Deep Learning: An Update for Radiologists

Phillip M. Cheng, MD, MS
Emmanuel Montagnon, PhD
Rikiya Yamashita, MD, PhD
Ian Pan, MD[1]
Alexandre Cadrin-Chênevert, B.Ing, MD
Francisco Perdigón Romero, MSc
Gabriel Chartrand, PhD
Samuel Kadoury, PhD
An Tang, MD, MSc

Deep learning is a class of machine learning methods that has been successful in computer vision. Unlike traditional machine learning methods that require hand-engineered feature extraction from input images, deep learning methods learn the image features by which to classify data. Convolutional neural networks (CNNs), the core of deep learning methods for imaging, are multilayered artificial neural networks with weighted connections between neurons that are iteratively adjusted through repeated exposure to training data. These networks have numerous applications in radiology, particularly in image classification, object detection, semantic segmentation, and instance segmentation. The authors provide an update on a recent primer on deep learning for radiologists, and they review terminology, data requirements, and recent trends in the design of CNNs; illustrate building blocks and architectures adapted to computer vision tasks, including generative architectures; and discuss training and validation, performance metrics, visualization, and future directions. Familiarity with the key concepts described will help radiologists understand advances of deep learning in medical imaging and facilitate clinical adoption of these techniques.

Online supplemental material is available for this article.

Abbreviations: CNN = convolutional neural network, GAN = generative adversarial network, R-CNN = regions with CNN features, ROC = receiver operating characteristic

RadioGraphics 2021; 41:1427–1445

https://doi.org/10.1148/rg.2021200210

Content Codes: AI IN

From the Department of Radiology, Keck School of Medicine of the University of Southern California, Los Angeles, Calif (P.M.C.); Research Center (E.M., F.P.R., S.K., A.T.) and Department of Radiology (A.T.), Centre Hospitalier de l'Université de Montréal, 1058-2117 rue Saint-Denis, Montréal, QC, Canada H2X 3J4; Department of Biomedical Data Science, Stanford University School of Medicine, Stanford, Calif (R.Y.); Warren Alpert Medical School, Brown University, Providence, RI (I.P.); Department of Medical Imaging, CISSS Lanaudière, Université Laval, Joliette, Québec, Canada (A.C.C., S.K.); École Polytechnique, Montréal, Québec, Canada (F.P.R.); and AFX Medical, Montréal, Québec, Canada (G.C.). Presented as an education exhibit at the 2020 RSNA Annual Meeting. Received October 25, 2020; revision requested April 14, 2021, and received May 2; accepted May 7. For this journal-based SA-CME activity, the authors I.P. and A. T. have provided disclosures (see end of article); all other authors, the editor, and the reviewers have disclosed no relevant relationships. **Address correspondence to** A.T. (e-mail: *an.tang@umontreal.ca*).

1Current address: Department of Radiology, Brigham and Women's Hospital, Boston, Mass.

SA-CME LEARNING OBJECTIVES

After completing this journal-based SA-CME activity, participants will be able to:

- Differentiate among four computer vision tasks using deep learning techniques on radiologic images: classification, detection, semantic segmentation, and instance segmentation.
- Identify building blocks that constitute components of more complex neural network architectures.
- Discuss neural network architectures adapted to different computer vision tasks.

See rsna.org/learning-center-rg.

TEACHING POINTS

- Four key computer vision tasks for which deep learning models have been applied to medical images are classification, object detection, semantic segmentation, and instance segmentation.
- Medical images need labels to be used for supervised learning, the most common form of machine learning, in which the goal is to predict labels for new inputs. Depending on the task, labels for classification may arise from radiology reports, expert reviews, or clinical or pathologic data.
- Classification networks are the simplest deep learning architectures, as their goal is simply to predict a category for an image. However, refinements of these architectures have translated into improvements in other applications as well, as the basic structures of these networks are often used as building blocks of more complex architectures.
- Detection architectures build on the architectural innovations of CNNs, often incorporating the backbone of a trained classification network. However, detection architectures must not only classify objects in an image but also predict the coordinates of bounding boxes that localize the detected objects.
- Architectures for segmentation tasks such as semantic segmentation and instance segmentation must label every pixel in an image.

Introduction

Deep learning is a subfield of artificial intelligence that has achieved recent success and popularity for many complex problems (1,2). The breakthrough performance gains of deep learning systems in automated image analysis tasks have a variety of direct applications and implications for radiology (3). In a previous article, Chartrand et al (4) reviewed the basic concepts underlying deep learning. We recommend referring to that article as an accessible introduction to the basic concepts. This article expands on the topics described in the prior article, with a deeper discussion of more recent and advanced topics.

Briefly, deep learning systems for imaging use multilayer neural networks to transform input images into useful outputs. A deep learning system learns not only the mappings of image features to the outputs but also the image features themselves. Example outputs include image categories (for image classification), object locations (for detection), and pixel labels (for segmentation). For image analysis, the fundamental architecture of deep learning systems is the convolutional neural network (CNN). A CNN designed for images contains convolutional layers that compare overlapping rectangular patches of the input to small learnable weight matrices (termed *kernels* or *filters*) that encode features.

Neural network architectures have rapidly evolved in size, complexity, and applications since the breakthrough performances of early CNNs in image classification. In this article, we review data requirements for training deep learning models, architectural building blocks that compose modern neural network architectures, the validation process for testing deep learning systems for radiology applications, and future directions in the field.

Definitions

Four key computer vision tasks for which deep learning models have been applied to medical images are classification, object detection, semantic segmentation, and instance segmentation (Fig 1).

Image Classification

Image classification is the task of predicting the class or label of an entire image and can be binary (two classes) or multiclass (more than two). An example is the binary classification of normal versus diseased chest radiographs.

Object Detection

Object detection refers to the identification and localization of individual examples of a specific entity of interest on an image or volume, such as the detection and localization of liver metastases on a CT image. An object detection algorithm typically specifies the location and spatial extent of detected objects with a rectangular box surrounding the object (bounding box).

Semantic Segmentation

Semantic segmentation assigns each pixel in an image to a specific class. For example, each pixel in the liver could be assigned to parenchyma, tumor, or blood vessel. The output of this task would be a binary (black and white) image mask for each class, in which a pixel is "on" if it belongs to that class.

Instance Segmentation

Instance segmentation is the pixel-level detection and delineation of multiple objects within the same class, such as lung nodules individually distinguished on a chest radiograph. In contrast to semantic segmentation, instance segmentation requires an object detection step to separate the different objects (instances) of the same class.

Data

Training an effective CNN is dependent on labeled data. In classification, the data are images with category labels. In detection, the data are images and rectangular bounding box coordinates delimiting features of interest. In segmentation, the data are images and image masks that provide labels for each pixel or voxel.

Preparing medical image data for machine learning tasks is a complex process that has been

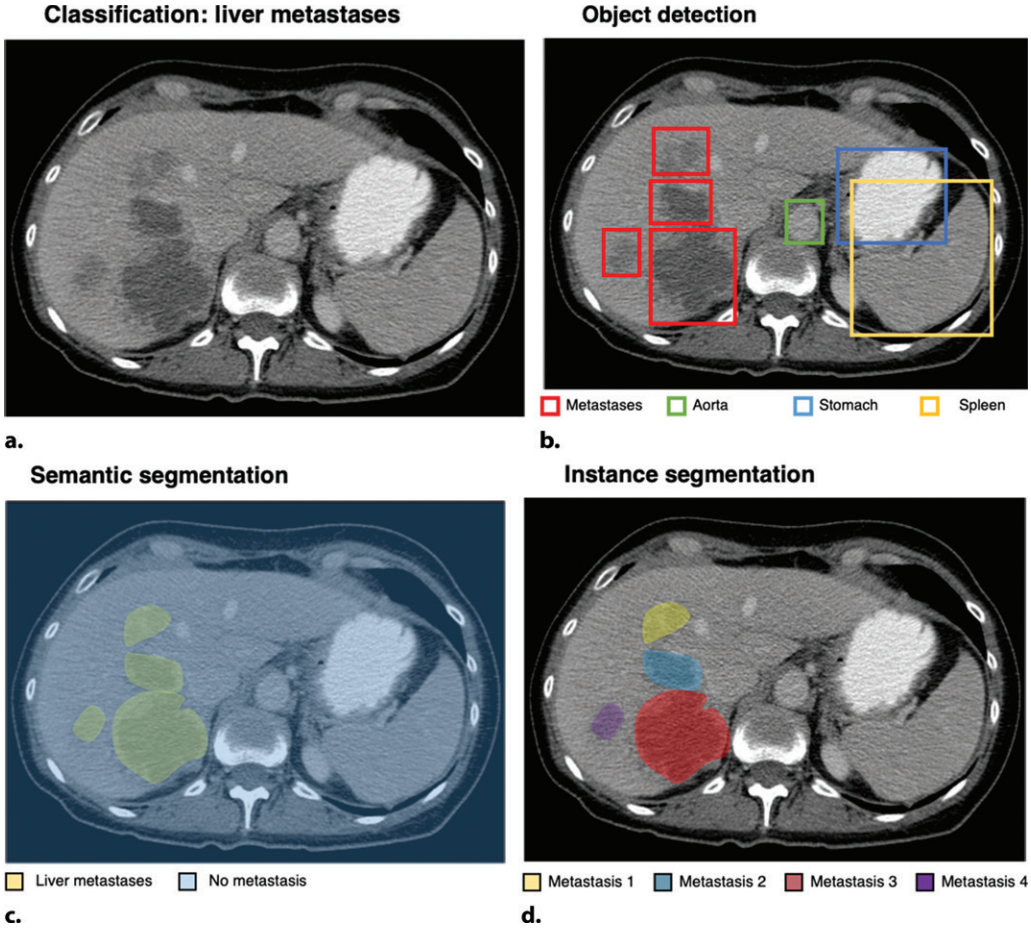


Figure 1. Computer vision tasks as depicted on axial contrast-enhanced CT images. **(a)** Classification aims to assign a label from a list to a given image (eg, liver metastases). **(b)** Object detection aims to locate lesions, structures, or organs (eg, liver metastases are in red squares, the aorta is in a green square, the stomach is in a blue square, and the spleen is in a yellow square). **(c)** Semantic segmentation assigns an object category label to each pixel in the image (eg, all liver metastases are in yellow). **(d)** Instance segmentation assigns individual labels to each pixel in the image (eg, individual liver metastases are segmented in red, blue, purple, and yellow).

reviewed in detail (5,6). For deep learning, it is critical to have training images that are representative of the task to be solved. Images from a single medical center may be insufficient to train a model for a given task or may be biased because of the sampled population. Multicenter datasets help to address these problems but introduce challenges related to privacy as well as standardization of image acquisition and labels.

With limited data, it is easy for a model to be trained to the point of predicting labels perfectly on the training data but poorly on new data; such a model is said to overfit the training set (7) or to exhibit poor generalization. One common way to expand the training dataset to prevent overfitting is image augmentation (Fig 2). Simple methods of increasing the number of training images include random translations, rotations, flips, scalings, crops, and brightness and contrast adjustments. There has also been interest in generative adversarial networks (GANs) (discussed further in this article) to produce fake images that resemble real images (9).

Medical images need labels to be used for supervised learning, the most common form of machine learning, in which the goal is to predict labels for new inputs. Depending on the task, labels for classification may arise from radiology reports, expert reviews, or clinical or pathologic data. Labels for detection and segmentation tasks are more complicated and time-consuming to create compared with classification datasets. Distributing the labeling task among more human labelers reduces the labeling burden on individuals but increases overall labeling work and raises consistency issues that may require averaged or consensus labels among several labelers. Recent experiments have found value in crowdsourced segmentation labels by nonexpert reviewers (10,11). For tasks with abundant imaging data, low-quality labels may be sufficient to train a network. Weak supervision describes training on such low-quality or noisy labels, as may arise from natural language processing of radiology reports (12).

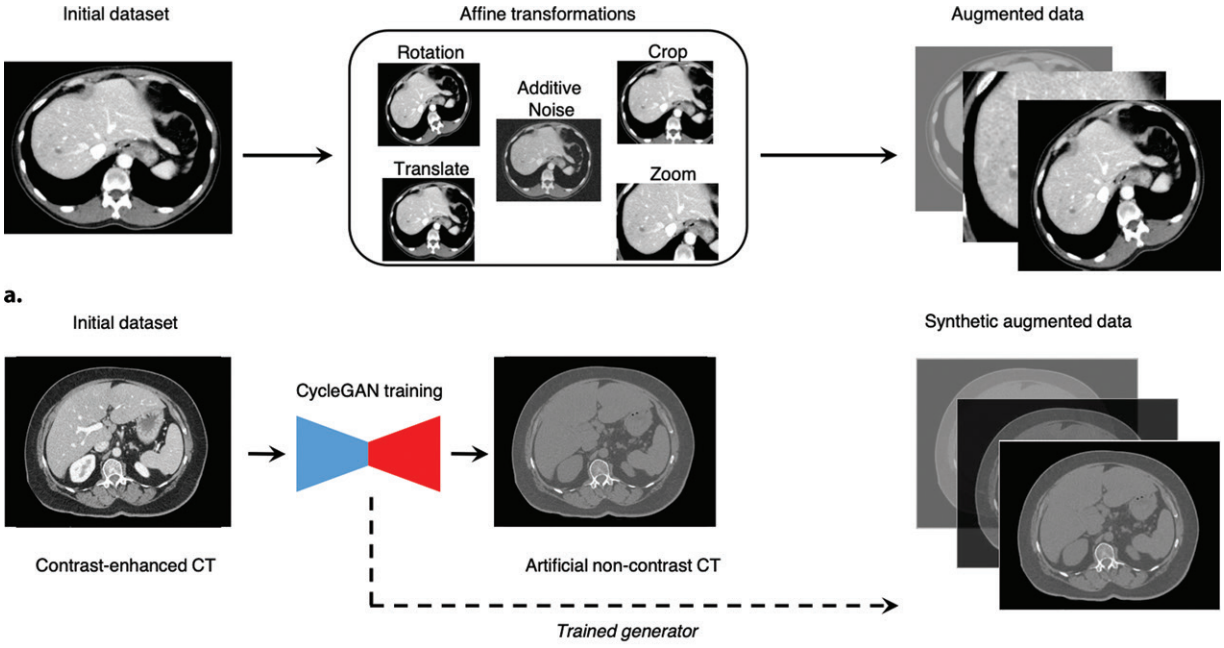


Figure 2. Diagrams demonstrate data augmentation. **(a)** Classic data augmentation consists of applying various transformations (random translations, rotations, flips, scalings, crops, and brightness and contrast adjustments) to initial CT images and using these new CT images for training. **(b)** Synthetic data augmentation uses a generative adversarial network (GAN) to produce additional synthetic images that have a statistical distribution similar to that of the initial dataset. In this example, a CycleGAN is trained to convert contrast-enhanced CT images to noncontrast images. The trained generator is then used to augment the initial dataset for training on a task segmenting noncontrast images, as proposed in reference 8.

Since the labeling process is expensive, semisupervised learning methods use unlabeled images to augment the dataset, allowing the network to learn more about the underlying structure of unseen data. The simplest semisupervised method is pseudo-labeling, whereby a partially trained model predicts labels (termed *pseudo-labels*) for the unlabeled data, and these pseudo-labeled images are then incorporated into further training (13).

Innovations in image augmentation and labeling cannot fully replace the need for labeled real image datasets with sufficient variations in subject or lesion appearance. Despite barriers in sharing medical image data, there have been increasing examples of public medical image datasets. Some prominent datasets are listed by the Data Science Institute at the American College of Radiology (14) and the Cancer Imaging Archive (15).

Convolutional Neural Networks

Toward Deeper Networks

One of the defining features of deep CNNs is the number of hidden layers within the networks (Fig 3). Shortly after the groundbreaking performance of AlexNet (17) in the 2012 ImageNet Challenge, many networks have been designed to improve its performance, with a trend toward larger and deeper neural networks (Fig 4). The increase in layers has been postulated to increase the capacity of a network to learn complex features (Fig 5). However, deeper networks can be more difficult to train, and the addition of layers has been observed to lead to performance degradation and higher training error (23,24). Further architectural refinements were required to improve model training and performance, as detailed further in this article.

Skip Connections

Skip connections are shortcut connections from one layer to a deeper layer, skipping one or more layers (Fig 6). A skip connection typically adds or concatenates the output of a shallower layer with the output of a deeper layer. These connections were empirically found to improve training of very deep neural networks, starting with the residual neural network (ResNet) (23). The informal intuition behind these connections is that they allow the skipped layers to fit a residual or error mapping, which may be easier than training those layers to fit a more complex full mapping. Further analysis has shown that skip connections facilitate training by eliminating large irregularities in the shape of the loss function, which measures the output error of the model (25).

Bottleneck Blocks

Bottlenecks in neural networks improve computational efficiency by reducing the number of feature maps (Fig 7). A feature map or channel

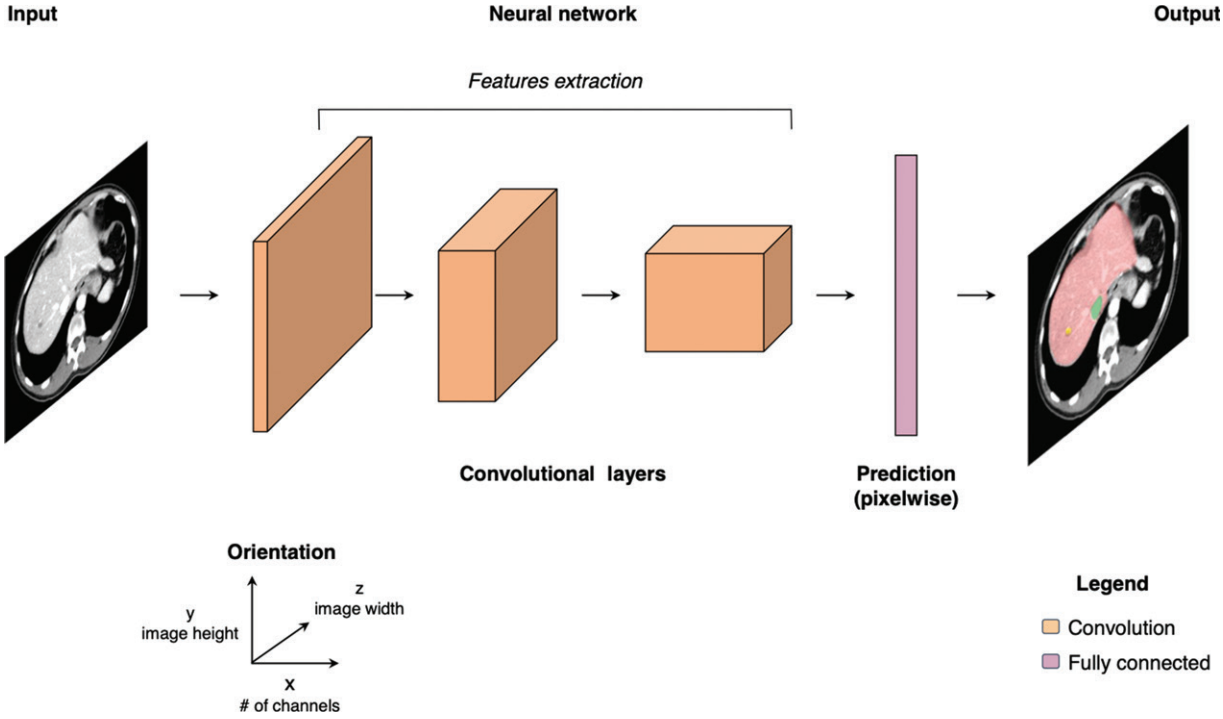


Figure 3. Fully convolutional networks typically consist of a stack of layers performing successive convolutions on an input (eg, an image), as depicted in this diagram. The first input layer on the left corresponds to the original image, with individual signal intensities for each pixel. The successive convolutional layers color coded in orange allow the extraction of features to compute intermediate representations. The changes in box size indicate the evolution of the dimension of the feature maps after successive convolutions and pooling operations. The prediction layer color coded in green predicts the class of each pixel. In this example, the presented architecture allows further segmentation and detection of lesions and organs (16).

in a CNN is the output of a convolution kernel applied to either an input image or to the set of feature maps produced by the previous neural network layer. The number of output feature maps from a layer is therefore the number of convolution kernels in the layer. Bottlenecks are implemented by a set of 1 × 1 convolution kernels, which preserve the spatial dimensions of the previous layer but can change the number of feature maps (dependent on the number of convolution kernels). Reducing the number of feature maps reduces the computational complexity of subsequent convolution operations and effectively compresses the input feature maps into a more compact representation. The number of feature maps can be subsequently augmented by a 1 × 1 convolution layer with more output channels than input channels. This architecture was used effectively in Inception modules (27), as well as in the building blocks of ResNets (23).

Multibranch Convolutions

Multibranch convolutional architectures use convolutional operations in parallel in place of a single convolution (Fig 8). Each branch, for instance, can process information at a different spatial scale; the outputs of the branches are then aggregated by concatenation or summation. Such multibranch architectures (Fig E1) are postulated to efficiently increase the representational power of the network, with prominent examples again seen in Inception modules (Fig E1) (27) and the ResNeXt architecture (23).

Wider Networks

Owing to diminishing returns in neural network performance with increasing depth, there has also been work on scaling the width of the networks, referring to the number of feature maps or channels per convolutional layer. Wide residual networks (Wide ResNets) in some cases can be trained more easily and perform better than deeper conventional ResNets (28) although at the expense of increased number of parameters and memory requirements. More recently, the EfficientNet family of models scales depth, width, and resolution of networks in a balanced manner to provide an effective trade-off between size and accuracy (29).

Ensembles of Networks

Combining the results of an ensemble of independently trained neural networks can improve performance (Fig 9). Ensembles have produced winning results in ImageNet image classification competitions (30), as well as in radiology tasks such as pediatric bone age prediction and pneumonia detection (31,32). Recent experimental

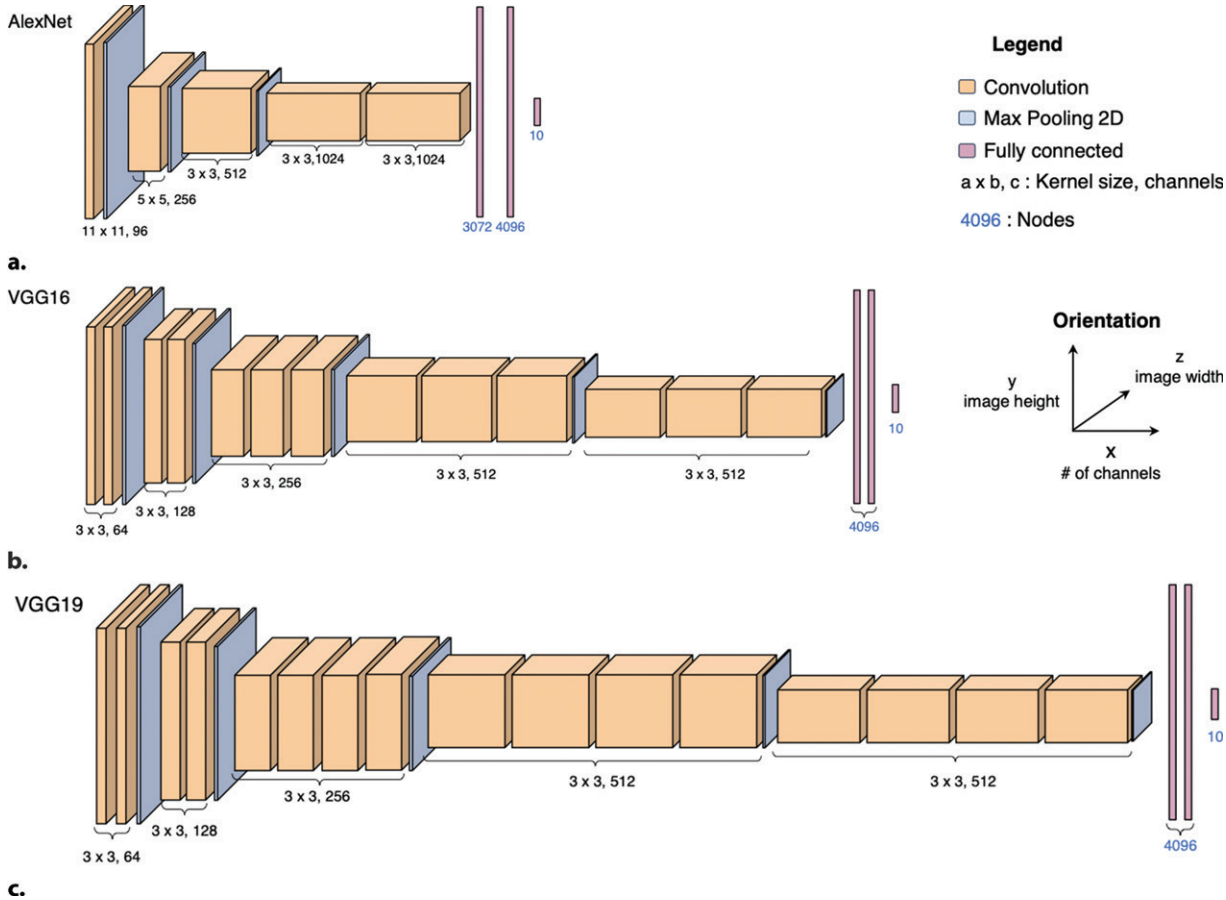


Figure 4. Evolution of deep neural networks toward deeper architectures. The increase in layers may increase the capacity of a network to learn complex features. Representative models are shown here: AlexNet **(a)** (17), VGG16 **(b)** (18), and VGG19 **(c)** (18). The numbers below the convolution layers color coded in orange indicate the two-dimensional *(2D)* kernel size and the number of channels. The maximum *(Max)* pooling operations color coded in blue consist in extracting the maximum value in a kernel to preserve information while reducing computation requirements. The changes in box size indicate the evolution of dimensions of the feature maps after successive convolutions and pooling operations. The fully connected layers color coded in pink allow reasoning about the entire image.

work with neural network ensembles suggests that independently trained networks effectively sample from different local optima in the solution space and improve accuracy through functional diversity (33).

Architectures Adapted to Tasks

Classification Architectures

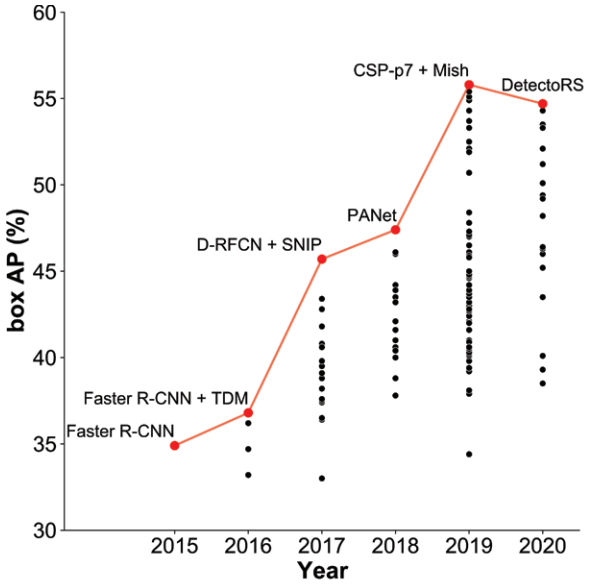
Classification networks are the simplest deep learning architectures, as their goal is simply to predict a category for an image. However, refinements of these architectures have translated into improvements in other applications as well, as the basic structures of these networks are often used as building blocks of more complex architectures (Fig 10).

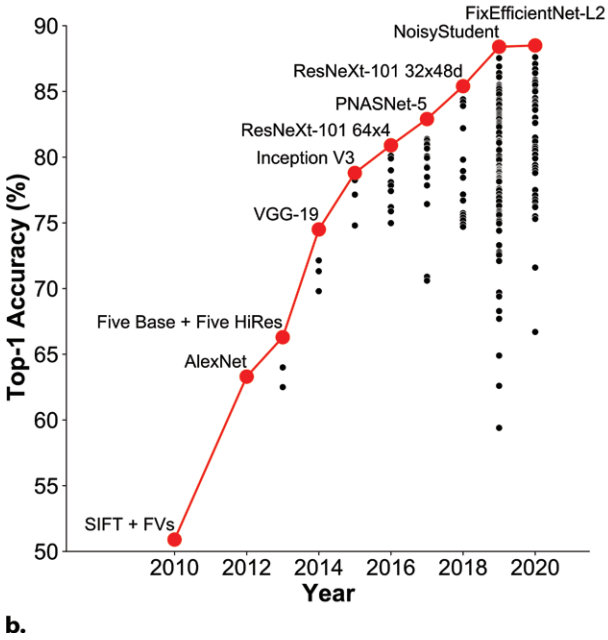
The basic CNN building blocks described previously are combined to create the architecture of a backbone encoding CNN network. This base network progressively downsamples the input image in the spatial dimensions while translating the spatial information into semantic information encoded in the channel dimension. The final layers distill the encoded semantic information into a limited number of task-specific classes.

Detection Architectures

Detection architectures build on the architectural innovations of CNNs, often incorporating the backbone of a trained classification network. However, detection architectures must not only classify objects in an image but also predict the coordinates of bounding boxes that localize the detected objects. The most common detection architectures can be organized into two categories on the basis of the number of stages in the detector (Fig 11).

Two-Stage Detection.—In two-stage detectors, the first stage is used to propose a sparse set of candidate regions for objects in the image, and the second stage classifies the proposals. Regions with CNN features (R-CNN) (37), Fast R-CNN (38), and Faster R-CNN (35) were a pioneering series of detectors that used this two-stage





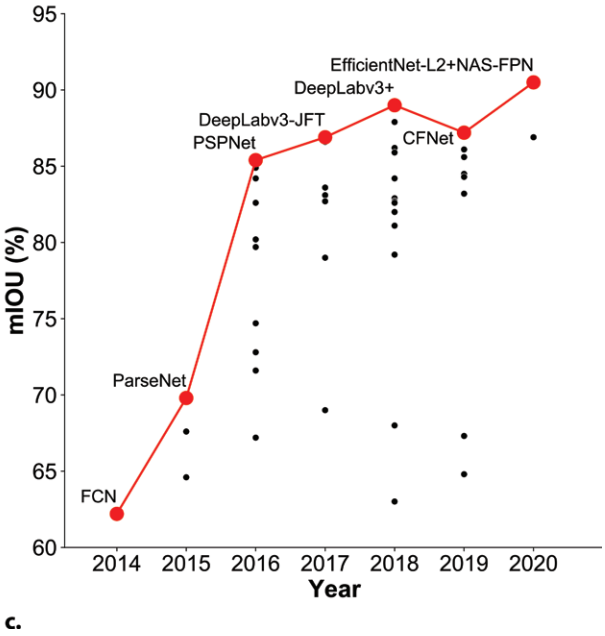


Figure 5. Graphs show the evolution of performance for three computer vision tasks (object detection, classification, and segmentation). The highest performance per model and year are shown in red. **(a)** Graph shows the average precision *(AP)* for object detection task on the Common Objects in Context (COCO) dataset (19). **(b)** Graph shows the top-1 accuracy (accuracy of predicted class with the highest probability) for classification task on the ImageNet dataset (20). **(c)** Graph shows the mean intersection over union *(mIOU)* for segmentation task on the PASCAL VOC 2012 dataset (21). The data were extracted from reference 22. The mIOU depicts the mean overlap of the predicted segmentations with ground truths. *CFNet* = cascade and fused network, *CSP* = cross stage partial network, *D-RFCN* = deformable region-based fully convolutional networks, *DetectoRS* = detecting objects with recursive feature pyramid and switchable atrous convolution, *FCN* = fully convolutional network, *FPN* = feature pyramid networks, *HiRes* = high resolution, *NAS* = neural architecture search, *PSPNet* = pyramid scene parsing network, *SNIP* = scale normalization for image pyramids, *TDM* = top-down modulation, *VGG* = Visual Geometry Group.

design. Successive architectures within the series were characterized by progressive optimizations, including sharing of computations between the first and second stages.

Single-Stage Detection.—Single-stage detectors directly provide classifications and bounding boxes in a single CNN. These networks have the advantage of high efficiency but until recently have been less accurate than two-stage approaches. A series of networks called You Only Look Once introduced the approach of predicting a fixed number of bounding boxes regularly distributed over an image and classifying the presence of objects within the boxes (36,39). The Single-Shot Detector network was one of the first to demonstrate that the pyramidal shape of the feature hierarchy of a CNN could be leveraged to predict objects at different scales (40).

A limiting factor for accuracy in early single-stage architectures was the large imbalance between true and false positives among the large number of candidate object locations. To address this problem, the RetinaNet architecture introduced a new focal loss function that helped focus training on difficult misclassified training examples (41). RetinaNets were the basis of several top-ranking solutions in the Radiological Society of North America pneumonia detection challenge (32).

Feature Pyramid Networks.—Feature pyramid networks (FPNs), proposed by Lin et al (42), are a cornerstone of modern object detection

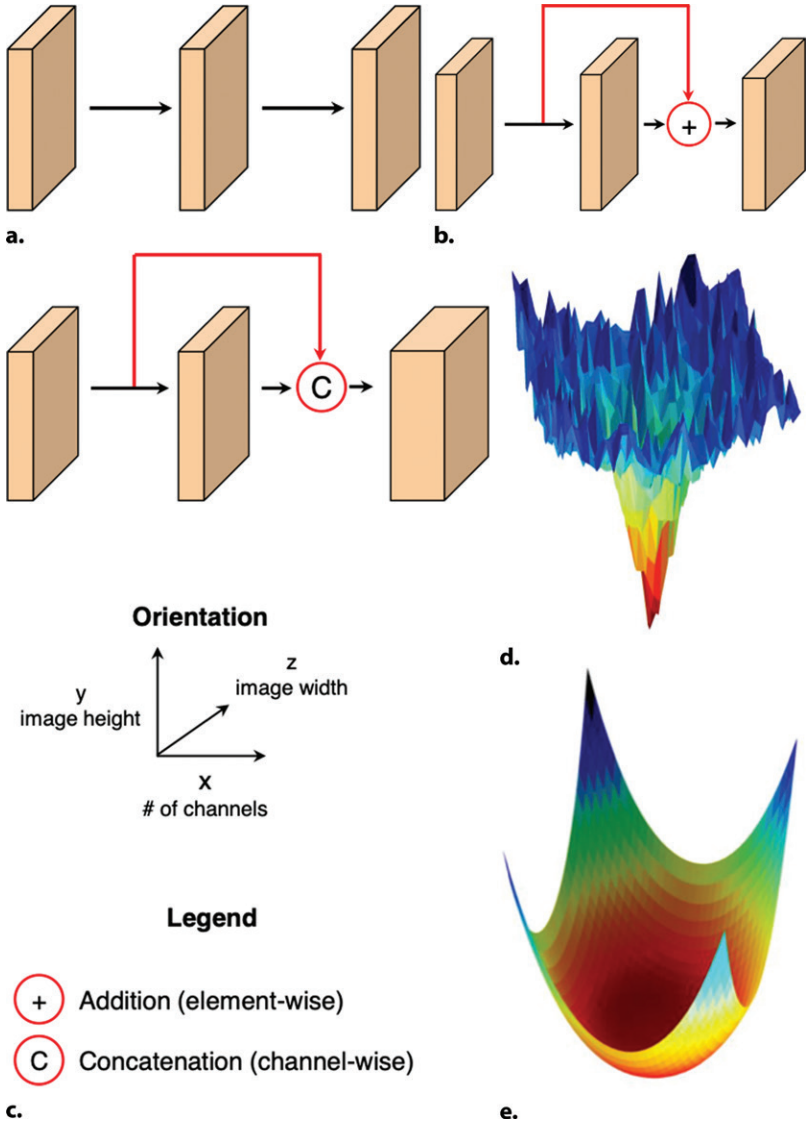
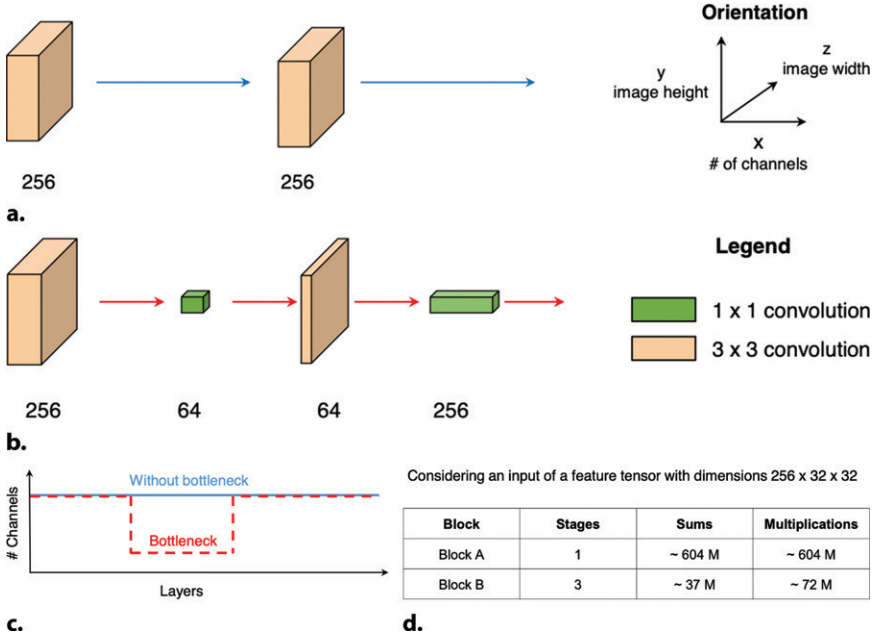


Figure 6. Skip connections are shortcut connections from one layer to a deeper layer, skipping one or more layers. **(a)** Diagram shows the standard neural network architecture, with successive connections between layers. **(b)** Diagram shows the skip connection by element-wise addition as used in the ResNet architecture, color coded in red. **(c)** Diagram shows the skip connection by channel-wise concatenation as used in DenseNet architecture, color coded in red. **(d)** Artistic rendering of the loss function in the case of direct connections, as presented in **a**. **(e)** Artistic rendering of the loss function in the case of skip connections, as shown in **b** and **c**. Skip connections tend to induce smoother loss landscapes compared with direct connections, thus facilitating convergence (ie, iteratively converging toward the minimum of the loss function) during training (25).



Block	Stages	Sums	Multiplications
Block A	1	~ 604 M	~ 604 M
Block B	3	~ 37 M	~ 72 M

Figure 7. Bottleneck blocks. **(a)** Diagram shows a convolution layer color-coded in orange that indicates a 3 × 3 kernel size with 256 channels. **(b)** Diagram shows a bottleneck block that takes advantage of convolution layers color coded in green to indicate a 1 × 1 kernel size to decrease the dimension of channels to 64 and reduce the computation burden. **(c)** Graph shows that the number of channels (color coded in blue) is preserved without a bottleneck block and reduced with bottleneck blocks (color coded in red). **(d)** Example table shows the calculations for the two scenarios in **a** and **b** (26).

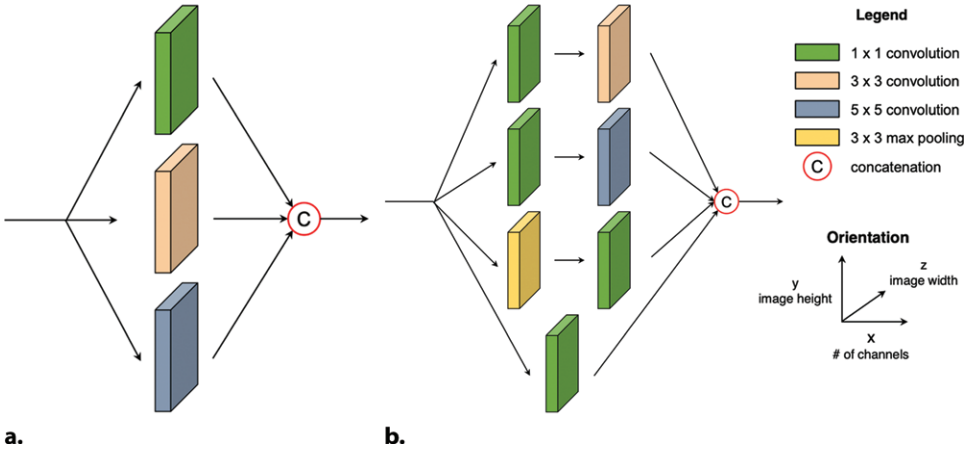


Figure 8. Multibranch convolutions diagrams. **(a)** The input is processed in parallel by several layers, and the output consists in the concatenation of these layers. **(b)** The Inception module diagram exhibits four parallel branches (27).

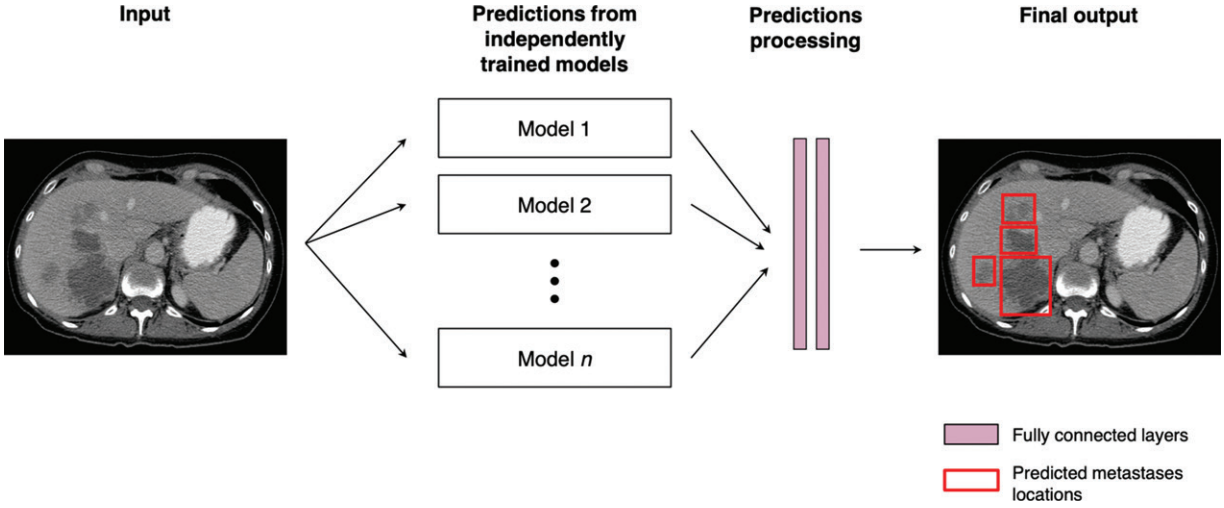


Figure 9. Ensemble networks diagram. A set of trained models, with identical or different architectures, is used to generate multiple predictions, which are then processed by fully connected layers to provide a prediction. This ensemble architecture can be used for various computer vision tasks (eg, detection, classification, and segmentation) (30).

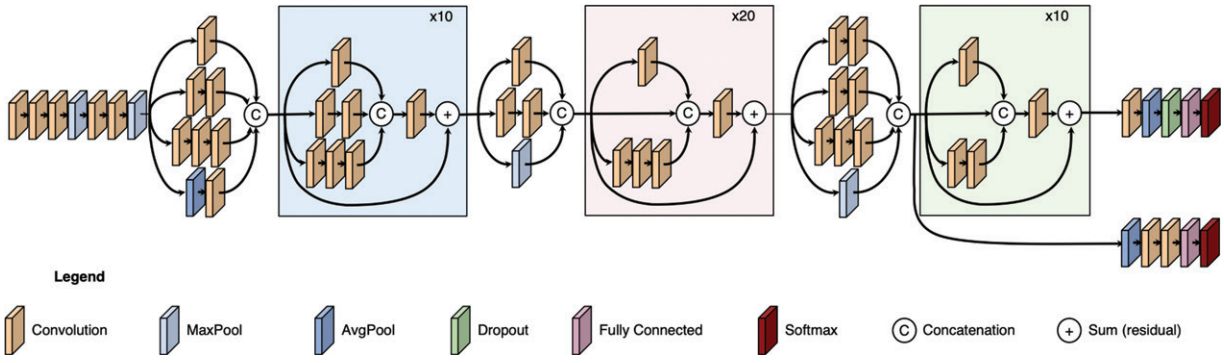


Figure 10. Architecture for classification diagram. A deep convolutional network for image classification can combine several convolution, maximum pooling, and average pooling layers. A convolution layer comprises image filters that detect features relevant to the task at hand. Maximum pooling operations *(MaxPool)* downsample by sliding a small window (eg, 2 × 2 pixels) across the image and taking the maximum value within the window. Average pooling *(AvgPool)* is analogous to maximum pooling, except the average value is used. Dropout randomly turns off neurons (ie, weights) within a layer to prevent overfitting. Fully connected layers are typically used at the end of the CNN to map the feature vector to the predicted classes. Throughout the architecture, feature maps from different branches are joined together (concatenation), and outputs from different layers are summed (residual connections). This increases the representational power of the CNN and stabilizes the training process (34).

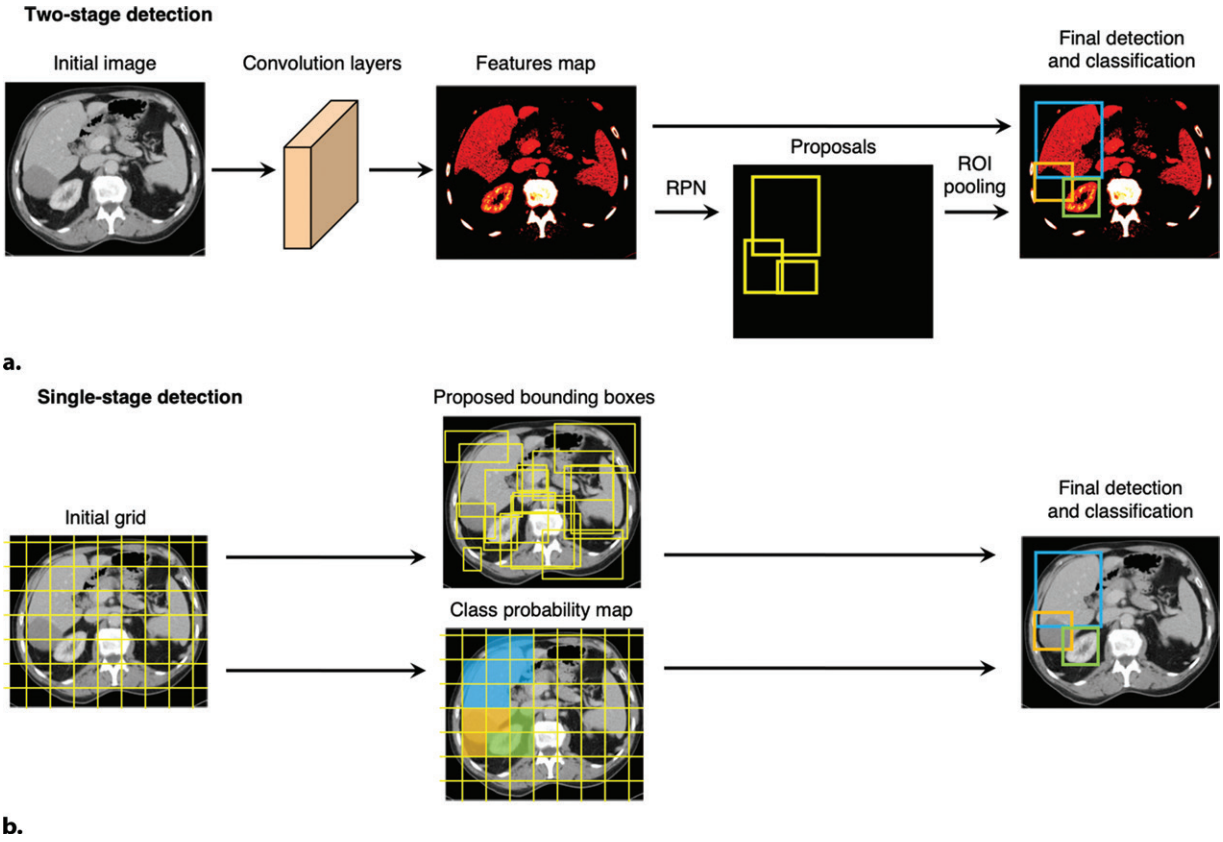


Figure 11. Architectures for object detection diagrams. **(a)** Two-stage detection involves a region proposal network *(RPN),* which generates a set of region of interest *(ROI)* proposals based on possible objectness (step 1) before further classification (step 2) (35). **(b)** Single-stage detection combines a class probability mask with regions extracted from the initial image (36).

approaches in both single-stage and two-stage detectors (Fig 12). CNN architectures progressively increase the number of feature maps throughout the depth of the network; for computational efficiency, the input image must be downsampled to accommodate more feature maps. Thus, there is a trade-off between the spatial resolution of the feature maps and the semantic richness of their contents. This is especially relevant in object detection, as the decreased spatial resolution of deeper feature maps leads to difficulty in identifying small objects. FPNs were developed as a solution to this problem and comprise two parts: a bottom-up pathway, which is simply the conventional CNN backbone, and a top-down pathway, which progressively upsamples the deeper semantically rich feature maps to a higher spatial resolution (43). Importantly, shallower feature maps are processed by a 1×1 convolutional layer and added to the output of each level of the top-down pathway to provide valuable spatial information for object detection. These more powerful feature maps are then provided as input to the final classification and bounding box heads.

Segmentation Architectures

Architectures for segmentation tasks such as semantic segmentation and instance segmentation must label every pixel in an image. Using CNNs efficiently for segmentation requires solving an upsampling problem, in which low-resolution semantically rich maps produced by convolutional and pooling layers must be converted to high-resolution segmentation masks.

Upsampling Techniques.—Two important operations that perform upsampling are unpooling and transpose convolution (Fig 13). Unpooling involves recording the locations of maxima in each pooling operation and later using these locations to convert a low-resolution feature map into a sparse higher-resolution representation. Transpose convolution is an upsampling operation using kernels with learnable weights. In contrast to conventional convolution, which sums the products of kernel elements with input pixel values, transpose convolution uses pixel values of the input as weights for copies of the kernel to be added to the higher-resolution output.

Encoder-Decoder.—The fully convolutional network (16) pioneered an encoder-decoder design for segmentation. An encoder network uses a series of downsampling convolutional layers from a classification model to output a low-resolution spatial map instead of classification scores. A

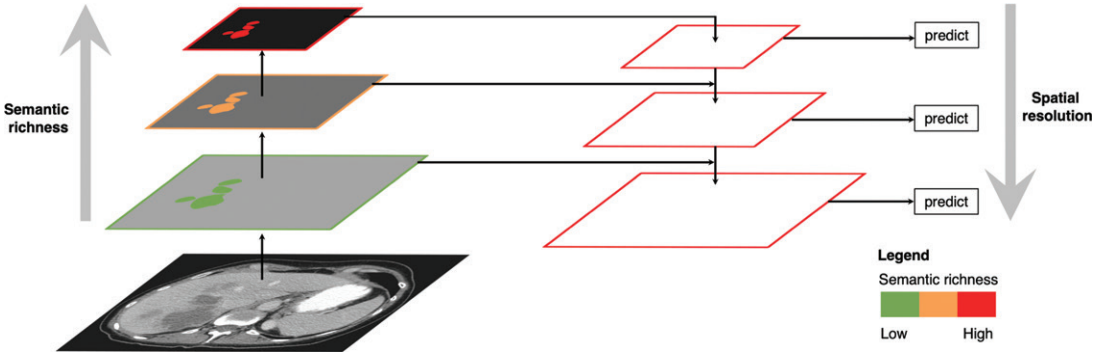
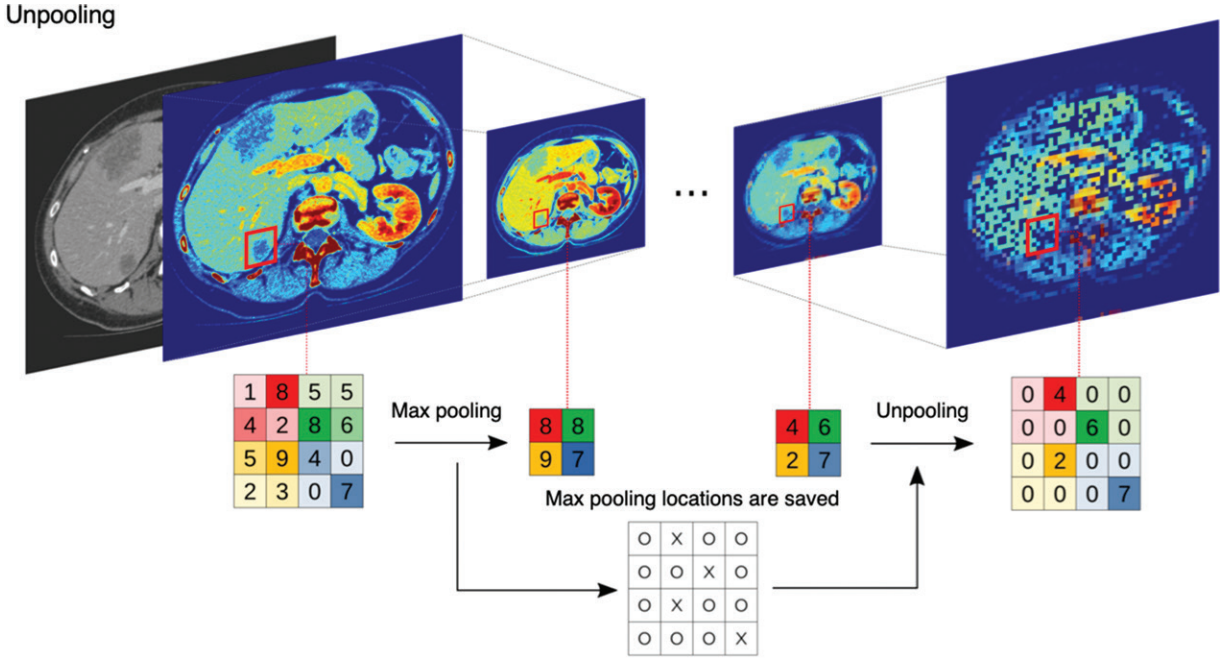
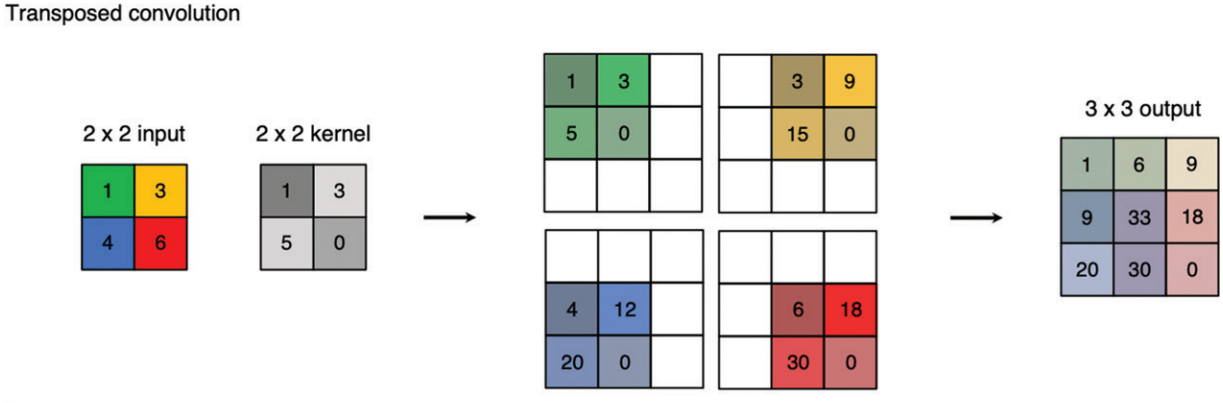


Figure 12. Feature pyramid network diagram. Bottom-up pathway (left) consists of consecutive convolutions producing a pyramidal hierarchy of feature maps at several scales. The coarsest feature map, which encodes the semantically strongest features, is then upsampled along the top-down pathway (right). Lateral connections (horizontal arrows) merge localization-rich information from bottom-up feature maps with semantic-rich information from the top-down feature maps. (Adapted and reprinted, under a CC BY 4.0 license, from reference 33.)



a.



b.

Figure 13. Diagrams of upsampling techniques. **(a)** Unpooling saves locations of maxima in pooling operations, using them later to upsample low-resolution feature maps. **(b)** An example of transposed convolution to upsample initial dimensions of a 2 × 2 input image to a 3 × 3 output image. First, each element of the input separately multiplies the kernel. Products are then summed, taking into account initial locations in the input, leading to a final 3 × 3 output (44).

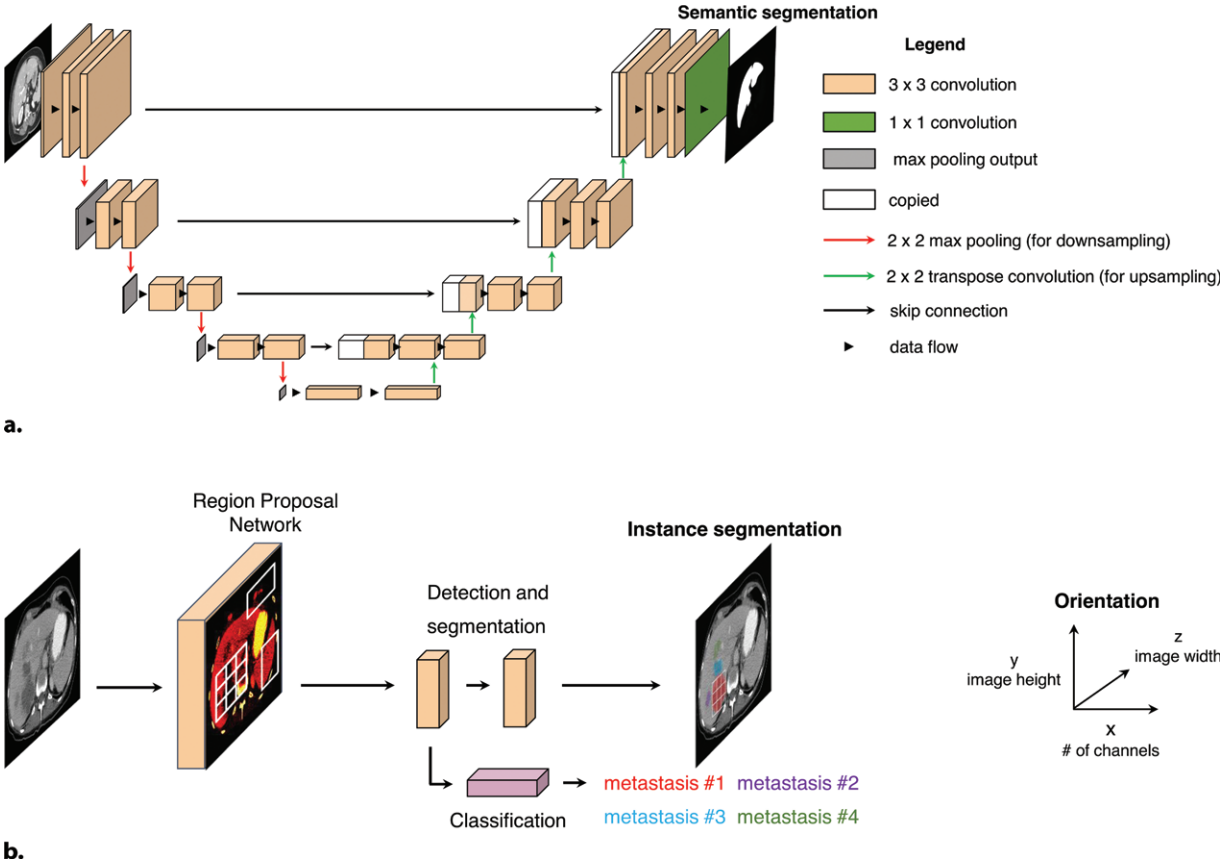


Figure 14. Diagrams show segmentation architectures: U-Net for semantic segmentation and Mask R-CNN for instance segmentation. **(a)** In the U-Net architecture, an input image first follows a succession of convolution and pooling operations to downsample the image and produce feature maps that represent an abstract representation. Then, features maps are upsampled and concatenated along the expanding path through skip connections to provide a segmentation map with the same spatial dimensions as the input image. The U-Net architecture, dedicated for segmentation, may be as deep as required by the data (46). **(b)** The Mask R-CNN architecture contains two parts: the first part includes a region proposal network applied to feature maps (red and yellow map beside convolution layer) to identify multiple regions of interest, followed by a second part to generate three outputs: bounding box for detection, classification for the category of object (eg, metastasis, hemangioma, or cyst), and instance segmentation of individual objects (47).

subsequent decoder network upsamples these maps by using transpose convolutions to produce per-pixel labeled outputs. SegNet uses both transpose convolution and unpooling in the decoder to upsample low-resolution encoder maps (45).

U-Net for Semantic Segmentation.—The U-Net is a popular architecture originally developed for segmenting microscopy images (46) but which continues to be widely used both within and outside the medical domain. U-Net has a symmetric U-shaped architecture in which a descending encoder portion downsamples the image and produces increasingly abstract representations and a subsequent ascending decoder portion uses transpose convolutions to upsample these representations to the original dimensions of the image (Fig 14). A key component of the U-Net design is the use of horizontal skip connections that facilitate the upsampling process by copying features from encoder stages directly to resolution-matched decoder stages.

An alternative method for evaluating features at several scales is to use dilated (also known as atrous) convolutions (Fig 15) (49). Dilated convolutions are convolution operations with expanded kernels that contain spaces between adjacent kernel elements. These kernels allow modeling of larger scale dependencies among pixels without losing resolution. Dilated convolutions are a central component of the DeepLab family of segmentation architectures (50).

Mask R-CNN for Instance Segmentation.—Compared with semantic segmentation, there has been less work on instance segmentation owing to fewer use cases and increased complexity of the instance segmentation problem. Instance segmentation can be considered as a problem of simultaneous object detection and semantic segmentation. The Mask R-CNN architecture is a prototypical two-stage network for instance segmentation that extends previous work on two-stage detection models (47). As in the detection

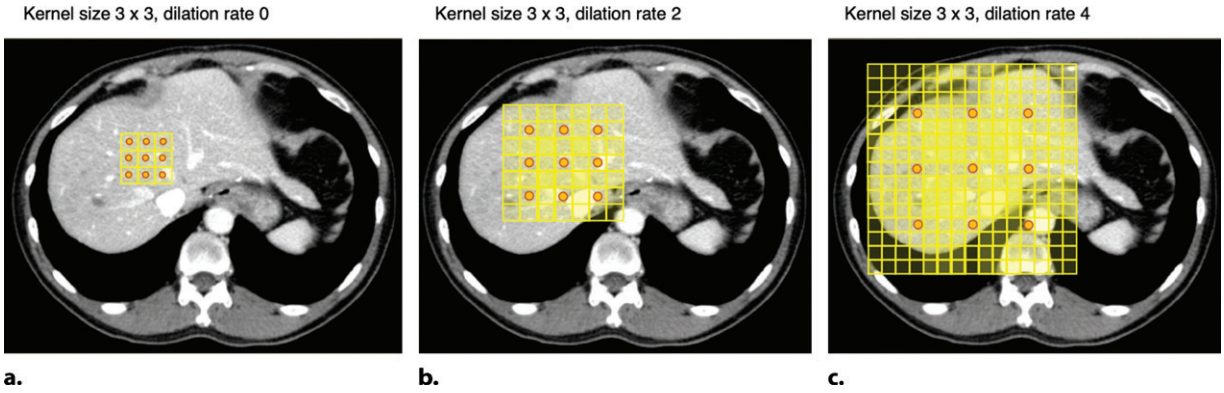


Figure 15. Dilated convolutions as depicted on axial contrast-enhanced CT images. **(a)** One-dilated convolution is equivalent to a standard convolution; that is the receptive field (yellow tiles) has the same size as the kernel (red dots). **(b, c)** Two-dilated **(b)** and four-dilated **(c)** convolutions increase the receptive field and thus introduce more contextual information in the process while maintaining a constant output shape (48).

models, the first stage proposes candidate regions of interest, while the second stage predicts bounding boxes and object classifications. Mask R-CNN adds a branch to the second stage that predicts a binary mask for the region of interest for each object category by using a convolutional architecture based on a fully convolutional network. The detection and mask components of the model are trained jointly to produce instance segmentation masks (Fig 14).

Generative Architectures

GANs (8) have rapidly evolved with a broad range of computer vision applications. Typically, a GAN consists of two distinct networks: a generator, which aims to learn to create "fake" images that appear to fit into the distribution of training samples, and a discriminator, dedicated to distinguishing samples from the training set (real) or from the generator (fake). In the original vanilla GAN (Fig 16), random noise is transformed by the generator and submitted to the discriminator. Training alternates between the discriminator and generator, with both networks improving to a point where ideally the generator produces realistic images. Conditional GANs, or cGANs (51), use extra input information (eg, labels or data from other modalities) to force structure on how the generator produces images. CycleGANs (52) use images as input to translate them from one domain to another, such as T2-weighted MR images into T1-weighted MR images (53).

In medical imaging, GANs have been applied to classic tasks such as detection, classification (54), and segmentation (55), as well as to image reconstruction (56), synthesis (57), and registration (58). GANs have also been used for data augmentation to increase the training dataset size (8).

Training and Validation

Training

Training a neural network involves fitting the model weights to a training dataset to achieve good performance on a given task, such as classification or detection. Several factors affect the training performance, speed of convergence (finding a solution), and whether the model will perform well on new data.

Model Selection.—Given the wide variety of neural network designs, selecting a suitable architecture for a task may be an iterative process. Design choices are often based on one's intuition about the task. Experimenting with different loss functions and probing intermediate results within the network can also provide useful information about how to use specific layers or network configurations. Model exploration may start with a simple architecture known to work on a similar task. Once convergence is observed on the first training iterations, the capacity of the network, represented by the number of trainable parameters, is increased until overfitting is observed on a validation dataset.

Hyperparameters.—While millions of parameters are automatically optimized during the training phase, some of them called hyperparameters are set manually, such as the number of layers and the learning rate. Using systematic grid search or random search for hyperparameters may be a good strategy to find suitable values (59).

Among the hyperparameters for training, the learning rate most directly affects the training convergence speed by specifying how much the model weights are updated with respect to the loss gradient at each training step. A small learning rate can result in slow training and possibly overfitting (60), while a large learning rate causes the training

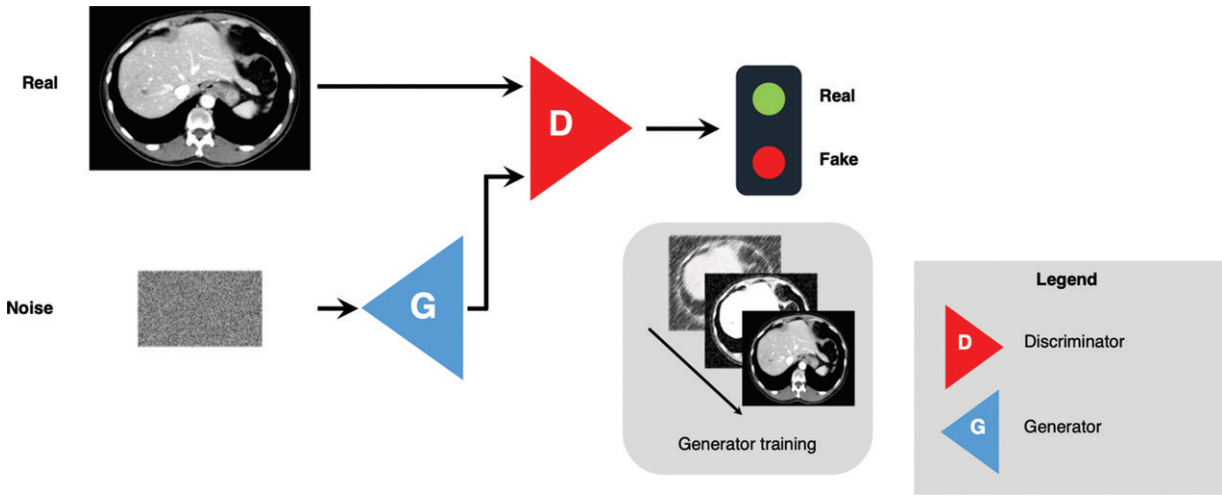


Figure 16. Standard GAN architecture diagram. A first network, known as the generator *(G)*, aims to transform a random input into a realistic image to fool a second network, known as the discriminator *(D)*. During training, the generator learns from the response of the discriminator (8).

process to diverge. A cyclical learning rate which rises and falls during training may reduce training time and improve accuracy (61), likely by overcoming local minima in the loss landscape.

Regularization.—Regularization refers to strategies designed to prevent overfitting during training, sometimes at the expense of increased training error. Regularization can take many forms. Early stopping involves monitoring validation error during training and stopping training when validation error starts increasing (2). Weight decay penalizes extreme values of network weights, considered a symptom of overfitting. Dropout directly affects the capacity of the model by randomly removing a subset of neurons from the network at each training step, forcing the network to employ different computation pathways to reach the same output, and making the network as a whole more robust (62).

Data Sampling.—Data sampling can also affect convergence speed and performance. A common challenge is imbalance of classes in a dataset, for instance when healthy cases significantly outnumber diseased cases. Randomly sampling such a dataset is likely to push the model toward classifying most cases as healthy, reaching high specificity but poor sensitivity for disease. Common solutions to mitigate this imbalance include strongly weighting the minority class in the loss function (63) or oversampling the minority class to expose the model equally to all classes (64). Focusing on difficult recurrently misclassified cases may also improve training efficiency (65). In addition, an example of normalization, the process of shifting and scaling variables so that they have comparable statistical distributions, can be found in Figure E2 (66).

Transfer Learning.—Owing to barriers in sharing medical image data, sufficient labeled training images are often not readily available for a given task. Transfer learning is a process by which models pretrained on larger generic image datasets such as ImageNet (20) can be fine-tuned for tasks on smaller datasets. Transfer learning can mitigate data requirements for model convergence and has become routinely used in medical imaging research.

Validation

The performance of a deep learning model on training data does not predict its ability to generalize to unseen data. A standard strategy to improve and predict the generalizability of a model is to split a dataset randomly into three subsets: training, validation, and test sets. The split should be disjoint, such that the same patient is not represented in more than one set. It may also be helpful to balance the sets by stratifying the split according to variables such as sex, age, or label prevalence.

The model's weights are optimized by using the training set data. The validation set is used to tune model hyperparameters, with periodic evaluation of model performance on validation set data guiding the training process. Progressively worsening performance of the model on the validation set data is a sign of overfitting.

The test dataset, unseen by the model during training, is used to assess a fully trained model's ability to generalize to new data. Ideally the test dataset is evaluated only once. Multiple evaluations of the test dataset among several training cycles may lead to overfitting on the test set, invalidating its utility for predicting real-world performance.

Real-world performance of a model can be further assessed by separate datasets that are completely external from the original data col-

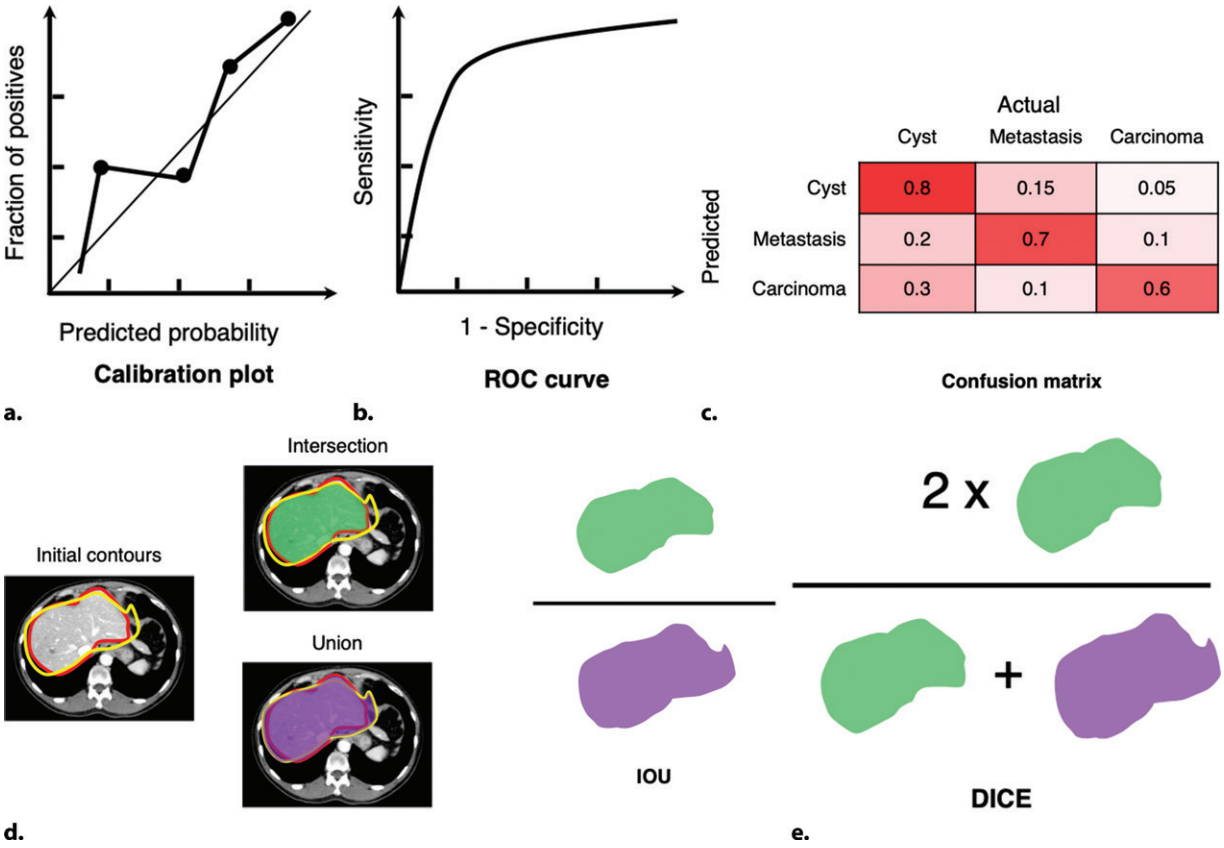


Figure 17. Common metrics used to assess model performances on various tasks: calibration plot for prediction probabilities **(a)**, receiver operating characteristic curve *(ROC)* for classification of binary classes or dichotomized ordinal classes **(b)**, confusion matrix for classification of multiple classes of objects **(c)**, intersection over union *(IOU)* for segmentation and detection **(d)**, and Dice score **(e)**. Both IOU and Dice score quantify the degree of spatial overlap between ground truth and predicted masks.

lection. Such an assessment may include temporal validation with newly recruited patients, or geographic validation with data from a different site (67). Geographic validation may be especially helpful to evaluate how a model works on data acquired with different equipment or technical parameters or with different patient populations.

Once a model is deployed, performance should be monitored to detect any bias or loss of accuracy. Modes of continuous learning have been proposed to keep models current with changing data and equipment configurations (68). For instance, feedback from radiologist users who may accept or reject the findings of the system could theoretically be used as new training data to improve performance.

Performance Metrics

Metrics are the quantitative assessment of model performance during training, validation, and monitoring steps. Appropriate selection of metrics is dependent on the task (Fig 17).

For simple binary classification (eg, assessing whether some disease process is present), the common biostatistical metrics of sensitivity, specificity, and positive and negative predictive probability can be applied. As deep learning models typically output a probability that an image belongs to a particular class, a decision threshold probability can be adjusted to trade off sensitivity and specificity in the model. Calibration plots assess the accuracy of a model's output probability by plotting the fraction of true positives in a test set as a function of the output probability. A diagonal line on the calibration plot represents perfect calibration, meaning that the output probability of the model accurately measures the model's uncertainty.

A receiver operating characteristic (ROC) curve depicts the trade-off of sensitivity and specificity by plotting the true-positive rate (sensitivity) versus the false-positive rate (1-specificity) as the decision threshold is varied. The area under the ROC curve (AUC) provides a measure of model performance across all decision thresholds, with a perfect model having an AUC of 1 and a random model having an AUC of 0.5. However, from a practical standpoint it is useful to report sensitivity and specificity at a decision threshold optimized for the intended use case of the model.

For multiclass classifications (eg, for multiple lesion types), statistics for each class can be

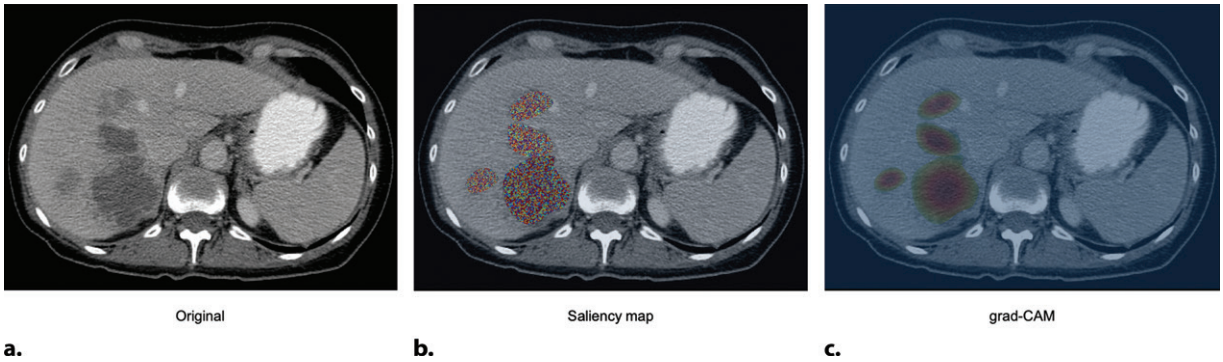


Figure 18. Common visualization techniques for a classification network as depicted on axial contrast-enhanced CT images. **(a)** Features are learned by the CNN from training samples. **(b)** The saliency map technique computes the gradient of the output class with respect to the input image, highlighting which pixels were involved in the final classification result (71). **(c)** Gradient-weighted class activation mapping *(grad-CAM)* uses class probability and backpropagation from the last convolution layer to provide an attention map (72).

reported. These statistics can be averaged across classes, optionally weighting each class according to its prevalence in the test set. Confusion matrices are contingency tables that tabulate the predicted classes for the instances of each actual class in a test set and are useful to help evaluate whether particular classes tend to be confused.

Object detection and segmentation require metrics that describe how well a predicted area matches the ground truth area, which is typically delineated by a radiologist. The intersection is the overlap between the predicted area and the ground truth area, while the union is the total area encompassed by the prediction and ground truth. Intersection over union and Dice score (69) combine these measures in slightly different ways, but both equal 1 in the case of a perfect match between prediction and ground truth.

For object detection, a particular detection can be considered correct by comparing intersection over union with a cutoff value (eg, 0.5). Precision is the number of true-positive detections as a fraction of all detections and measures the model's positive predictive value. Recall is the number of true-positive detections as a fraction of all ground truth objects and measures the model's sensitivity. Thresholding the confidence values assigned to bounding boxes by an object detection model is used to trade-off precision and recall, resulting in a precision-recall curve (analogous to an ROC curve). Average precision (AP) summarizes model precision across the entire range of recall values and is calculated by a modified area under the precision-recall curve, analogous to the AUC. The mean AP is the average of the AP calculated for all the detected types of objects.

Visualization

While deep neural networks can perform state-of-the-art image classification, clinical users typically want to visualize the areas in an image that explain a particular classification. Such visualization can increase a user's confidence in the system or help reveal confounding factors that influence the system's classification, such as external markers on a chest radiograph (70).

The most common methods for visualization calculate gradients on the basis of a forward and backward pass through the network for an image (Fig 18). Saliency maps (71) compute gradients of the class score with respect to image pixels; these gradients indicate which pixels need to be changed the least to affect the class score the most. Class activation maps (CAMs) are exemplified by gradient-weighted CAM (Grad-CAM) (72), which computes gradients of the class score with respect to channels in the last convolutional layer in the model rather than to the input image. These gradient values are used to produce a weighted sum of the channels in this layer, resulting in a heat-map of important features in the input image. Although this map is coarser than a saliency map, the features tend to be more specific to the predicted class.

Visualization techniques are less relevant in object detection and segmentation tasks, where the model output already provides relevant localization information.

Future Directions

A proposed solution to privacy concerns regarding multisite sharing of clinical image data is federated learning (6). Federated learning allows data to stay with the originating hospital, with neural network training instead distributed among the different institutions. However, there remain formidable obstacles to such a strategy, including accounting for the heterogeneity of patient populations across institutions without centralized access to all the data.

Most deep learning models in radiology process two-dimensional (2D) images even when the image datasets are three-dimensional (3D). Increased availability of medical 3D image datasets will likely result in evolution and optimization of 3D CNN architectures. Evolving alternatives to 3D CNNs include combinations of 2D CNNs with neural networks specialized for sequence data to process sequential 2D images of a 3D volume.

As deep learning models transition into clinical applications, we must consider the ethical ramifications of their use (73). Bias in machine learning remains under-researched. For instance, deep learning researchers usually report aggregate metrics over the entire dataset without consideration of subgroups, especially underserved populations that are underrepresented in the training data.

Since deep learning models are likely to serve as adjunctive tools for radiologists, work on model interpretability is crucial to adoption and usage of these typically black-box models. For example, in contrast to a CNN that learns deterministic weights, a Bayesian neural network learns parameters of random variables used to sample weights. These parameters can be used to express the uncertainty of the model's predictions and thereby help radiologists understand a model's limitations.

Good model performance does not guarantee improved patient outcomes. The value of a model is dependent on its impact on clinical decisions and the nature and prevalence of the clinical problem. As a result, analogous to other technology assessments, controlled studies measuring practical clinical endpoints are necessary for understanding the clinical value of deep learning.

Conclusion

Deep learning is an artificial intelligence technique that has been successful in computer vision. Familiarity with the key concepts described in this article will help radiologists stay informed on the advances in deep learning and facilitate clinical adoption of these techniques.

***Disclosures of Conflicts of Interest.*—I.P.** *Activities related to the present article:* disclosed no relevant relationships. *Activities not related to the present article:* consultant for MD.ai. *Other activities:* disclosed no relevant relationships. **A.T.** *Activities related to the present article:* disclosed no relevant relationships. *Activities not related to the present article:* research scholarships from Fonds de recherche du Québec en Santé (FRQ-S) and Fondation de l'association des radiologistes du Québec; active grants from Institut devalorisation d données (IVADO), Onco-Tech Project Grant (consortium composed of Oncopole, Medteq, Institut TransMedTech, Société de recherche sur le cancer), and the Canadian Institutes of Health Research (CIHR #389385); and speakers honoraria from Siemens Healthineers and Eli Lilly. *Other activities:* disclosed no relevant relationships.

References

1. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521(7553):436–444.
2. Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge, Mass: MIT Press, 2016.
3. Litjens G, Kooi T, Bejnordi BE, et al. A survey on deep learning in medical image analysis. Med Image Anal 2017;42:60–88.
4. Chartrand G, Cheng PM, Vorontsov E, et al. Deep Learning: A Primer for Radiologists. RadioGraphics 2017;37(7):2113–2131.
5. Montagnon E, Cerny M, Cadrin-Chênevert A, et al. Deep learning workflow in radiology: a primer. Insights Imaging 2020;11(1):22.
6. Willemink MJ, Koszek WA, Hardell C, et al. Preparing Medical Imaging Data for Machine Learning. Radiology 2020;295(1):4–15.
7. Salman S, Liu X. Overfitting Mechanism and Avoidance in Deep Neural Networks. CoRR. 2019;abs/1901.06566. http://arxiv.org/abs/1901.06566. Published January 19, 2019. Accessed May 1, 2021.
8. Sandfort V, Yan K, Pickhardt PJ, Summers RM. Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks. Sci Rep 2019;9(1):16884.
9. Goodfellow IJ, Pouget-Abadie J, Mirza M, et al. Generative Adversarial Nets. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ, eds. Advances in Neural Information Processing Systems 27: NIPS 2014, Montreal, Canada, December 8–13, 2014; 2672–2680.
10. Heim E, Roß T, Seitel A, et al. Large-scale medical image annotation with crowd-powered algorithms. J Med Imaging (Bellingham) 2018;5(3):034002.
11. Mehta P, Sandfort V, Gheysens D, Braeckevelt GJ, Berte J, Summers RM. Segmenting The Kidney On CT Scans Via Crowdsourcing. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), Venice, Italy, April 8–11, 2019. Piscataway, NJ: IEEE, 2019; 829–832.
12. Zech J, Pain M, Titano J, et al. Natural Language-based Machine Learning Models for the Annotation of Clinical Radiology Reports. Radiology 2018;287(2):570–580.
13. Lee DH. Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. ICML Workshop on Challenges in Representation Learning, Atlanta, GA, 2013.
14. DSI Use Cases. American College of Radiology Data Science Institute. https://www.acrdsi.org/. Accessed July 2, 2020.
15. The Cancer Imaging Archive (TCIA). https://www.cancerimagingarchive.net/. Accessed July 2, 2020.
16. Long J, Shelhamer E, Darrell T. Fully Convolutional Networks for Semantic Segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Piscataway, NJ: IEEE, 2015; 3431–3440.
17. Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira F, Burges C, Bottou L, Weinberger K, editors. Advances in Neural Information Processing Systems 25. Curran Associates; 2012:1097–1105.
18. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. Presented at the International Conference on Learning Representations, San Diego, CA, May 7–9, 2015.
19. Lin TY, Maire M, Belongie S, et al. Microsoft COCO: Common Objects in Context. arXiv:14050312 [prepint] http://arxiv.org/abs/1405.0312. Posted May 1, 2014. Accessed March 3, 2021.
20. Deng J, Dong W, Socher R, Li LJ, Li K, Li F. ImageNet: A Large-Scale Hierarchical Image Database. 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, June 20–25, 2009. Piscataway, NJ: IEEE, 2009; 248–255.
21. Everingham M, Gool L, Williams CK, Winn J, Zisserman A. The Pascal Visual Object Classes (VOC) Challenge. Int J Comput Vis 2010;88(2):303–338.
22. Papers with Code. Browse the State-of-the-Art in Machine Learning. https://paperswithcode.com/sota/. Accessed October 21, 2020.

23. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, June 27–30, 2016; 770–778.
24. Sun S, Chen W, Wang L, Liu X, Liu TY. On the Depth of Deep Neural Networks: A Theoretical View. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, February 12–17, 2016: AAAI Press, 2016; 2066–2072.
25. Li H, Xu Z, Taylor G, Studer C, Goldstein T. Visualizing the Loss Landscape of Neural Nets. In: Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R, eds. Advances in Neural Information Processing Systems 31. Red Hook, NY: Curran Associates, 2018; 6389–6399.
26. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetV2: Inverted Residuals and Linear Bottlenecks. ArXiv:180104381 [preprint] http://arxiv.org/abs/1801.04381. Posted January 13, 2018. Accessed October 21, 2020.
27. Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2015; 1–9.
28. Zagoruyko S, Komodakis N. Wide Residual Networks. ArXiv: 1605.07146 [preprint] http://arxiv.org/abs/1605.07146. Posted May 23, 2016. Accessed May 1, 2021.
29. Tan M, Le QV. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: Chaudhuri K, Salakhutdinov R, eds. Proceedings of the 36th international conference on machine learning. Cambridge, MA: PMLR, 2019;97:6105–6114.
30. Ju C, Bibaut A, van der Laan M. The relative performance of ensemble methods with deep convolutional neural networks for image classification. J Appl Stat 2018;45(15):2800–2818.
31. Halabi SS, Prevedello LM, Kalpathy-Cramer J, et al. The RSNA Pediatric Bone Age Machine Learning Challenge. Radiology 2019;290(2):498–503.
32. Pan I, Cadrin-Chênevert A, Cheng PM. Tackling the Radiological Society of North America Pneumonia Detection Challenge. AJR Am J Roentgenol 2019;213(3):568–574.
33. Fort S, Hu H, Lakshminarayanan B. Deep Ensembles: A Loss Landscape Perspective. ArXiv:191202757 [preprint] http://arxiv.org/abs/1912.02757. Posted June 25, 2020. Accessed August 5, 2020.
34. Szegedy C, Ioffe S, Vanhoucke V, Alemi A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. ArXiv:160207261. [preprint] http://arxiv.org/abs/1602.07261. Posted February 26, 2016. Accessed May 18, 2018.
35. Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R, eds. Advances in Neural Information Processing Systems 28. Red Hook, NY: Curran Associates, 2015; 91–99.
36. Redmon J, Divvala S, Girshick R, Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ: IEEE, 2016; 779–788.
37. Girshick R, Donahue J, Darrell T, Malik J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014; 580–587.
38. Girshick R. Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, 2015; 1440–1448.
39. Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017; 6517–6525.
40. Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector. In: Leibe B, Matas J, Sebe N, Welling M, eds. Computer Vision: ECCV 2016—Lecture Notes in Computer Science, vol 9905. Cham, Switzerland: Springer, 2016; 21–37.
41. Lin TY, Goyal P, Girshick R, He K, Dollar P. Focal Loss for Dense Object Detection. In: Proceedings of the IEEE International Conference on Computer Vision. Piscataway, NJ: IEEE, 2017; 2980–2988.
42. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature Pyramid Networks for Object Detection. ArXiv:161203144 [preprint] http://arxiv.org/abs/1612.03144. Posted December 9, 2016. Accessed September 14, 2020.
43. Hui J. Understanding Feature Pyramid Networks for object detection (FPN). Medium. https://medium.com/@jonathan_hui/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c. Published 2020. Accessed September 14, 2020.
44. Mishra D. Transposed Convolution Demystified. Towards Data Science. https://towardsdatascience.com/transposed-convolution-demystified-84ca81b4baba. Published March 10, 2020. Accessed October 21, 2020.
45. Badrinarayanan V, Kendall A, Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Trans Pattern Anal Mach Intell 2017;39(12):2481–2495.
46. Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF, eds. Medical Image Computing and Computer-Assisted Intervention: MICCAI 2015. Cham, Switzerland: Springer, 2015; 234–241.
47. He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, October 22–29, 2017. Piscataway, NJ: IEEE, 2017; 2980–2988.
48. Dumoulin V, Visin F. A guide to convolution arithmetic for deep learning. ArXiv:160307285 [preprint] http://arxiv.org/abs/1603.07285. Posted March 23, 2016. Accessed October 21, 2020.
49. Yu F, Koltun V. Multi-Scale Context Aggregation by Dilated Convolutions. Presented at the 4th International Conference on Learning Representations, ICLR 2016, San Juan, PR, May 2–4, 2016.
50. Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Trans Pattern Anal Mach Intell 2018;40(4):834–848.
51. Mirza M, Osindero S. Conditional Generative Adversarial Nets. ArXiv:14111784. [preprint] http://arxiv.org/abs/1411.1784. Published November 6, 2016. Accessed August 11, 2020.
52. Zhu JY, Park T, Isola P, Efros AA. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, October 22–29, 2017. Piscataway, NJ: IEEE, 2017; 2242–2251.
53. Welander P, Karlsson S, Eklund A. Generative Adversarial Networks for Image-to-Image Translation on Multi-Contrast MR Images: A Comparison of CycleGAN and UNIT. ArXiv 1806.0777 [preprint] http://arxiv.org/abs/1806.07777. Posted June 20, 2018. Accessed May 1, 2021.
54. Yi X, Walia E, Babyn PS. Unsupervised and semi-supervised learning with Categorical Generative Adversarial Networks assisted by Wasserstein distance for dermoscopy image Classification. ArXiv 1804.03700 [preprint] http://arxiv.org/abs/1804.03700. Posted April 10, 2018. Accessed May 1, 2021.
55. Rezaei M, Yang H, Harmuth K, Meinel C. Conditional Generative Adversarial Refinement Networks for Unbalanced Medical Image Semantic Segmentation. 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, January 7–11, 2019. Piscataway, NJ: IEEE, 2019; 1836–1845.
56. Li Z, Zhang T, Wan P, Zhang D. SEGAN: Structure-Enhanced Generative Adversarial Network for Compressed Sensing MRI Reconstruction. Proceedings of the AAAI Conference on Artificial Intelligence 33(1):1012–1019.
57. Jin D, Xu Z, Tang Y, Harrison AP, Mollura DJ. CT-Realistic Lung Nodule Simulation from 3D Conditional Generative Adversarial Networks for Robust Lung Segmentation. In: Frangi A, Schnabel J, Davatzikos C, Alberola-López C, Fichtinger G, eds. Medical Image Computing and Computer Assisted Intervention: MICCAI 2018. Cham, Switzerland: Springer, 2018; 732–740.

58. Yan P, Xu S, Rastinehad AR, Wood BJ. Adversarial Image Registration with Application for MR and TRUS Image Fusion. In: Shi Y, Suk HI, Liu M, eds. Machine Learning in Medical Imaging: MLMI 2018. Cham, Switzerland: Springer, 2018;197–204
59. Bergstra J, Bengio Y. Random Search for Hyper-Parameter Optimization. J Mach Learn Res 2012;13(10):281–305.
60. Smith LN. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. ArXiv:180309820 [preprint] http://arxiv.org/abs/1803.09820. Posted March 26, 2018. Accessed August 8, 2020.
61. Smith LN. Cyclical Learning Rates for Training Neural Networks. 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, March 24–31, 2017. Piscataway, NJ: IEEE, 2017; 464–472.
62. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. J Mach Learn Res 2014;15(56):1929–1958.
63. Huang W, Song G, Li M, Hu W, Xie K. Adaptive Weight Optimization for Classification of Imbalanced Data. In: Sun C, Fang F, Zhou ZH, Yang W, Liu ZY, eds. Intelligence Science and Big Data Engineering: IScIDE 2013. Berlin, Germany: Springer, 2013; 546–553.
64. Buda M, Maki A, Mazurowski MA. A systematic study of the class imbalance problem in convolutional neural networks. Neural Netw 2018;106:249–259.
65. Shrivastava A, Gupta A, Girshick R. Training Region-based Object Detectors with Online Hard Example Mining. ArXiv:160403540. [preprint] http://arxiv.org/abs/1604.03540. Posted April 12, 2016. Accessed September 17, 2020.
66. Wu Y, He K. Group Normalization. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y, eds. Computer Vision: ECCV 2018—15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part XIII. Springer; 2018:3–19.
67. Park SH, Han K. Methodologic Guide for Evaluating Clinical Performance and Effect of Artificial Intelligence Technology for Medical Diagnosis and Prediction. Radiology 2018;286(3):800–809.
68. Pianykh OS, Langs G, Dewey M, et al. Continuous Learning AI in Radiology: Implementation Principles and Early Applications. Radiology 2020;297(1):6–14.
69. Bertels J, Eelbode T, Berman M, et al. Optimizing the Dice Score and Jaccard Index for Medical Image Segmentation: Theory and Practice. In: Shen D, Liu T, Peters TM, et al, eds. Medical Image Computing and Computer Assisted Intervention: MICCAI 2019. Cham, Switzerland: Springer, 2019; 92–100.
70. Zech JR, Badgeley MA, Liu M, Costa AB, Titano JJ, Oermann EK. Confounding variables can degrade generalization performance of radiological deep learning models. CoRR. 2018;abs/1807.00431 [preprint] http://arxiv.org/abs/1807.00431. Posted July 2, 2018. Accessed May 1, 2021.
71. Simonyan K, Vedaldi A, Zisserman A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. ArXiv:13126034. [preprint] http://arxiv.org/abs/1312.6034. Posted December 20, 2013. Accessed July 23, 2017.
72. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, October 22–29, 2017. Piscataway, NJ: IEEE, 2017; 618–626.
73. Geis JR, Brady AP, Wu CC, et al. Ethics of Artificial Intelligence in Radiology: Summary of the Joint European and North American Multisociety Statement. J Am Coll Radiol 2019;16(11):1516–1521.