	https://www.mammoimage.org/databases/
Digital Database for Screening Mammography (DDSM) [66]	Mammography	2620	10,480	http://www.eng.usf.edu/cvprg/Mammography/Database.html
Cervical cancer datasets				
Intel and MobileODT on Kaggle [72]	Cervical cancer screening	–	1480	https://www.kaggle.com/c/intel-mobileodt-cervical-cancer-screening/data
ISBI Challenge Database [73]	Cytology image	–	16 real EDF image and 945 synthetic images	https://cs.adelaide.edu.au/~carneiro/isbi14_challenge/dataset.html
PAP Smear Benchmark Database [73]	Microscopic	–	917	http://mde-lab.aegean.gr/downloads
SIPaKMeD Database [73]	Pap smear slides	–	4049	https://www.cs.uoi.gr/~marina/sipakmed.html
Diabetic retinopathy datasets				
DiaretDB0 [74]	Fundus	–	130	https://www.it.lut.fi/project/imageret/diaretdb0/
DiaretDB1 [74]	Fundus	–	89	http://www2.it.lut.fi/project/imageret/diaretdb1/
E-Ophtha [74]	Fundus	–	381	http://www.adcis.net/en/third-party/e-ophtha/
Kaggle DR Challenge [74]	Fundus	–	88,702	https://www.kaggle.com/c/diabetic-retinopathy-detection/data
Messidor [74]	Fundus	–	1200	http://www.adcis.net/en/third-party/messidor/
Messidor-2 [74]	Fundus	–	1784	http://www.adcis.net/en/third-party/messidor2/
Lung cancer datasets				
Lung Image Database Consortium and Image Database (LIDC/IDRI) [75]	CT	1018	7371	https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI
National Lung Screening Trial (NLST) [75]	CT	53,454	75,000	https://cdas.cancer.gov/datasets/nlst/

Table 2 (continued)

Datasets	Type of image	Number of cases	Number of images	Download links
Skin disease datasets				
DermNet NZ [76]	Clinical	–	20,000 +	https://www.dermnetnz.org/
Dermofit Image Library [76]	Dermoscopic	–	1300	https://licensing.edinburgh-innovations.ed.ac.uk/i/software/dermofit-image-library.html
ISIC 2019 [76]	Dermoscopic	–	25,331	https://challenge2019.isic-archive.com/
PH^2 Dataset [76]	Dermoscopic	–	200	https://sites.google.com/site/robustmelanomascreening/dataset
Interactive Atlas of Dermoscopy (EDRA)	Dermoscopic	1000 +	2000 +	http://derm.cs.sfu.ca/Welcome.html
International Skin Imaging Collaboration (ISIC 2020) [76]	Dermoscopic	2000	33,126	https://challenge2020.isic-archive.com/

Table 3 Medical image processing task challenges

Medical Image processing task challenges	Medical dataset
Brain tumors segmentation in multimodal magnetic resonance imaging (MRI) scans [68]	Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)
Women's cervix type classification based on images [77]	Intel and MobileODT on Kaggle
Detection of Diabetic Retinopathy (DR) by Kaggle [78]	Kaggle DR Challenge
Lung nodule detection challenges [79]	Lung Image Database Consortium and Image Database (LIDC/IDRI)
International Skin Imaging Collaboration (ISIC) challenges for Skin lesion detection [80]	ISIC 2016, 2017, 2018, 2019, 2020 dataset
The Medical Segmentation Decathlon (MSD) [81]	MSD dataset
The COVID-19–20 Lung CT Lesion Segmentation Challenge [82]	CT images in COVID-19 and COVID-19-AR datasets

effective deep features. The support vector machine (SVM) was used to classify brain tumor MRIs as either benign or malignant. They used imbalance dataset (155 tumor and 98 normal) in this study. To overcome this problem, they utilized data augmentation to increase the number of normal samples to 155. Consequently, the proposed model yields an accuracy, sensitivity, and specificity of 96.77%, 97.83%, and 95.74%, respectively.

Ozyurt et al. [84] used CNN with the neutrosophy approach for brain tumor classification. The neutrosophic method was used to segment MRI brain images. Further, the AlexNet was applied to extract features from segmented images. SVM and k-nearest neighbors (KNN) were utilized to classify the extraction features as benign and malignant. The SVM classifier has the best performance, and it yields an accuracy of 95.62%, a sensitivity of 96.25%, and a specificity of 95%, whereas the KNN yields an accuracy, sensitivity, and specificity of 90.62%, 90%, and 91.25%, respectively.

Swati et al. [85] developed a model based on a pre-trained VGG19 and transfer learning for multiclass brain tumor classification on MRI images. The transfer learning was used to solve the overfitting problem that is associated with training VGG19 with small dataset. The transfer learning approach used in this study is a blockwise fine-tuning which divides VGG19 into six blocks (B1–B6) based on pooling layers. Further, they fine-tuned the last block (B6) of VGG19, and incrementally fine-tuned the early blocks of VGG19 to investigate the performance of shallow fine-tuning and deep fine-tuning. They found that the performance improved with the gradual fine-tuning of the early blocks of VGG19. Thus, the deep fine-tuning from B1 to B6 yielded the best performance (94.82%) while the shallow fine-tuning B6 has the worst accuracy of 86.81%.

They compared pre-trained AlexNet and VGG16 with VGG19, and they found VGG19 surpasses both pre-trained models by achieving an accuracy of 94.82% while pre-trained AlexNet and VGG16 have an accuracy of 89.95% and 94.65%, respectively.

They discussed the impact of the hyperparameters on the model performance and convergence, and they found that the learning and scheduling rates play an important role on model performance and convergence. If these values are very large, the model would fail to converge, and it showed

poor performance. If the values of learning and scheduling rates are small, the model would converge at a slow rate.

Breast

For breast cancer classification, Beevi et al. [34] used the VGG for mitosis detection in breast cancer in histopathological images. They adopted transfer learning to overcome the overfitting problem. The model achieved an accuracy of 94%.

Wei et al. [32] and Chang et al. [13] used the GoogleNet inception model. Wei et al. developed a model called BiCNN and used several techniques to train the BiCNN: from scratch as well as using data augmentation, fine-tuning transfer learning, and combining data augmentation with transfer learning. BiCNN achieved the best performance at 97% by combining data augmentation with transfer learning, whereas the worst performance was obtained by training GoogleNet from scratch (80%). Chang et al. adopted data augmentation and transfer learning to overcome the overfitting problem. Data augmentation increased the number of samples from 1398 to 11,184 images. The model achieved an accuracy of 86% and an AUC of 0.93.

Khan et al. [86] developed a model that was trained by concatenated features extracted from three pretrained models—GoogleNet, VGG, and ResNet—for detection and classification of breast cancer in cytological images. They used data augmentation along with transfer learning to overcome the overfitting problem; data augmentation increased the number of samples to 8,000 images. The proposed model showed a significant classification accuracy of 97.52% by transferring the learned features from multiple networks, whereas transferring them from individual networks produced accuracies of 94.6%, 94.2%, and 95.4% with GoogleNet, VGG, and ResNet, respectively.

Cervical and Oral

Wieslander et al. [3] used ResNet and a VGG to develop a model for oral and cervical cancer classification in microscopic images. They utilized data augmentation to solve the overfitting problem. ResNet surpassed the VGG in both oral and cervical classification. ResNet achieved an accuracy of 82.58% for oral cancer and 85.45% for cervical cancer, whereas the VGG had accuracies of 80.66% and 85.38% for oral and cervical cancer, respectively.

Ariji et al. [4] utilized AlexNet for oral cancer classification in CT. They adopted data augmentation to overcome the overfitting problem. AlexNet proved to have performance comparable to that of a radiologist. It reached an accuracy of 78.2%, where two radiologists with 20 years of experience achieved 83.1%.

In addition, Anantharaman et al. [87] employed GoogleNet to develop a mobile application for mouth sore (cold and canker) classification. They collected 75 images from Google, and dentists assigned labels to the images. The authors adopted transfer learning to solve the overfitting problem. The proposed model achieved 66% accuracy.

Anantharaman et al. [30] utilized Mask RCNN for cold and canker detection and segmentation. They collected their dataset from Google Images; then, a pathologist provided the ground truth annotation using VGG Image Annotator [88]. Mask R-CNN achieved average Dice coefficients of 0.74 and 0.71 for cold and canker, respectively.

Lung

CNNs and their pretrained networks have also been used for lung cancer classification in CT images. Li et al. [8] designed a model with a single convolutional layer as well as max pooling, dropout, and three fully connected layers. The model achieved 92% precision and 89% recall. Rao et al. [89] developed the CanNet model using convolutional, ReLU, max pooling, and dropout layers that were repeated twice, followed by two fully connected layers. They also designed a model using the architecture of LeNet-5. CanNet outperformed LeNet-5 by achieving 76% accuracy, whereas LeNet-5 yielded 56% accuracy. The authors indicated that the size of the dataset was unsuitable for training LeNet-5.

Fang [36] used AlexNet and GoogleNet for lung cancer classification. The author adopted data augmentation and transfer learning to avoid overfitting. A fine-tuned GoogleNet achieved 81% accuracy, surpassing AlexNet by 2.0%.

Hussein et al. [90] used AlexNet and Gaussian process (GP) regression [91] to develop a model known as TumorNet. AlexNet was used for feature extraction, and the GP was utilized for classification. The authors applied data augmentation to avoid overfitting. TumorNet achieved 92% regression accuracy and a standard error of the mean of 1.59.

Zhao et al. [92] combined 2D and 3D CNNs to design a model for pulmonary nodule detection. Firstly, a pretrained GoogleNet was used to generate candidate nodules. Then, a 3D CNN was applied to classify the candidate nodules. The model achieved 83% accuracy, 86% sensitivity, and 80% specificity.

Tang et al. [93] combined Faster R-CNN and a 3D CNN to develop an ensemble model for pulmonary nodule detection in CT images. The authors adopted Faster R-CNN to generate nodule candidates and utilized hard negative mining [94] to reduce the negative samples. Then, they applied the 3D CNN to classify the proposed nodule candidates. The proposed model achieved an average recall at seven predefined false positive points of 0.815.

Deep learning was applied for an early detection of coronavirus disease (COVID-19) during the global pandemic. Oh

et al. [95] trained a CNN on chest X-ray images for COVID-19 diagnosis. First, they applied full convolutional-DenseNet to extract lung and heart contours from chest images.

Then, a patch-based CNN was used for COVID-19 diagnosis. The proposed model achieved accuracy, precision, recall, and F1-score of 88.9%, 83.4%, 85.9%, and 84%, respectively.

Hu et al. [96] proposed a model based on a CNN for COVID-19 detection and classification using CT images. They achieved accuracy, precision, recall, and *F*1-score of 96.2%, 97.3%, 94.5%, and 95.3%, respectively.

Skin

Al-Masni et al. [25] proposed a deep learning network for multiple skin lesion segmentation and classification. First, they used a full resolution convolution network (FrCN) to segment the skin lesion on dermoscopy images. Then, they utilized different pre-trained CNNs including Inception-v3, ResNet-50, Inception-ResNetv2, and DenseNet-201 [97] for classifying segmented lesions into multiple classes. The proposed model was evaluated using the International Skin Imaging Collaboration (ISIC) dataset 2016, 2017, and 2018 which have binary, three, and seven classes of skin lesions, respectively. They applied the data augmentation to increase the number of samples, and in some cases, they performed up-sampling to solve the problem of imbalance dataset.

The balanced, segmented, and augmented datasets improve the *F*1-score by 8.05%. The *F*1-scores on ISIC 2016 are 78.39%, 80.85%, 82.59%, and 81.73% for Inception-v3, ResNet-50, Inception-ResNet-v2, and DenseNet-201, respectively. The ResNet-50 outperforms other networks on both ISIC 2017 and ISIC 2018 by achieving *F*1-scores of 75.75% and 81.28%, respectively, where Inception-v3, Inception-ResNet-v2, and DenseNet-201 yielded scores of 74.92% and 77.84%, 75.72% and 78.46%, and 65.93% and 79.47% on ISIC 2017 and ISIC 2018, respectively. The authors indicated that the segmentation and classification of skin lesions using deep learning networks become more complicated as the number of trained classes increases.

Kwasigroch et al. [98] utilized the CNN, specifically VGG19, ResNet, and VGG19 with the SVM for skin lesion classification. They applied upsampling and data augmentation to solve the problems of imbalance dataset and overfitting, respectively. The VGG19 surpasses ResNet and VGG19-SVM by achieving an accuracy of 81.2%, where ResNet and VGG19-SVM yield scores of 75.5% and 80.7%, respectively.

Liu et al. [99] utilized a CNN with mid-level feature learning for skin lesion classification. They applied a segmentation network (U-Net) [60] to extract the regions of interest (ROI) from skin lesion. They used the pretrained ResNet and DenseNet to extract features from ROI images. Further, they proposed a novel mid-level feature learning as feature representation based on the distance metric learning that describes the relationship between different classes of skin lesions. The metric learning was used to study the similarity between the samples images and a reference image set to separate different skin lesion classes. Thus, the ResNet surpass DenseNet by yielding an accuracy of 87.25%, where DenseNet yielded a score of 87%.

Other Diseases

A CNN and its pretrained networks were applied to microscopic images to develop an automated diagnosis system. Talo [2] used ResNet50 and DenseNet-161 [97] for histopathological image classification. The author adopted transfer learning and dropout layers to overcome the overfitting problem. The proposed model was trained on grayscale and color images. DenseNet161 yielded the highest performance on grayscale images (97.89% accuracy), ResNet-50 performed better on color images, achieving 98.87% accuracy. ResNet-50 was faster than DensNet-161 on both types of images.

Nguyen et al. [11] applied Inception v3 [24], ResNet 152, and ResNet-v2 [23] for microscopic image classification. The proposed model was trained by concatenated features extracted from the three networks. The model achieved 92.57% accuracy, whereas Inception v3, ResNet-152, and ResNet-v2 yielded 90%, 89%, and 92% accuracy, respectively.

Further, Li et al. [9] utilized AlexNet and GoogleNet for gland segmentation. They applied window sliding to divide each input image into slides, after which fine-tuned AlexNet and GoogleNet were used for feature extraction and classification. The fine-tuned AlexNet outperformed the fine-tuned GoogleNet by achieving 0.73 and 0.84 Jaccard and Dice coefficients, respectively, whereas GoogleNet yielded 0.72 and 0.83 Jaccard and Dice coefficients, respectively.

Mazo et al. [1] used VGG16, VGG19, GoogleNet, and ResNet for cardiovascular tissue classification in histopathological images. They applied transfer learning to overcome the overfitting problem. The pretrained ResNet outperformed VGG16, VGG19, and GoogleNet by achieving an F-score of 83%, where VGG16, VGG19, and GoogleNet produced F-scores of 0.82%, 0.81%, and 0.75%, respectively. The pretrained ResNet improved the precision by 5% and recall by 6% compared to the model trained from scratch.

Tajbakhsh et al. [35] investigated the performance of a fine-tuned AlexNet against a CNN trained from scratch by developing four models for various medical image processing tasks using three medical image modalities: colonoscopy classification on a colonoscopy video, intima-media segmentation, polyp detection on a colonoscopy video, and pulmonary embolism detection on a CT image. The

fine-tuned AlexNet surpassed the fully trained AlexNet by achieving ROCs of 0.7 and 0.98 for polyp detection and colonoscopy classification, respectively, as well as 24.71 localization error for intima-media segmentation. For pulmonary embolism detection, the ROCs of finetuned and fully trained AlexNet were similar, reaching 0.88. Therefore, the four models demonstrated the feasibility of transfer learning for medical image processing.

Sa et al. [7] used Faster R-CNN for intervertebral disc detection in X-ray images. They leveraged transfer learning to deal with small datasets and used two datasets with different sizes (92 and 1082 X-ray images). The fine-tuned Faster RCNN achieved APs of 0.65 and 0.90 in the small and large datasets, respectively, and the average detection time was 3 s per image.

Yang et al. [12] used Faster R-CNN to detect six cell classes (red blood, white blood, yeast, crystal, cast, and epithelium) in microscopic images. Faster R-CNN achieved an AP of greater than 0.90 for all categories, and it took only 0.07 s to detect the cells in each image.

Similarly, Mo et al. [100] applied Faster R-CNN for polyp detection in endoscopic videos. They used transfer learning to avoid overfitting. The fine-tuned Faster R-CNN achieved 86.2% precision, 98% recall, an $F1$-score of 91.7%, and 25 pixels as the mean Euclidean distance between polyp centers.

Chen et al. [101] utilized an FCN for gland and nucleus segmentation in histopathological images and proposed a deep contour aware network (DCAN). The DCAN was trained to generate a segmentation mask and simultaneously draw contour detection. A multi-level contextual feature FCN was used to deal with large appearance variations in the histological structure. An auxiliary classifier was applied to deal with the vanishing gradient problem. Contour information was integrated with the FCN to separate touching objects. DCAN achieved an overall $F1$-score of 0.88 (0.90 benign and 0.77 malignant), and it ranked first for segmentation tasks at both the 2015 MICCAI Gland Segmentation [80] and 2015 MICCAI Nuclei Segmentation Challenges.

Milletari et al. [102] developed a model known as V-Net based on U-Net for prostate segmentation in MRI. V-Net uses a 3D CNN instead of a 2D CNN as a traditional U-Net, and residual functions are applied in each stage. Max pooling is replaced by a convolution of $2\times2\times2$ voxels and stride of 2. V-Net achieved an average Dice coefficient of 0.87.

Vuola et al. [14] used U-Net along with MaskCNN for nucleus segmentation in microscope images. U-Net and Mask R-CNN were ensembled to obtain more accurate results—U-Net was designed for biomedical images semantic segmentation, whereas Mask R-CNN was developed for instance segmentation. Mask R-CNN provides good nucleus detection, and U-Net is more accurate in nucleus segmentation. U-net performs well on large nuclei, whereas Mask R-CNN is more effective for grouped nuclei. Combining these networks yielded accurate segmentation with a mAP of 0.523, whereas U-Net and Mask R-CNN separately reached mAPs of 0.515 and 0.519, respectively.

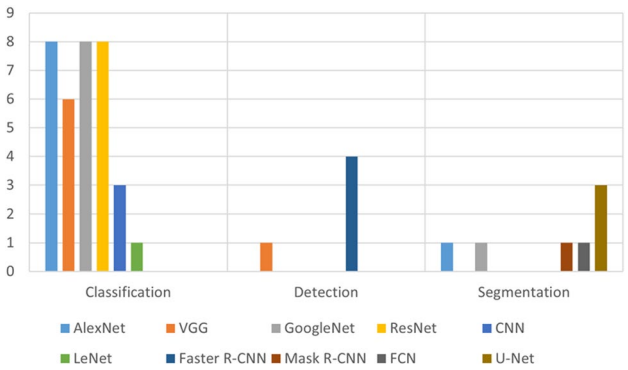

Fig. 16 Distribution of supervised learning techniques for various medical image tasks

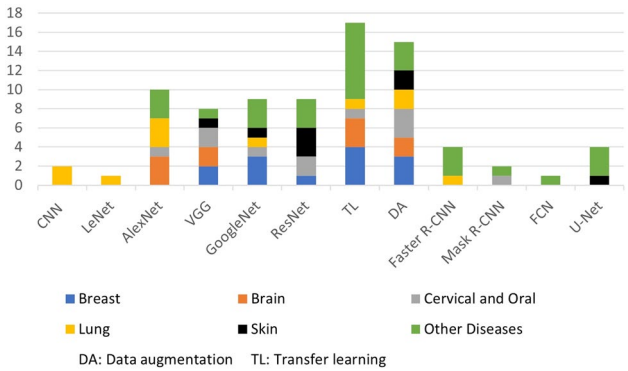

Fig. 17 Distribution of supervised learning techniques for various diseases

Discussions

Trends

We first discuss the overall trends in the distribution of supervised learning algorithms for each medical image processing task. As Fig. 16 represents, AlexNet, GoogleNet, and ResNet are the most frequently adapted networks for medical image classification. Faster R-CNN and U-Net networks are widely used for detection and segmentation tasks, respectively.

Then, we explore the use of different supervised learning methods for various diseases. Figure 17 shows the distribution of different diseases across supervised learning algorithms. Transfer learning and data augmentation are widely

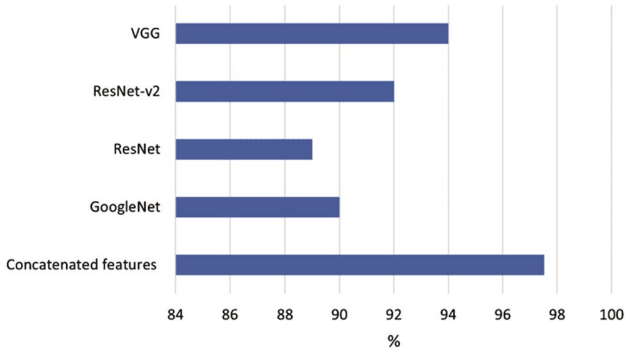

Fig. 18 Accuracies for transferred features from multiple and individual networks

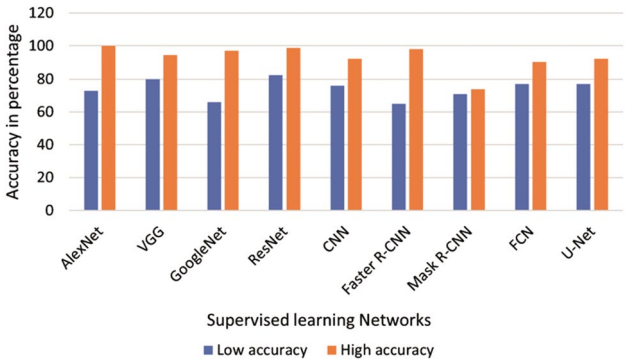

Fig. 19 Lowest and highest accuracy of supervised learning networks

used in medical image processing as training supervised architectures require large datasets and medical image datasets are scarce.

There are other trends related to transfer learning methods. Figure 18 shows the accuracy of transferred concatenated features extracted from multiple and individual architectures. The transferred features from multiple supervised learning architectures are more accurate than those from individual architectures.

Accuracy and Influence Factors

The accuracies of the supervised learning networks in medical image processing range from 60 to 100%. Figure 19 shows the highest and lowest accuracies for several supervised learning networks. GoogleNet and Faster R-CNN have the largest gap between their highest and lowest accuracy. The highest accuracy occurs when the number of samples is 7,909, with the combination of transfer learning and data augmentation [32]; meanwhile, the lowest accuracy occurs when the number of samples is so small and only transfer learning is utilized [87]. Faster R-CNN obtained 0.90 and

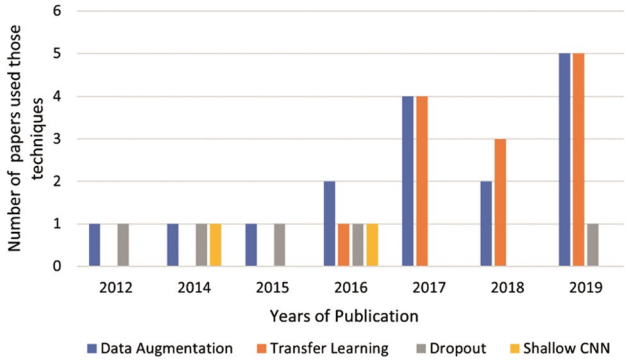

Fig. 20 Distributions of papers discussing techniques to overcome the overfitting problem over the years

0.65 of APs with transfer learning and a dataset of 1082 and 92 samples, respectively [7].

However, there are varied factors that affect the supervised learning algorithms' performance, and we do not consider them in Fig. 19. These factors include the number of samples [89], the number of layers [89, 103], the number of classes [25], the image modality and quality [104], and the hyperparameters values [85, 103]. For instance, AlexNet achieved the highest sensitivity and specificity of 100% in the binary classification of a brain tumor in MRI images [6]. In contracts, AlexNet reached 89% of sensitivity and 89.84% of specificity on multiclass brain tumors classification using MRI images [85]. Therefore, identifying a suitable supervised deep learning algorithms for medical images processing tasks is a significant challenging task.

Challenges and Issues

The big challenges of applying supervised learning in medical image processing is *the bottelnecks of medical images labelling*. Supervised learning requires a massive number of samples for good performance and robust generalization, and high-quality labeled medical image datasets are scarce [5, 32, 34, 62–64], and most publicly available medical image datasets consist of small numbers of patients [5]. Creating biomedical datasets is expensive and time consuming [34, 62–64]. Medical image requires clinician's interpretation for collecting, labelling, and annotating medical images. Furthermore, data collection involving human subjects requires privacy and ethical oversight by institutional review boards (IRBs).

However, data labelling bottlenecks can be addressed using several techniques, such as data augmentation, transfer learning, dropouts, and shallow CNNs. Figure 20 shows how the reviewed papers related to these techniques have been distributed over the years. The use of data augmentation increases from one paper in 2012 to five papers in

2019. Transfer learning was used in one paper in 2016 and subsequently became more frequently adopted to overcome the overfitting problem. The dropout approach was applied to solve this problem from 2012 to 2016, and again in 2019. The use of shallow CNNs is very rare.

Another significant issue in biomedical datasets is *data or class imbalance*. In imbalanced datasets, the class distribution is asymmetrical among the categories, and for instance, the cancer cell dataset is naturally imbalanced (the number of abnormal samples is more than the number of normal samples). With an imbalanced dataset, the model learns the attributes of the majority class more than the minority class [46]. Thus, imbalanced data significantly affect model performance. Data augmentation can be used to overcome the imbalanced dataset problem by increasing the number of samples. Lu et al. [6] had an imbalanced dataset of 38 normal brain MRI images and 177 pathological ones. They resolved this issue by utilizing data augmentation to increase the number of normal samples to 144 images and subsequently achieved brain classification with 100% accuracy. Ariji et al. [4] developed a model with a dataset that consisted of 127 positive metastatic lymph CT images and 314 negative samples. The numbers of positive and pathological samples were increased to 10,638 and 10,724, respectively, with data augmentation. Consequently, they achieved an accuracy of 78.2% (Fig. 20).

Digital medical imaging modalities are also challenging for applications of deep learning in medical image processing. For instance, the inputs for most state-of-the-art deep learning models are 2D images [102], although some medical image types are 3D images, (e.g., CT). An additional method is needed to avoid losing information from 3D medical images, such as computing the median intensity of multiview CT scans [36, 90], replacing a 2D CNN with a 3D CNN [93, 102], and incorporating a 2D CNN with a 3D CNN [92].

Microscopic images have various characteristics, including size, resolution, stain types, and enormous numbers of heterogenous and overlapping cells [33, 34]. The variations in the appearance of histopathological images increase the number of false positives, thereby affecting the model performance. Beevi et al. [34] utilized an optimal multi-thresholding known as the krill herd algorithm [105] to reduce the false positives for nucleus detection on histopathological images. They consequently achieved a high accuracy of 94%. Yousefi and Nie [33] used a class-agnostic detector, which detected several nuclei from histopathological images without knowing their class, then applied a CNN to assign classes to the detected nuclei with a 98% accuracy.

Conclusion

This survey article provides an overview of the application of supervised learning in medical image processing, focusing on classification, detection, and segmentation. We explained the performance matrices of supervised learning and summarized the available medical image datasets for various diseases. This study explored the various state-of-the-art supervised learning architectures, including CNNs and the corresponding pretrained algorithms (LeNet-5, AlexNet, ZFNet, VGG, GoogleNet, and ResNet), R-CNNs and their different versions (Fast R-CNN, Faster R-CNN, and Mask R-CNN), FCNs, and U-Nets. We discussed the challenges associated with applying supervised learning in medical image analysis. The literature demonstrated that data augmentation, transfer learning, and dropout techniques have widely been used in medical image processing to overcome the lack of labeled datasets as supervised learning needs large labeled datasets to learn and to achieve high performance. Supervised learning algorithms show promising results in medical image analysis that could be leveraged to improve the speed and accuracy of diagnosis for various diseases.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Mazo C, Bernal J, Trujillo M, Alegre E. Transfer learning for classification of cardiovascular tissues in histological images. Comput Methods Programs Biomed. 2018;165:69–76.
2. Talo M. Automated classification of histopathology images using transfer learning. Artif Intell Med. 2019;101: 101743.
3. Wieslander H, Forslid G, Bengtsson E, Wahlby C, Hirsch J-M, Runow Stark C, Kecheril Sadanandan S (2017) Deep convolutional neural networks for detecting cellular changes due to malignancy. In: Proceedings of the IEEE international conference on computer vision workshops, pp 82–89
4. Ariji Y, Fukuda M, Kise Y, Nozawa M, Yanashita Y, Fujita H, Katsumata A, Ariji E. Contrast-enhanced computed tomography image assessment of cervical lymph node metastasis in patients with oral cancer using a deep learning system of artificial intelligence. Oral Surg Oral Med Oral Pathol Oral Radiol. 2019;127(5):458–63.
5. Ker J, Wang L, Rao J, Lim T. Deep learning applications in medical image analysis. IEEE Access. 2017;6:9375–89.
6. Lu S, Lu Z, Zhang Y-D. Pathological brain detection based on alexnet and transfer learning. J Comput Sci. 2019;30:41–7.
7. Sa R, Owens W, Wiegand R, Studin M, Capoferri D, Barooha K, Greaux A, Rattray R, Hutton A, Cintineo J et al (2017) Intervertebral disc detection in x-ray images using faster r-cnn. In: 2017 39th annual international conference of the IEEE engineering in medicine and biology society (EMBC), pp. 564–567. IEEE

8. Li Q, Cai W, Wang X, Zhou Y, Feng DD, Chen M (2014) Medical image classification with convolutional neural network. In: 2014 13th international conference on control automation robotics & vision (ICARCV), pp 844–848. IEEE
9. Li W, Manivannan S, Akbar S, Zhang J, Trucco E, McKenna SJ (2016) Gland segmentation in colon histology images using hand-crafted features and convolutional neural networks. In: 2016 IEEE 13th international symposium on biomedical imaging (ISBI), pp 1405–1408. IEEE
10. Liu S, Liu S, Cai W, Pujol S, Kikinis R, Feng D (2014) Early diagnosis of Alzheimer's disease with deep learning. In: 2014 IEEE 11th international symposium on biomedical imaging (ISBI), pp 1015–1018. IEEE
11. Nguyen LD, Lin D, Lin Z, Cao J (2018) Deep cnns for microscopic image classification by exploiting transfer learning and feature concatenation. In: 2018 IEEE international symposium on circuits and systems (ISCAS), pp. 1–5. IEEE
12. Yang S, Fang B, Tang W, Wu X, Qian J, Yang W (2017) Faster r-cnn based microscopic cell detection. In: 2017 international conference on security, pattern analysis, and cybernetics (SPAC), pp 345–350. IEEE
13. Chang, J., Yu, J., Han, T., Chang, H.-j., Park, E.: A method for classifying medical images using transfer learning: A pilot study on histopathology of breast cancer. In: 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom), pp. 1–4 (2017). IEEE
14. Vuola AO, Akram SU, Kannala J (2019) Mask-rcnn and u-net ensembled for nuclei segmentation. In: 2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019), pp. 208–212. IEEE
15. Ghahramani Z. Unsupervised learning. In: Summer school on machine learning. New York: Springer; 2003. p. 72–112.
16. Das DK, Bose S, Maiti AK, Mitra B, Mukherjee G, Dutta PK. Automatic identification of clinically relevant regions from oral tissue histological images for oral squamous cell carcinoma diagnosis. Tissue Cell. 2018;53:111–9.
17. Montagnon E, Cerny M, Cadrin Ĉhenevert A, Hamilton V, Derennes T, Ilinca A, Vandenbroucke-Menu F, Turcotte S, Kadoury S, Tang A. Deep learning workflow in radiology: a primer. Insights Imaging. 2020;11(1):1–15.
18. Ruiz-Santaquiteria J, Bueno G, Deniz O, Vallez N, Cristobal G. Semantic versus instance segmentation in microscopic algae detection. Eng Appl Artif Intell. 2020;87: 103271.
19. Sokolova M, Japkowicz N, Szpakowicz S (2006) Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In: Australasian joint conference on artificial intelligence. Springer, New York, pp 1015–1021
20. Greiner M, Pfeiffer D, Smith R. Principles and practical application of the receiver-operating characteristic analysis for diagnostic tests. Prev Vet Med. 2000;45(1–2):23–41.
21. Seliya N, Khoshgoftaar TM, Van Hulse J (2009) A study on the relationships of classifier performance metrics. In: 2009 21st IEEE international conference on tools with artificial intelligence, pp 59–66. IEEE
22. Kulkarni A, Chong D, Batarseh FA. Foundations of data imbalance and solutions for a data democracy. In: Sps S, editor. Data democracy. Amsterdam: Elsevier; 2020. p. 83–106.
23. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-first AAAI Conference on Artificial Intelligence
24. 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
25. Al-Masni MA, Kim D-H, Kim TS. Multiple skin lesions diagnostics via integrated deep convolutional networks for segmentation and classification. Comput Methods Programs Biomed. 2020;190: 105351.
26. Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A. The pascal visual object classes (voc) challenge. Int J Comput Vision. 2010;88(2):303–38.
27. Padilla R, Netto SL, Da Silva EA (2020) A survey on performance metrics for object detection algorithms. In: 2020 International conference on systems, signals and image processing (IWSSIP), pp 237–242. IEEE
28. Shi R, Ngan KN, Li S (2014) Jaccard index compensation for object segmentation evaluation. In: 2014 IEEE international conference on image processing (ICIP), pp. 4457–4461. IEEE
29. Wang Z, Wang E, Zhu Y. Image segmentation evaluation: a survey of methods. Artif Intell Rev. 2020;53(8):5637–74.
30. Anantharaman R, Velazquez M, Lee Y (2018) Utilizing mask r-cnn for detection and segmentation of oral diseases. In: 2018 IEEE international conference on bioinformatics and biomedicine (BIBM), pp 2197–2204. IEEE
31. Aydin OU, Taha AA, Hilbert A, Khalil AA, Galinovic I, Fiebach JB, Frey D, Madai VI. On the usage of average hausdorff distance for segmentation performance assessment: hidden error when used for ranking. Eur Radiol Exp. 2021;5(1):1–7.
32. Wei B, Han Z, He X, Yin Y (2017) Deep learning model based breast cancer histopathological image classification. In: 2017 IEEE 2nd international conference on cloud computing and big data analysis (ICCCBDA), pp 348–353. IEEE
33. Yousefi S, Nie Y (2019) Transfer learning from nucleus detection to classification in histopathology images. In: 2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019), pp. 957–960. IEEE
34. Beevi KS, Nair MS, Bindu G. Automatic mitosis detection in breast histopathology images using convolutional neural network based deep transfer learning. Biocybern Biomed Eng. 2019;39(1):214–23.
35. Tajbakhsh N, Shin JY, Gurudu SR, Hurst RT, Kendall CB, Gotway MB, Liang J. Convolutional neural networks for medical image analysis: Full training or fine tuning? IEEE Trans Med Imaging. 2016;35(5):1299–312.
36. Fang T (2018) A novel computer-aided lung cancer detection method based on transfer learning from googlenet and median intensity projections. In: 2018 IEEE international conference on computer and communication engineering technology (CCET), pp 286–290. IEEE
37. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
38. Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In: European conference on computer vision. New York: Springer; 2014. p. 818–33.
39. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst. 2012;25:1097–105.
40. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning, pp. 448–456. PMLR
41. Salimans T, Kingma DP. Weight normalization: a simple reparameterization to accelerate training of deep neural networks. Adv Neural Inf Process Syst. 2016;29:901–9.
42. Ba JL, Kiros JR, Hinton GE (2016) Layer normalization. arXiv preprint arXiv:1607.06450

43. Qiao S, Wang H, Liu C, Shen W, Yuille A (2019) Micro-batch training with batchchannel normalization and weight standardization. arXiv preprint arXiv:1903.10520
44. 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
45. Sun M, Song Z, Jiang X, Pan J, Pang Y. Learning pooling for convolutional neural network. Neurocomputing. 2017;224:96–104.
46. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580
47. Ma W, Lu J (2017) An equivalence of fully connected layer and convolutional layer. arXiv preprint arXiv:1712.01252
48. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, et al. Imagenet large scale visual recognition challenge. Int J Comput Vision. 2015;115(3):211–52.
49. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86(11):2278–324.
50. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255. IEEE
51. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
52. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Doll'ar P, Zitnick CL. Microsoft coco: common objects in context. In: European conference on computer vision. New York: Springer; 2014. p. 740–55.
53. Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng. 2009;22(10):1345–59.
54. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
55. Zhang A, Lipton ZC, Li M, Smola AJ (2021) Dive into deep learning. arXiv preprint arXiv:2106.11342
56. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp. 1440–1448
57. Ren S, He K, Girshick R, Sun J. Faster r-cnn: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 2016;39(6):1137–49.
58. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440
59. He K, Gkioxari G, Doll'ar P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969
60. Ronneberger O, Fischer P, Brox T. Unet: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention. New York: Springer; 2015. p. 234–41.
61. Arganda-Carreras I, Seung S, Cardona A, Schindelin J (2012) ISBI challenge: segmentation of neuronal structures in EM stacks
62. Chamberlain D, Kodgule R, Ganelin D, Miglani V, Fletcher RR (2016) Application of semi-supervised deep learning to lung sound analysis. In: 2016 38th annual international conference of the IEEE engineering in medicine and biology society (EMBC), pp. 804–807. IEEE
63. Cheplygina V, de Bruijne M, Pluim JP. Not-so-supervised: a survey of semisupervised, multi-instance, and transfer learning in medical image analysis. Med Image Anal. 2019;54:280–96.
64. Madani A, Moradi M, Karargyris A, Syeda-Mahmood T (2018) Semi-supervised learning with generative adversarial networks for chest x-ray classification with ability of data domain adaptation. In: 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018), pp 1038–1042. IEEE
65. Van Engelen JE, Hoos HH. A survey on semi-supervised learning. Mach Learn. 2020;109(2):373–440.
66. Monté-Rubio GC, Falcón C, PomarolClotet E, Ashburner J. A comparison of various mri feature types for characterizing whole brain anatomical differences using linear pattern recognition methods. Neuroimage. 2018;178:753–68.
67. Frau-Pascual A, Augustinack J, Varadarajan D, Yendiki A, Fischl B, Aganj I (2019) Detecting structural brain connectivity differences in dementia through a conductance model. In: 2019 53rd Asilomar conference on signals, systems, and computers, pp. 591–595. IEEE
68. Baid U, Ghodasara S, Mohan S, Bilello M, Calabrese E, Colak E, Farahani K, Kalpathy-Cramer J, Kitamura FC, Pati S et al (2021) The rsna-asnr-miccai brats 2021 benchmark on brain tumor segmentation and radiogenomic classification. arXiv preprint arXiv:2107.02314
69. Moreira IC, Amaral I, Domingues I, Cardoso A, Cardoso MJ, Cardoso JS. Inbreast: toward a full-field digital mammographic database. Acad Radiol. 2012;19(2):236–48.
70. Mahmood T, Li J, Pei Y, Akhtar F, Imran A, Rehman KU. A brief survey on breast cancer diagnostic with deep learning schemes using multi-image modalities. IEEE Access. 2020;8:165779–809.
71. Sravan V, Swaraja K, Meenakshi K, Kora P, Samson M (2020) Magnetic resonance images based brain tumor segmentation-a critical survey. In: 2020 4th international conference on trends in electronics and informatics (ICOEI) (48184), pp 1063–1068. IEEE
72. Aina OE, Adeshina SA, Aibinu A (2019) Deep learning for image-based cervical cancer detection and diagnosis—a survey. In: 2019 15th international conference on electronics, computer and computation (ICECCO), pp 1–7. IEEE
73. Rahaman MM, Li C, Wu X, Yao Y, Hu Z, Jiang T, Li X, Qi S. A survey for cervical cytopathology image analysis using deep learning. IEEE Access. 2020;8:61687–710.
74. Sarhan MH, Nasseri MA, Zapp D, Maier M, Lohmann CP, Navab N, Eslami A. Machine learning techniques for ophthalmic data processing: a review. IEEE J Biomed Health Inform. 2020;24(12):3338–50.
75. Monkam P, Qi S, Ma H, Gao W, Yao Y, Qian W. Detection and classification of pulmonary nodules using convolutional neural networks: a survey. IEEE Access. 2019;7:78075–91.
76. Goswami T, Dabhi VK, Prajapati HB (2020) Skin disease classification from image-a survey. In: 2020 6th international conference on advanced computing and communication systems (ICACCS), pp. 599–605. IEEE
77. Kaggle (2021) "Which cancer treatment will be most effective?" Intel & MobileODT Cervical Cancer Screening. https://www.kaggle.com/c/intel-mobileodt-cervical-cancer-screening. Accessed 1 May 2021
78. Kaggle (2021) "Identify signs of diabetic retinopathy in eye images." Diabetic Retinopathy Detection. https://www.kaggle.com/c/diabetic-retinopathy-detection/overview. Accessed 1 May 2021
79. Setio AAA, Traverso A, De Bel T, Berens MS, Van Den Bogaard C, Cerello P, Chen H, Dou Q, Fantacci ME, Geurts B, et al. Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the luna16 challenge. Med Image Anal. 2017;42:1–13.
80. ISIC (2021) "The 2020 Live Challenge is open!" ISIC Challenge. https://challenge.isic-archive.com/. Accessed 1 May 2021

81. Antonelli M, Reinke A, Bakas S, Farahani K, Landman BA, Litjens G, Menze B, Ronneberger O, Summers RM, van Ginneken B et al (2021) The medical segmentation decathlon. arXiv preprint arXiv:2106.05735
82. Roth H, Xu Z, Diez CT, Jacob RS, Zember J, Molto J, Li W, Xu S, Turkbey B, Turkbey E et al (2021) Rapid artificial intelligence solutions in a pandemic-the covid-19–20 lung ct lesion segmentation challenge
83. Toğaçar M, Cömert Z. Ergen B (2020) Classification of brain mri using hyper column technique with convolutional neural network and feature selection method. Expert Syst Appl. 2020;149:113274.
84. Ozyurt F, Sert E, Avci E, Dogantekin E. Brain tumor detection based on convolutional neural network with neutrosophic expert maximum fuzzy sure entropy. Measurement. 2019;147:106830.
85. Swati ZNK, Zhao Q, Kabir M, Ali F, Ali Z, Ahmed S, Lu J. Brain tumor classification for mr images using transfer learning and fine-tuning. Comput Med Imaging Graph. 2019;75:34–46.
86. Khan S, Islam N, Jan Z, Din IU, Rodrigues JJC. A novel deep learning based framework for the detection and classification of breast cancer using transfer learning. Pattern Recogn Lett. 2019;125:1–6.
87. Anantharaman R, Anantharaman V, Lee Y (2017) Oro vision: deep learning for classifying orofacial diseases. In: 2017 IEEE international conference on healthcare informatics (ICHI), pp 39–45. IEEE
88. Dutta A, Gupta A, Zissermann A (2016) Vgg image annotator (via). http://www.robots.ox.ac.uk/˜vgg/software/via
89. Rao P, Pereira NA, Srinivasan R (2016) Convolutional neural networks for lung cancer screening in computed tomography (ct) scans. In: 2016 2nd international conference on contemporary computing and informatics (IC3I), pp 489–493. IEEE
90. Hussein S, Gillies R, Cao K, Song Q, Bagci U (2017) Tumornet: Lung nodule characterization using multi-view convolutional neural network with Gaussian process. In: 2017 IEEE 14th international symposium on biomedical imaging (ISBI 2017), pp 1007–1010. IEEE
91. Wilson AG, Knowles DA, Ghahramani Z (2011) Gaussian process regression networks. arXiv preprint arXiv:1110.4411
92. Zhao A, Deng J, Zhong L, Duan X, Zhang J, Peng Y (2019) Research on automatic detection algorithm of pulmonary nodules based on deep learning. In: 2019 4th international conference on mechanical, control and computer engineering (ICMCCE), pp 893–8934. IEEE
93. Tang H, Kim DR, Xie X (2018) Automated pulmonary nodule detection using 3d deep convolutional neural networks. In: 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018), pp 523–526 (2018). IEEE
94. Shrivastava A, Gupta A, Girshick R (2016) Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 761–769 (2016)
95. Oh Y, Park S, Ye JC. Deep learning covid-19 features on cxr using limited training data sets. IEEE Trans Med Imaging. 2020;39(8):2688–700.
96. Hu S, Gao Y, Niu Z, Jiang Y, Li L, Xiao X, Wang M, Fang EF, MenpesSmith W, Xia J, et al. Weakly supervised deep learning for covid-19 infection detection and classification from ct images. IEEE Access. 2020;8:118869–83.
97. 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
98. Kwasigroch A, Mikolajczyk A, Grochowski M (2017) Deep neural networks approach to skin lesions classification—a comparative analysis. In: 2017 22nd international conference on methods and models in automation and robotics (MMAR), pp 1069–1074. IEEE
99. Liu L, Mou L, Zhu XX, Mandal M. Automatic skin lesion classification based on mid-level feature learning. Comput Med Imaging Graph. 2020;84: 101765.
100. Mo X, Tao K, Wang Q, Wang G (2018) An efficient approach for polyps detection in endoscopic videos based on faster r-cnn. In: 2018 24th international conference on pattern recognition (ICPR), pp 3929–3934. IEEE
101. Chen H, Qi X, Yu L, Dou Q, Qin J, Heng P-A. Dcan: deep contour-aware networks for object instance segmentation from histology images. Med Image Anal. 2017;36:135–46.
102. Milletari F, Navab N, Ahmadi S-A (2016) Vnet: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 fourth international conference on 3D vision (3DV), pp 565–571. IEEE
103. Al Qasem O, Akour M, Alenezi M. The influence of deep learning algorithms factors in software fault prediction. IEEE Access. 2020;8:63945–60.
104. Kugelman J, Alonso-Caneiro D, Read SA, Vincent SJ, Chen FK, Collins MJ. Effect of altered oct image quality on deep learning boundary segmentation. IEEE Access. 2020;8:43537–53.
105. Gandomi AH, Alavi AH. Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numerical Simul. 2012;17(12):4831–45.