Journal of *Imaging*

MDPI

Review

A Survey on Deep Learning in COVID-19 Diagnosis

Xue Han [1,2], Zuojin Hu [1], Shuihua Wang [2] and Yudong Zhang [2,*]

[1] School of Mathematics and Information Science, Nanjing Normal University of Special Education, Nanjing 210038, China
[2] School of Computing and Mathematical Sciences, University of Leicester, Leicester LE1 7RH, UK
* Correspondence: yudongzhang@ieee.org

Abstract: According to the World Health Organization statistics, as of 25 October 2022, there have been 625,248,843 confirmed cases of COVID-19, including 65,622,281 deaths worldwide. The spread and severity of COVID-19 are alarming. The economy and life of countries worldwide have been greatly affected. The rapid and accurate diagnosis of COVID-19 directly affects the spread of the virus and the degree of harm. Currently, the classification of chest X-ray or CT images based on artificial intelligence is an important method for COVID-19 diagnosis. It can assist doctors in making judgments and reduce the misdiagnosis rate. The convolutional neural network (CNN) is very popular in computer vision applications, such as applied to biological image segmentation, traffic sign recognition, face recognition, and other fields. It is one of the most widely used machine learning methods. This paper mainly introduces the latest deep learning methods and techniques for diagnosing COVID-19 using chest X-ray or CT images based on the convolutional neural network. It reviews the technology of CNN at various stages, such as rectified linear units, batch normalization, data augmentation, dropout, and so on. Several well-performing network architectures are explained in detail, such as AlexNet, ResNet, DenseNet, VGG, GoogleNet, etc. We analyzed and discussed the existing CNN automatic COVID-19 diagnosis systems from sensitivity, accuracy, precision, specificity, and F1 score. The systems use chest X-ray or CT images as datasets. Overall, CNN has essential value in COVID-19 diagnosis. All of them have good performance in the existing experiments. If expanding the datasets, adding GPU acceleration and data preprocessing techniques, and expanding the types of medical images, the performance of CNN will be further improved. This paper wishes to make contributions to future research.

Keywords: COVID-19; diagnosis; deep learning; convolutional neural networks; CT images; transfer learning; X-ray images; classification

Citation: Han, X.; Hu, Z.; Wang, S.; Zhang, Y. A Survey on Deep Learning in COVID-19 Diagnosis. *J. Imaging* **2023**, *9*, 1. https://doi.org/10.3390/jimaging9010001

Academic Editor: Reyer Zwiggelaar

Received: 15 November 2022
Revised: 5 December 2022
Accepted: 16 December 2022
Published: 20 December 2022

1. Introduction

WHO declared COVID-19 as a global pandemic in March 2020. The COVID-19 pandemic has affected thousands of people. It has affected people's ordinary lives and the global economy. COVID-19 can cause respiratory, gastrointestinal, and neurological syndromes [1]. Cough, fever, and other respiratory issues are the most common symptoms [2].

Rapid COVID-19 diagnosis is essential during the pandemic. Currently, the commonly used diagnostic methods include the molecular assay, chest computed tomography (CT) scan combined with the evaluation of clinical symptoms [3], artificial intelligence (AI) methods [4], potential electrochemical (EC) biosensors [5], surface plasmon resonance (SPR)-based biosensors [6], field-effect transistor (FET)-based biosensors [7], etc. As a frequently employed auxiliary detection technology, CT images can show the changes in the lung caused by virus infection [8]. Compared with CT images, X-ray images are more accessible to obtain. It is also an important means of medical detection.

The most common diagnostic measure for COVID-19 is through reverse transcription-polymerase chain reaction (RT-PCR) assays of nasopharyngeal swabs [9]. However, the high false negative rate of RT-PCR [10] may affect the timely treatment of infected patients.

The X-ray and CT can image the lungs of patients with COVID-19. Lung imaging can reveal the niduses' spatial location and the infection's extent. CT images have a fast turnaround and excellent sensitivity [11]. They can visualize the degree of infection in the lung. Based on the significant features of COVID-19 on X-ray or CT images, many researchers have used artificial intelligence and computer vision to classify X-ray or CT images. These images were classified into two categories: those without COVID-19 and those infected with COVID-19. Other researchers divided the images into healthy, infected with COVID-19, and infected with pneumonia. Some algorithms can detect the extent of infection based on the features of X-ray or CT images. This research is meant to help doctors diagnose COVID-19 accurately and quickly.

Artificial intelligence is widely used in medical [12–14]. Its accuracy rates and prediction are high [15]. AI can be applied to multiple phases, such as prediction, diagnosis, virus detection, response, prevention, and recovery, to accelerate research [16,17]. During the COVID-19 epidemic, AI recognized chest X-rays or CT images. Features in X-ray or CT images are extracted for COVID-19 diagnosis by segmenting regions of interest and capturing fine structures. One of the important subfields of AI is machine learning (ML) [18]. It is already widely applied to medical images [19–21]. Deep learning (DL) is a promising technology in machine learning [22]. Deep learning has multiple hidden layers for learning and can perform classification or detection tasks well [23,24]. The role of deep learning in image recognition is vital [19,25–27]. The convolutional neural network (CNN) is a kind of deep network. It is popular in computer vision applications. CNN has been successfully applied to biological image segmentation [28], traffic sign recognition [29–31], face recognition [32,33], and other fields. Later studies added rectified linear units, dropout, data augmentation, and other techniques to CNN. This decreased the error rate of deep learning for image classification tasks to less than 3% in 2016 [34] and exceeded human performance.

This paper mainly summarizes and discusses methods and experiments based on convolutional neural networks, which classify chest X-ray or CT images into infected and non-infected with COVID-19. All along, CNNs have been very widely used in medical images [35–37]. The severity and spread of COVID-19 around the world are alarming. CNN is used to extract features from chest X-rays or CT images. CNN, combined with other algorithms or architectures, divided the images into two categories: infected with COVID-19 and uninfected, or three categories: infected with COVID-19, infected with other pneumonia, and uninfected [38–40]. The COVID-19 diagnosis based on convolutional neural networks can assist doctors in making judgments quickly and accurately. We gave a detailed explanation of CNN, its existing technologies, and network types. The following contents also summarize, analyze and compare the automatic diagnosis systems established by scholars based on convolutional neural networks. Finally, we suggested future research on improving the performance of COVID-19 classification models.

The parts of this paper are organized as follows: Section 2 reviews the characteristics of medical images currently commonly used for COVID-19 diagnosis. Section 3 introduces convolutional neural networks (CNN) and current methods commonly used to improve CNN. Section 4 introduces several mature and well-performing convolutional neural networks. Section 5 analyzes and discusses the current experiments and methods of applying CNN to COVID-19 diagnosis. Section 6 summarizes the full text and provides suggestions for future research. To help understand more clearly, all abbreviations and full names in this paper are shown in Table 1.

Table 1. List of all abbreviations and terms.

Abbreviation	Term
AI	Artificial intelligence
ANN	Artificial neural network
AT	Angle transformation
AVNC	Attention-based VGG-style network for COVID-19
BiLSTM	Bidirectional long short-term memories
BN	Batch normalization
CB	Convolution block
CBAM	Convolutional block attention module
CGAN	Conditional generative adversarial net
CNN	Convolution neural network
CO-IRv2	Optimized inceptionResNetV2 for COVID-19
COVID-19	Coronavirus disease 2019
CT	Computed tomography
DA	Data augmentation
DC-Net	Deep COVID network
DCNN	Deep convolutional neural network
DenseNet	Dense convolutional network
DL	Deep learning
DNN	Deep neural network
EC	Electrochemical
ELU	Exponential linear unit
FCB	Fully connected block
FET	Field-effect transistor
FMP	Fractional max pooling
GANs	Generative adversarial networks
GGO	Ground glass opacity
GoogleNet	Google inception net
GPU	Graphics processing unit
ICS	Internal covariate shift
LReLU	Leaky rectified linear unit
LSTM	Long short-term memory
mAlexNet	Modified AlexNet
ML	Machine learning
PAM-DenseNet	DenseNet with parallel attention module
PDF	Probability density function
PReLU	Parametric rectified linear unit
ReLU	Rectified linear unit
ResNet	Residual network
RICAP	Random image cropping and patching
RReLU	Randomized leaky rectified linear unit
RT-PCR	Reverse transcription-polymerase chain reaction
RVFL	Random vector functional-link net
SE block	Squeeze and excitation blocks
SPEA-II	Strength Pareto evolutionary algorithm-II
SPR	Surface plasmon resonance
SRGAN	Super-resolution generative adversarial network
TSA	Tree seed algorithm
VGG	Visual geometry group
VSBN	VGG-style base network
WCNN4	Wavelet CNN-4

2. Imaging Modalities for COVID-19 Diagnosis

Computed tomography (CT), published in 1972 [41], has become a widely used tool for diagnostic imaging. CT is a cross-sectional scan of a certain part of the human body one by one. Its advantages include clear images and fast scanning. CT is widely used to detect a variety of diseases. X-rays penetrate a person's body and take an X-ray image. X-ray images are also an important basis for diagnosing diseases.

2.1. Chest Computed Tomography

CT takes images from different angles. One shot was taken for each rotation angle. Using a large number of projection images taken from different angles, we back-calculated a fault plane image by a mathematical algorithm. This is computed tomography. Many scholars have studied the features of COVID-19 on CT images. In the study of patients in Rome and Italy [8], ground glass opacity (GGO) was found in 100% of confirmed patients on CT images. Ground-glass opacity (GGO) means that the density will be slightly increased on high-resolution CT images, and the bronchovascular will still be visible. This sign is often the manifestation of early lung disease. Timely detection and diagnosis of GGO are important for clinical management. Multilobe and posterior lung involvement was present in 93% of patients. Bilateral pneumonia occurred in 91% of patients. Cellina et al. [42] concluded that GGO was more common bilaterally in the peripheral lung areas under the pleura on CT images of COVID-19. During the disease, the number of consolidations increases, forming fibrotic stripes. Consolidation refers to accumulating fluids, fibrin, and cellular components in the alveolar airspaces. It reduces alveolar air content and increases parenchymal density. Wang et al. [43] found that CT manifestations of mild/common-type infection were multifocal lesions, GGO, involving multiple segments or lobes. CT manifestations of heavy/critical-type infection show consolidation of multiple lesions and interlobular septal thickening. In the paper [44], Bernheim et al. mentioned that the CT image features of COVID-19 are consolidative pulmonary opacities and bilateral and peripheral ground glass. The longer the onset of symptoms, the more findings on CT images. These findings include bilateral and peripheral disease, consolidation, linear opacities, greater total lung involvement, the "reverse halo" sign, and a "crazy-paving" pattern. In Guan's study [2], 56.4% of the COVID-19 subjects showed GGO.

2.2. Chest X-ray

X-rays are emitted from one end, passed through the body, and picked up by a detector at the other end. What is received is a two-dimensional image. So X-ray imaging is fast and cheap. Chest X-ray images of patients diagnosed with COVID-19 show air space consolidation and the bilateral distribution of peripheral hazy lung opacities [45]. When COVID-19 is suspected, the preferred imaging modality for the chest is the X-ray [46]. The radiation dose of chest X-rays is lower than CT, and chest X-rays are cheaper [47]. Because portable chest X-ray is easy to carry and clean, chest X-ray can reduce the risk of COVID-19 transmission during testing [48]. According to Oh [49], although the sensitivity of chest X-ray results is not high (69%) [45], chest X-rays can be used to determine the sequence of treatment for patients infected with COVID-19. Diagnosing chest X-rays can alleviate the saturated healthcare system during the COVID-19 pandemic.

3. Convolutional Neural Networks

Convolutional neural networks (CNN) are widely applied to computer vision currently. In the aspect of medical image processing about health, CNN performs outstanding [50]. CNN uses multi-layer superposition to extract from low-level features to high-level features. It is like the hierarchical structure of the human brain function [51–54]. A CNN comprises an input layer, an output layer, convolution layers, pooling layers, and fully connected layers. Input the raw data in the input layer. Convolution operations are performed to extract features in the convolution layers. The scale of parameters is further reduced in the pooling layers. Fully connected layers connect all the features and output them to the classifier.

CNN is a kind of local perception. Divide the entire image into multiple small windows that can overlap locally. The local features of the image are identified by utilizing sliding windows. A window can be regarded as a filter, which is a neuron. The convolutional layer has a set of such filters.

Figure 1 is a 6×6 image. The size of filter one is 3×3. Put filter one in the upper left corner of the 6×6 image to do the convolution operation, and obtain the result -1. The step size of the sliding window is one. Filter one moves one step to the right and then performs the

convolution operation, and the result is −2. Do convolution operation step by step from top to bottom and from left to right, and a 4 × 4 matrix can be obtained as shown in Equation (1). Doing the same operation with filter2 will obtain another different 4 × 4 matrix.

$$n_{out} = \text{floor}\left(\frac{n_{in} + 2 * p - k}{s}\right) + 1, \quad (1)$$

where n_{in} is the number of input features. p is the size of padding. k is the kernel size. s is the stride. n_{out} is the number of output features.

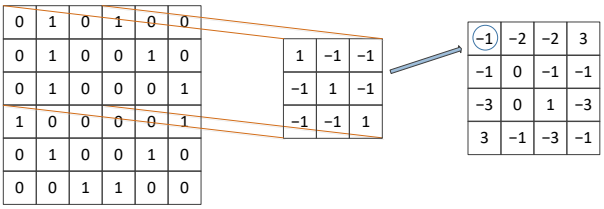


Figure 1. Convolution operation.

The dimensionality of the convolution results can be further reduced in the pooling layers. The location of the pooling layer is usually in the middle of the two convolutional layers [55]. Pooling layers reduce the feature model size of the previous convolutional layer, speed up the calculation, and reduce the probability of overfitting [56]. Pooling operations divide the feature image into regions and choose a value in one pooling area to replace. In this way, the data and parameters of the feature image are compressed. Average pooling [57] and max pooling [58] are the two most typical pooling operations.

The fully connected layers come after the convolution layers and the pooling layers. There may be one or more fully connected layers [59–61]. In the fully connected layers, each neuron is fully connected to all neurons in the previous layer. The local features of the previous outputs are integrated into global features through fully connected layers.

3.1. Batch Normalization

A deep neural network may have many hidden layers. When training each layer, the parameters are updated, causing the input data distribution of the upper layer to change accordingly. Layer-by-layer accumulation will cause the input distribution of the front layer to change drastically. There will be changes not only in the input layer but also in the hidden layers. The phenomenon is defined as Internal Covariate Shift (ICS) [62]. The front layers should constantly adapt to the update of the parameters in the last layers, which will reduce the training efficiency of the whole network.

Whitening the neural network's input layer can make the network training converge faster [63,64]. The whitening operation refers to linearly transforming the input data to reach a normal distribution with means of 0 and variances of 1. Each neuron in hidden layers can be seen as input to the next layer. Scholars can perform a whitening operation on each hidden layer neuron to solve the ICS. Transform increasingly skewed data distributions to more standard normal distributions. This is Batch Normalization (BN). BN is located after the input linear activation and before the nonlinear transformation. That is, BN is performed on the activation values a_i of neurons in each hidden layer. If there are n activation values in a mini-batch, the process of BN is as follows [62]:

(1) Calculate the mean μ, as shown in Equation (2);

$$\mu = \frac{1}{n}\sum_{i=1}^{n} a_i \quad (2)$$

(2) Calculate the variance δ^2, as shown in Equation (3);

$$\delta^2 = \frac{1}{n}\sum_{i=1}^{n}(a_i - \mu)^2 \tag{3}$$

(3) Normalize a_i^*, as shown in Equation (4);

$$a_i^* = \frac{a_i - \mu}{\sqrt{\delta^2 + e}} \tag{4}$$

where e is a constant used to ensure numerical stability.

(4) Scale and shift, as shown in Equation (5).

$$BN(a_i) = \varphi a_i^* + \omega, \tag{5}$$

φ is the parameter that scales the normalized value. ω is the parameter that shifts the normalized value. These parameters are learnable. Simply normalizing the input may lead to changing what the input represents. The addition of these parameters can restore the representation of the network.

3.2. Dropout Technology

Between the input and the output layers, there are hidden layers. A deep neural network (DNN) usually has many hidden layers. The training set can be modeled correctly by adapting the weights on the incoming connections of the neurons in hidden layers [60]. However, the weight matrix performs poorly on the test set and has poor generalization ability. This phenomenon is called overfitting.

Dropout can solve the problem of overfitting. Dropout means that some neurons in the network are dropped out. However, instead of being deleted, they are temporarily dropped out of the network, including their output and input connections [65], as shown in Figure 2. The neurons that are dropped out are randomly selected. Each neuron is left with a fixed probability p. The neurons that are left form new networks thinner than the original network. These networks use the same weight matrix, so there is no increase in the number of parameters.

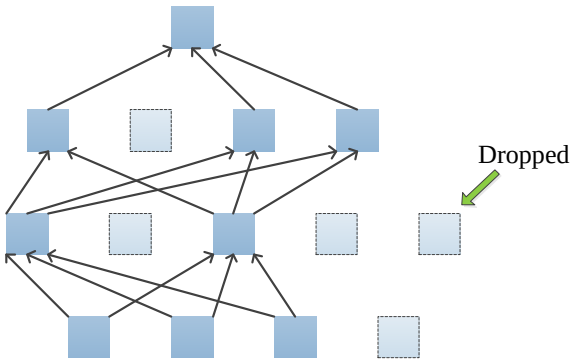


Figure 2. Dropout neural net.

When the network is training, a neuron is left with probability p (p is randomly generated from a Bernoulli distribution). It is connected to the next layer with a weight w. Then the weight value is ultimately $p \times w$. To make the weights connecting the neurons in the next layer consistent with those during training, the neurons during testing are always present, and the weights are reduced to $p \times w$, as shown in Figure 3.

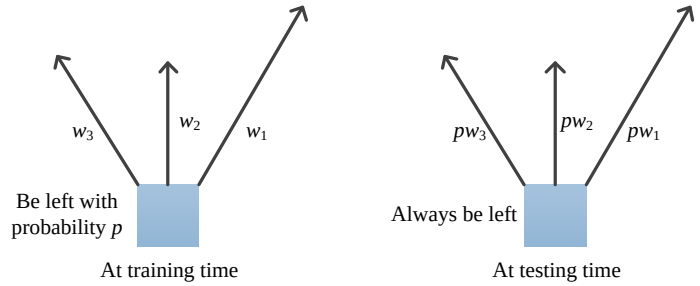


Figure 3. The output during testing is consistent with the output during training.

The dropout operation of the neural network is as Equation (6) [65]:

$$\begin{aligned} \widetilde{d}^l &= r^l \times d^l \\ c_i^{l+1} &= w_i^{l+1} \times \widetilde{d}^l + b_i^{l+1} \\ d_i^{l+1} &= f\left(c_i^{l+1}\right) \end{aligned} \tag{6}$$

where r^l is a vector produced by the probability p^l of the neurons in the hidden layer l. d^l is the output vector of hidden layer l. c^l is the input vector of the hidden layer l. w^l and b^l are the weights and biases of hidden layer l. $f()$ is the activation function. $\widetilde{d}^l$ is the output vector of hidden layer l with the dropout. $\widetilde{d}^l$ is processed to be used as the input of the hidden layer $l+1$.

3.3. ReLU Function and Its Variants

3.3.1. ReLU

As the activation function, ReLU functions increase the nonlinear relationship between the layers of CNN to complete complex tasks. The neurons of each layer in the convolutional neural network are weighted and shifted, then output to the next layer, as shown in Equation (7).

$$y = wx + b, \tag{7}$$

where x is the input vector, y is the output vector. w is the weight vector for the layer, and b is the bias vector for the layer. If there are two hidden layers, the input is weighted and biased twice to obtain the output, as shown in Equations (8) and (9).

$$y = w_2(w_1x + b_1) + b_2 \tag{8}$$

$$y = w_2w_1x + w_2b_1 + b_2 \tag{9}$$

From Equation (9), it can be concluded that multiple hidden layers can be combined into one layer if only linear transformations of weights and biases are applied to the input. The linear expression ability is limited and insufficient to fit nonlinear neural networks. The activation functions process the outputs of the neurons in the previous layers. The processing results are passed as inputs to the neurons of the next layer. Activation functions can add nonlinear factors. Rectified linear unit (ReLU) is a kind of activation function commonly used in CNN. ReLU function is defined as Equation (10):

$$\mathrm{ReLU}(m) = \begin{cases} 0 & m < 0 \\ m & m \geq 0 \end{cases} \tag{10}$$

ReLU is illustrated in Figure 4a. The ReLU function is a piecewise linear function. It has unilateral inhibition, which enables sparse activation of neurons in a deep neural network. Sparsity networks are more helpful for mining features. The ReLU function accelerates the convergence, and the gradient vanishing problem is solved [66]. A large parameter update may cause some ReLU neurons to never be activated. The phenomenon of "dying

ReLU" appeared [67]. Variants of the ReLU function appeared to solve the problem, such as LReLU (Figure 4b), PReLU (Figure 4c), RReLU (Figure 4d), ELU (Figure 4e), etc.

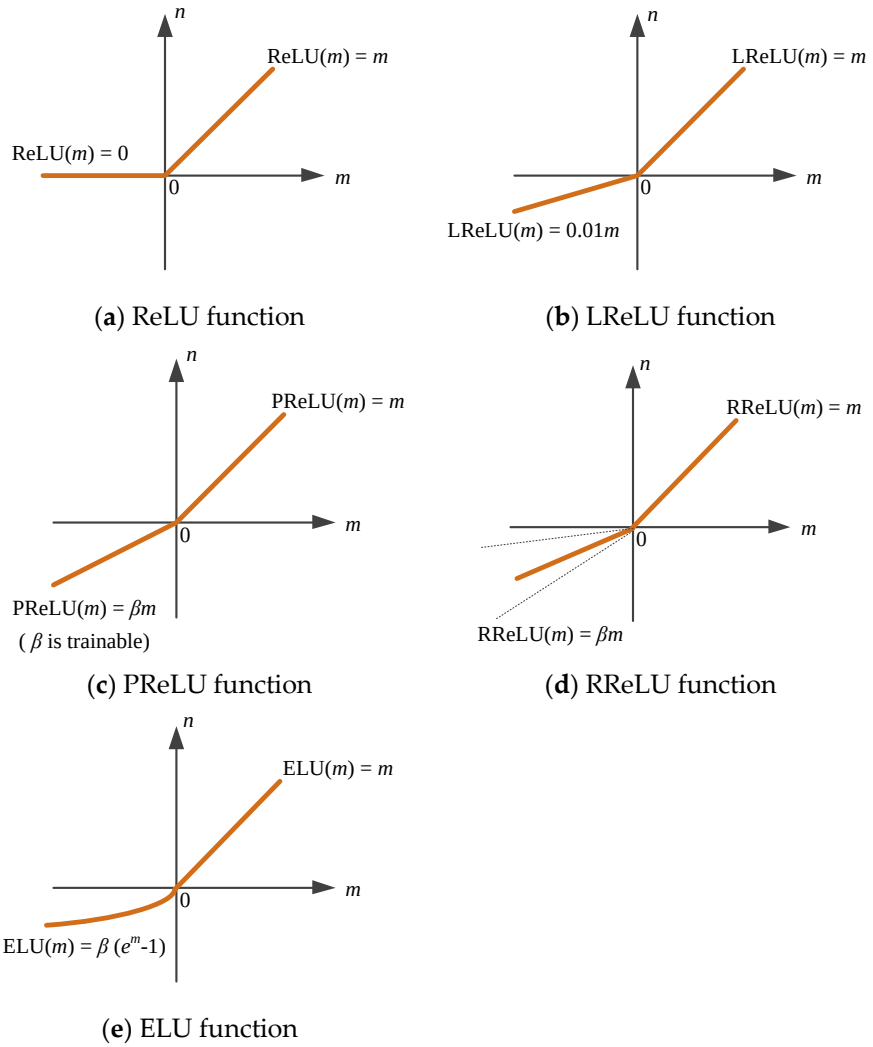


Figure 4. ReLU function and its variants.

3.3.2. Leaky ReLU

Leaky ReLU (LReLU) solves the dying ReLU problem using a small slope [68]. The LReLU function is defined as Equation (11):

$$\mathrm{LReLU}(m) = \begin{cases} \beta m & m < 0 \\ m & m \geq 0 \end{cases}. \tag{11}$$

where β is generally 0.01. The LReLU has the advantage that if the input of the LReLU activation function is less than zero, the gradient can also be calculated, unlike ReLU, which keeps the gradient zero.

3.3.3. Parametric ReLU

Parametric rectified linear unit (PReLU) is a variant of LeakyReLU. The parameter β of PReLU [69] is not set artificially but is obtained through training. The parameter is the key to improving the classification performance. PReLU is defined as Equation (12):

$$\mathrm{PReLU}(m) = \max(0, m) + \beta \min(0, m) \tag{12}$$

When $\beta = 0$, PReLU becomes ReLU. β controls the slope of the negative axis.

3.3.4. Randomized Leaky ReLU

Randomized leaky rectified linear unit (RReLU) is another variant of LeakyReLU. RReLU is defined as Equation (13). Parameter β is randomly valued during training and becomes a fixed value during testing. During the training process, the distribution of β satisfies the normal distribution with a standard deviation of 1 and a mean of 0. When testing, Srivastava [70] et al. took an average of all β using the method of dropout. The NDSB competitor suggested that it would be better if β is sampled from C (3,8).

$$\mathrm{RReLU}(m) = \begin{cases} \beta m & m < 0 \\ m & m \geq 0 \end{cases} \quad \beta \sim C(x,y),\ x < y \ and \ x,y \in [0,1) \tag{13}$$

3.3.5. Exponential Linear Unit

The exponential linear unit (ELU) is also an activation function. It has similarities and differences with ReLU. ELU is an activation function with negative values, and it is defined as Equation (14):

$$\mathrm{ELU}(m) = \begin{cases} m & m > 0 \\ \beta(e^m - 1)\ m \leq 0 \end{cases}, \ \mathrm{ELU}'(m) = \begin{cases} 1 & m > 0 \\ ELU(m) + \beta\ m \leq 0 \end{cases} \quad (\beta > 0) \tag{14}$$

where β controls the saturation value for negative net input [71]. ELU solves the problem of gradient vanishing. The mean values of the ELU output are close to zero, which makes the convergence faster. Because ELU is nonzero for negative values, there is no "dying ReLU" phenomenon. ELU is an unsaturated function, and there is no problem with vanishing gradients or exploding gradients.

3.4. Pooling

Pooling usually follows the convolution operation. The essence of pooling is sampling. The dimension of the input feature map can be reduced using pooling by choosing an appropriate method. Pooling can reduce the operation's complexity and improve the operation's speed. The addition of pooling layers in the architecture of CNN can reduce the scale of weights and parameters [72], reducing the input's dimensions and memory consumption [73–75]. The use of pooling layers can also solve overfitting [76].

3.4.1. Max Pooling

Max pooling is an extensively applied pooling method in CNNs [58]. This method is popular because it does not require tuning parameters. The max pooling technique is to find the max element in the pooling region [77,78]. Max pooling can exhibit the max value of the feature map in the $k \times k$ neighborhood. Not only the scale of the feature map space is optimized, but also the translation invariance of the network is preserved by max pooling [79].

In Max pooling, the whole image is divided into several blocks of the same size that do not overlap. We discard all other nodes and take only the max value in each block. For example, there is a pooling region of size 4×4. The pooling filter size is 2×2, and the stride is 2. These setups enable pooling to be applied to regions of the image that do not overlap. The pooling filter selects the max value of each block to obtain the final output. Figure 5 shows the process of max pooling. Max pooling focuses on the max element and discards the others. This may lead to wrong results of the disappearance of salient features [80].

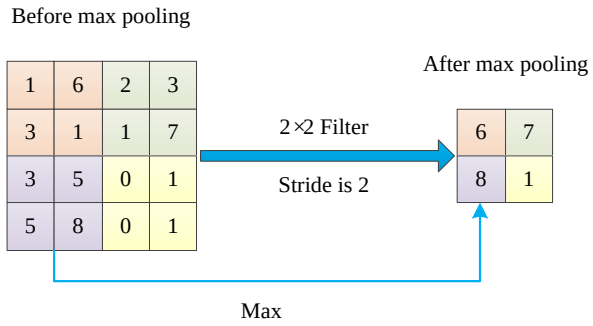


Figure 5. The process of max pooling.

3.4.2. Average Pooling

The input sample is divided into multiple blocks of the same size and not overlapping each other. Average pooling is to calculate the average value of all elements in each block and present output. Use the input sample in Section 3.4.1. Similarly, the average pooling filter size is 2×2, and the stride is 2. The purpose is to ensure that the areas swept by the pooling filter do not overlap. Calculate the average of all values in each 2×2 pooling block as the output. Figure 6 shows the process of average pooling. In average pooling, if the averages are low, the contrast will be diminished [79]. The convolutional features will be decreased [81] if most of the averages are zero. Wang et al. [82] proposed using a rank-based average pooling module in the network for COVID-19 COVID-19 recognition to avoid overfitting.

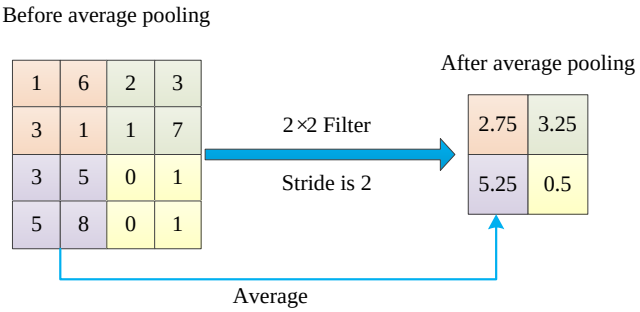
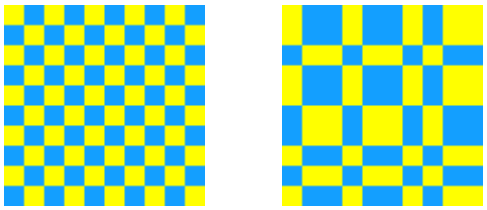


Figure 6. The process of average pooling.

3.4.3. Fractional Max Pooling

Fractional max pooling (FMP) is a special max pooling. It has an important parameter μ called the pooled fraction. Because μ represents the ratio of the input-spatial size to the output-spatial size of the pooling region, the image size can be reduced by setting μ. For regular max pooling, μ is set to an integer. The size of the feature map is rapidly reduced. For fractional max pooling, μ is set to a non-integer (a fraction). The decay rate of the feature map will slow down. If the focus is on accuracy, set $\mu \in (1, 2)$, as shown in Figure 7. Suppose the focus is on speed, set $\mu \in (2, 3)$ [83].



(**a**) Original image (**b**) Result of set $\mu = 1.4$

Figure 7. Illustration of FMP.

There are two types of pooling regions: disjoint rectangular areas and overlapping square areas. The pooling regions can be chosen in pseudorandom and random ways. FMP

works better in a random way [83]. Wang et al. [84] used fractional max pooling (FMP) instead of max pooling and average pooling in a network for COVID-19 recognition.

3.4.4. Other Popular Pooling Methods

Both average pooling and max pooling have disadvantages. Yu et al. [85] proposed a way to mix the two. This method is called mixed pooling. The weights of average pooling and max pooling are merged [86], which can overcome their shortcomings. In many experiments, the performance of mixed pooling is better than that of max pooling and average pooling in image classification [87–89].

Tree pooling [90] uses the data from pooling filters to learn and combines these learned filters. The performance of tree pooling is better than that of average pooling, max pooling, and mixed pooling. However, more parameters must be learned than mixed average and max pooling [91]. Tree pooling is suitable for lower network layers focusing on functional responses [92].

Fergus and Zeiler et al. [59] proposed stochastic pooling. Stochastic pooling randomly selects values. It is not suitable for negative activations. With little training data, stochastic pooling tends to generate strong activation, leading to overfitting. Rank-based stochastic pooling [93] may solve this problem. It estimates the ranking of the activations within the pooling region. Stochastic pooling needs to address issues related to scaling [94]. Wang et al. [95] proposed the stochastic pooling module and a stochastic pooling neural network for COVID-19 diagnosis. Zhang et al. [96] proposed a deep-learning model for COVID-19 diagnosis. It used stochastic pooling instead of max pooling and average pooling.

4. Data Augmentation

Data augmentation (DA) is an effective way to expand datasets. The larger the size of the data, the better the generalization ability of the trained model. When the number of samples in the dataset is relatively small, data augmentation increases the amount of training data by modifying the existing training data [97]. Data augmentation is mainly used to prevent overfitting, especially when the training dataset is small. There are many ways for data augmentation [98–101].

4.1. Geometric Transforms

Geometric transforms are standard methods of data augmentation. Training data can be effectively increased with geometric transforms [102]. Flipping is one of the easiest ways to implement data augmentation [103]. Flipping is a mirror image reflected along a line. Horizontal axis flipping is commonly used. Figure 8a shows the original image. Figure 8b shows the image after horizontal flipping. Figure 9a shows the original image. Figure 9b shows the image after vertical flipping. No matter what direction the image is flipped, the image remains the same.

(**a**) Original image (**b**) Flipped image

Figure 8. Horizontal flipping.

(**a**) Original image (**b**) Flipped image

Figure 9. Vertical flipping.

Rotation augmentation rotates the image to the left or right according to the selected angle. The image is rotated around a central point [104]. The angle of rotation should be chosen appropriately. A wide rotation may make recognition unsafe [105]. Figure 10a shows the original image. Figure 10b shows the result of a clockwise rotation of 10 degrees.

(**a**) Original image (**b**) Clockwise rotation of 10 degrees

Figure 10. Rotation operation.

Shear is the non-vertical projection effect of a plane object on the projected plane. Horizontal shear and vertical shear are commonly used in data enhancement. If the original coordinate of a point is $(x,\ y)$, the coordinate after horizontal shear is (x'_h, y'_h), as shown in Equation (15),

$$[x'_h,\ y'_h, 1] = [x, y, 1] \times \begin{bmatrix} 1 & 0 & 0 \\ \beta_h & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{15}$$

where β_h is horizontal shear parameter. The coordinate after the vertical shear is (x'_v, y'_v), as shown in Equation (16), where β_v is vertical shear parameter.

$$[x'_v,\ y'_v, 1] = [x, y, 1] \times \begin{bmatrix} 1 & \beta_v & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{16}$$

Cropping cuts a portion of the input image. It is an effective tool for extracting patches [106,107]. Images can be classified according to patches rather than the whole image. An image can be cropped into several pieces. It does not lose important information about the image when cropping.

4.2. Noise Injection

There are many ways for data augmentation using noise injection, such as Gaussian noise, salt-and-pepper noise, speckle noise, etc. Noise injection has been successfully used in plant leaf disease recognition [108], robot speech commands [109], fruit classification [110], and so on. Gaussian noise injection is more popular. Gaussian noise follows the normal distribution. The probability density function (PDF) $f(x)$ of Gaussian noise can be expressed

as Equation (17), where α^2 is the variance and β is the mean value. Gaussian injection means injecting random values from a Gaussian distribution into the pixels of the image.

$$f(x) = \frac{1}{\sqrt{2\pi\alpha^2}} e^{-\frac{(x-\beta)^2}{2\alpha^2}} \tag{17}$$

In addition to adding noise to the input layer, you can also add noise to other layers. DeVries et al. add noise to a learned feature space instead of the input space [111]. Xie et al. [112] add noise to the loss layer. These prevent the network from overfitting.

4.3. Color Space

Implementing data augmentation in a color space is also a practical approach. The transformations are based on gray or RGB color values [113]. Color augmentation can be completed by isolating a color channel and converting an image into a representation in one color channel. Color augmentation can be completed by manipulating the RGB values to change the image's brightness. Color augmentation can be completed by changing the color histograms of the image [98].

4.4. Random Erasing

Random erasing means randomly selecting an area of the image, and its pixels are erased by random or mean pixel values [114]. It can augment the recognition of occluded images. Random erasing requires no parameters to learn and is easy to implement.

The size of the source image is $M \times N$. The size of the randomly selected erasing rectangle region is $M' \times N'$. Then, select randomly a point $W = (x, y)$ in the source image region. If $x + M' < M$ and $y + N' < N$, $S = (x, y, x + M', y + N')$ is the erasing region. Each pixel in S is assigned a random value between 0 and 255. Finally, a new image with a part of the regions erased is obtained.

4.5. Kernel Filters

Kernel filters can sharpen or blur images. The data augmentation can be obtained by convolving the kernel filter with the image. A Gaussian blur filter [115] is used to slide an $n \times n$ matrix across the image. A blurry image is yielded. An unsharp masking [116] slides an $n \times n$ matrix over the image. A sharpened image is yielded. The blurred image can resist motion blur. The sharpened image allows for more detail. Figure 11a shows the original image. Figure 11b shows the image after sharpening. Figure 11c shows the image after blurring.

(**a**) Original image (**b**) Sharpened image (**c**) Blurry image

Figure 11. Sharpening and Blurring operation.

4.6. Mixing Images

Mixing images is a method for data augmentation of two or more images. Calculating the average of the pixels can achieve a mixed image. This strategy of data augmentation may seem irrational, but it is effective. Ionue [117] came up with Samplepaining technology. Two images were randomly selected from the dataset, and one was overlaid on another. A new sample is obtained by superimposing two images. This simple data augmentation technology significantly improved classification accuracy. Summers and Dinneen [118] mixed the images in a nonlinear manner. Takahashi et al. [119] generate a new training image by mixing four images obtained through random image cropping and patching (RICAP).

4.7. Data Augmentation Methods Based on Deep Learning

There are some DA ways based on deep learning. The feature space in CNN is a low-dimensional representation in high-level layers [98]. Feature space augmentation opens up opportunities for many vector operations for data augmentation [111]. Adversarial training helps to search the space for possible augmentations. It can improve the weakness in the boundaries of learnable decisions. Generative Adversarial Networks (GANs) are a way to obtain additional information from a dataset [120]. Figure 12 shows the architecture of GAN.

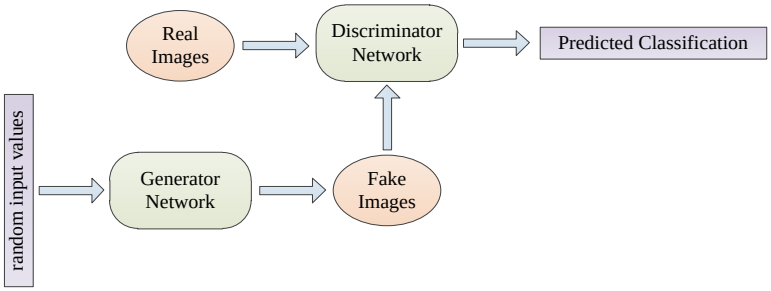


Figure 12. GAN architecture.

Neural style transfer transfers the style of an image created in CNN to another image and preserves the original content [121]. Overfitting of the model can be prevented. Meta-learning refers to optimizing neural networks with neural networks [122]. There is neural augmentation [123], smart augmentation [124], and auto-augmentation [125].

5. Pretrained Models

With the development of CNN, many mature networks perform well in image recognition, object recognition, natural language processing, and other fields. These networks include AlexNet, ResNet, DenseNet, VGG, GoogleNet, etc. During the COVID-19 pandemic, these networks could be fine-tuned and used to classify chest CT or chest X-ray images into different categories (such as infection and health).

5.1. AlexNet

AlexNet contains five convolutional layers and three fully connected layers [66]. Compared with LeNet, it uses more new technology [126]. Figure 13 is the structure of AlexNet.

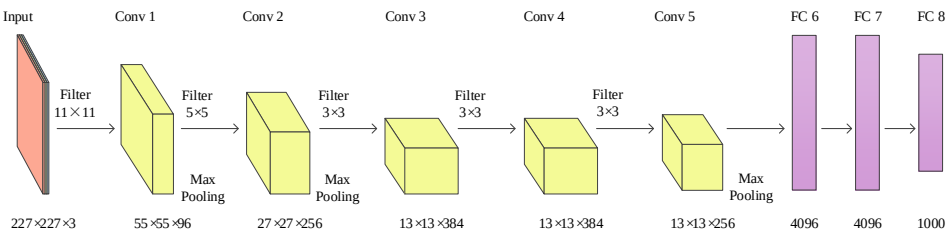


Figure 13. Structure of AlexNet.

(1) AlexNet uses ReLU as the activation function. ReLU solves the problem of gradient descent.
(2) For fully connected layer training, AlexNet randomly ignores some neurons using dropout to avoid model overfitting.
(3) AlexNet uses overlapping max pooling in CNN. The step size in AlexNet is smaller than the size of the pooling kernel, so overlapping pooling is obtained.
(4) AlexNet uses the local normalization scheme, which is more helpful for generalization.
(5) AlexNet uses the computing power of parallel GPU, and the training of CNN is accelerated.
(6) Two different forms of data augmentation are used in AlexNet. The first form of data augmentation is the translation and horizontal reflection. The second form of data augmentation is the change in the intensity of the image color channel.

5.2. ResNet

ResNet is developed and optimized based on AlexNet. One of the advantages of residual neural networks is identity mapping. ResNet has 152 layers. Simply increasing the depth of the network can lead to the problem of vanishing gradients or exploding gradients [127,128]. This problem has been addressed by normalized initialization [63,69,128,129] and intermediate normalization layers. However, with the network layers stacking, the accuracy of the training dataset saturates or even descends. That is the degradation. The degradation is not due to overfitting. Stacking layers in the deep model will make the training error larger [130,131]. ResNet uses a deep residual learning framework to mitigate the degradation [132].

The shallow network gradually stacks layers to make it a deep network. In a deep network, if the stacked layers are identity mapping, and the other layers copy the shallow network, the performance can be almost the same as the shallow network. A few stacked layers are called a block. x is the input to the first layer of the block. The expected mapping is $G(x)$. The fitting function of the block is Equation (18):

$$D(x) = G(x) - x. \tag{18}$$

So $G(x)$ is recast into $D(x) + x$. It is easier to approximate $D(x)$ to $G(x) - x$ than to approximate $D(x)$ to $G(x)$. $G(x) - x$ is the residual mapping. To the extreme, the identity mapping $G(x) = x$ can be obtained by pushing the residual $D(x)$ to zero. Figure 14 shows that feedforward neural networks with shortcut connections can achieve $D(x) + x$.

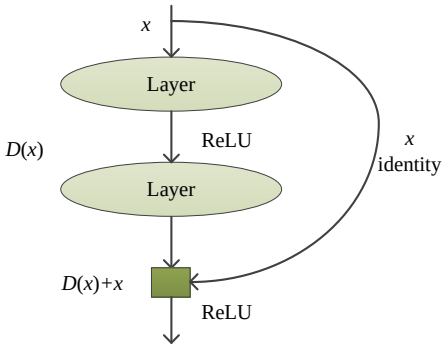


Figure 14. Shortcut Connection.

5.3. DenseNet

All the front layers in DenseNet are densely connected to the back layers. Each layer connects to every other layer. One of its characteristics is that the connection of features on channels enables feature reuse. DenseNet has fewer parameters, faster computation, and better performance.

DenseNet is a densely connected convolutional network. Feature maps from all preceding layers are concatenated as additional inputs to this layer. Its feature maps are passed on as inputs to all subsequent layers, which preserves the feedforward nature of

the network. If DenseNet has N layers, there are $N(N+1)/2$ connections. x_0 is the input image of a DenseNet. x_n is the output of the network at layer n. Layer n receives feature maps from all preceding layers as inputs (19):

$$x_n = F_n([x_0,\ x_1, \cdots, x_{n-1}]), \tag{19}$$

where $F_n()$ is a nonlinear transformation function of layer n. It is a composite function of operations [133]. It includes batch normalization (BN), rectified linear units (ReLU), pooling [126], or convolution.

Pooling and convolution in CNN will change the size of feature maps. In DenseNet, only feature maps of the same size can be densely connected. To resolve this contradiction, DenseNet divides the network into several dense blocks. The feature maps of each layer in one dense block have the same size. They are densely connected. Between two adjacent dense blocks are transition layers. Pooling and convolution in the transition layer make the feature map smaller. Figure 15 is the structure of DenseNet.

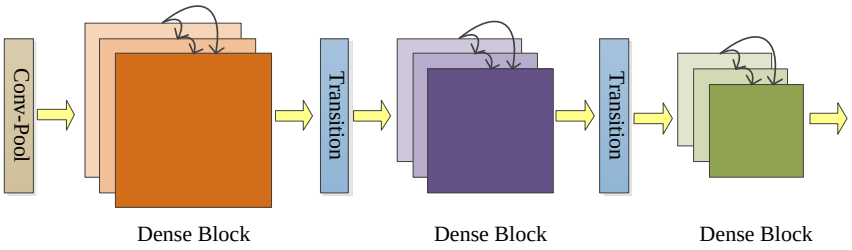


Figure 15. Structure of DenseNet.

5.4. VGG

Visual Geometry Group (VGG) won first place in localization and second place in the classification on ImageNet Challenge in 2014. VGG is an improvement on AlexNet by deepening the depth of the network. VGG uses 3×3 convolution filters in all layers. It enables the network to add more convolutional layers. Increasing the depth of the network leads to better performance. Figure 16 is the structure of VGG16.

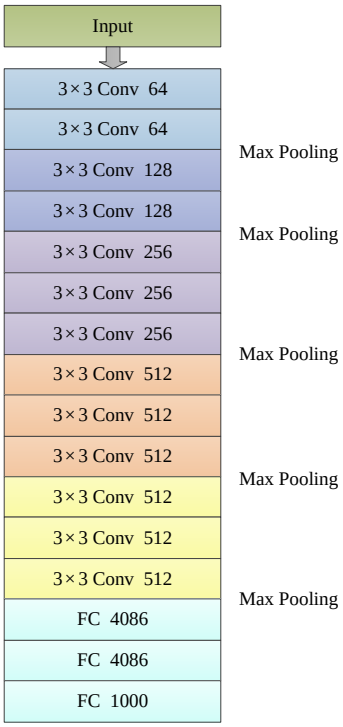


Figure 16. Structure of VGG16.

VGG uses a stack of three 3×3 convolution kernels. Three 3×3 convolution kernels have the same receptive field of 7×7. The three nonlinear rectification layers make the decision function more discriminative. Then, the number of parameters is decreased. If the input and the output of three 3×3 convolution layers both have X channels, the number of parameters is $3 \times 3^2 \times X^2 = 27X^2$. One 7×7 convolution layer has the parameters is $7^2 \times X^2 = 49X^2$, 81% more parameters [61].

VGG uses the max pooling layer. The size of the pooling kernel is 2×2, with stride 2. Smaller pooling kernels can capture more details of the information. Max pooling is easier to capture the changes in the image and obtain more differences in local information.

5.5. GoogleNet

GoogleNet believes that increasing the number of layers (depth) or the number of neurons at each layer (width) of the networks can improve the performance of the networks. Increasing the size of a network has two drawbacks:

(1) Larger sizes of networks generate more parameters. When the data in the training set is small, too many parameters will cause the network to overfit.
(2) Larger size of networks increases computation dramatically. Adding a convolution layer will result in a quadratic increase in computation.

To overcome the two drawbacks, GoogleNet uses a sparse connection instead of a full connection. Inception architecture is constructed in GoogleNet to realize sparse connections [134]. The idea of Inception is to find the optimal local sparse structure in the network and repeat it. The filter sizes of Inception are 1×1, 3×3, and 5×5. The output filter banks concatenate into a single vector as the input of the next step. At the same time, it is also necessary to add an alternative parallel pooling path in each stage, as shown in Figure 17.

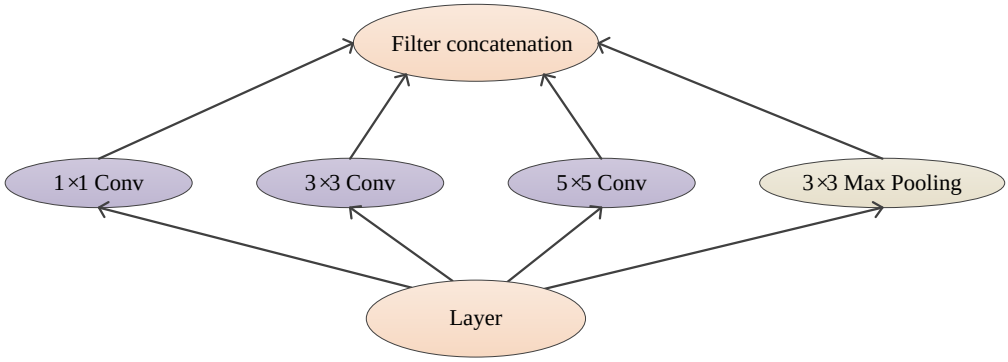


Figure 17. Basic Inception Module.

But in Inception architecture, the outputs of the convolutional layer and the pooling layer will increase. It can become expensive and a computational blow-up. Dimension reductions and projections can solve this problem. Using 1×1 convolutions can reduce the computation before 3×3 and 5×5 convolutions and rectify linear activation, as shown in Figure 18.

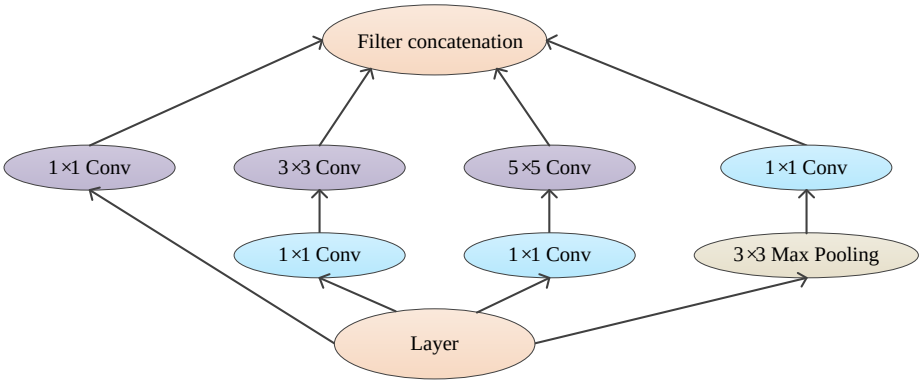


Figure 18. Inception module with 1 × 1 convolution.

5.6. Transfer Learning

In some cases, it is not easy to obtain training data that matches the feature space of the test data [135]. If a test dataset is related to a training dataset that has already been trained, the parameters of the trained model (pre-trained model) can be transferred to the target model, as shown in Figure 19. This avoids training the target model from scratch. It speeds up and optimizes the learning of the target model. Transfer learning can provide related but not identical existing datasets for the target model.

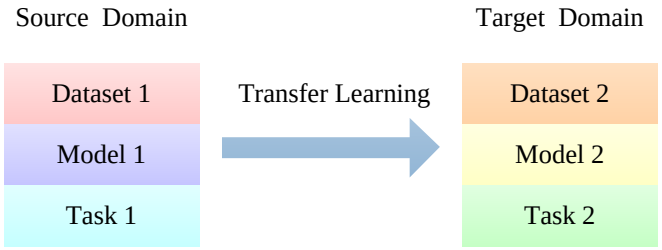


Figure 19. The process of transfer learning.

Transfer learning has been applied to include text sentiment classification [136], image classification [137–139], human activity classification [140], software defect classification [141], and multi-language text classification [142–144]. The networks that can be used as pre-trained are also the focus of research.

The feature space of the source domain is F_s, and the feature space of the target domain is F_t. If $F_t = F_s$, this is homogeneous transfer learning. If $F_t \neq F_s$, this is heterogeneous transfer learning. The strategies adopted by homogeneous transfer learning [135] include correcting the difference in the marginal distribution in the source, correcting the difference in the conditional distribution in the source, or correcting the difference between the marginal and conditional distribution in the source. Heterogeneous transfer learning [135] uses strategies that align the input spaces of the source and target domains. If the domain distributions are still unequal, further modulations are needed.

There are three strategies for transfer learning: (i) inductive transfer learning, (ii) transductive transfer learning, and (iii) unsupervised transfer learning [145,146]. There are two common types of inductive learning. One is to use the source domain to obtain a trained learning model and fine-tune different target layers [147]. Another is multi-task—learning multiple tasks simultaneously from the same input [148]. The target task is similar to the source task in transductive transfer learning, but the domains differ. The factors affecting transductive transfer learning include domain adaptation [149], sample selection bias [150], and covariate shift [151]. Unsupervised transfer learning mainly solves unsupervised learning tasks in the target domain.

There are four kinds of approaches to transfer learning [145]. The first is instance-based transfer learning. A common approach is to reweight instances of the source domain [152–154].

These worked best when the conditional distributions were the same in the source domain and target domain. The second is feature-based transfer learning. The standard methods include asymmetric feature transformation [155] and symmetric feature transformation [156]. The third is parameter-based transfer learning. Source tasks and target tasks share the parameters of the models [157,158]. Target learners are formed by optimally combining the reweighted source learners [158–160]. The last one is relational-based transfer learning [161–163]. If the source and target data have similar relationships, the relationship among the data can be transferred. Figure 20 shows the classification of transfer learning.

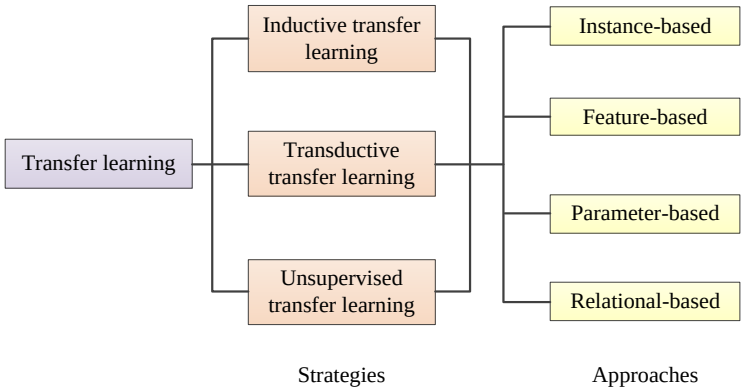


Figure 20. Strategies and approaches of transfer learning.

6. Application, Analysis, and Discussion

Currently, the diagnosis of COVID-19 mainly relies on epidemiological history, clinical symptoms, laboratory results, chest imaging findings, nucleic acid testing, or homologous comparison of gene sequencing [164]. COVID-19 is highly contagious and has an incubation period. It needs repeated testing over some time to determine infection. It is difficult for doctors to diagnose COVID-19 quickly and accurately. Chest computed tomography (CT) and chest X-ray images can reflect whether the patient's lung is infected and the degree of damage. However, doctors rely on experience to judge CT images, and X-ray images can also lead to misdiagnosis. Popular artificial intelligence (AI) systems can quickly analyze large amounts of image data. After learning, the system can accurately classify chest CT images and X-ray images. This will help doctors diagnose COVID-19 quickly and accurately.

6.1. CNN Applied to CT Images

The papers mentioned in Table 2 used convolutional neural networks (CNN) to analyze chest CT images and proposed various algorithms for COVID-19 diagnosis. These algorithms modify the parameters and architecture of the convolutional neural network (CNN) to obtain better accuracy.

Aslan et al. [165] improved the architecture of AlexNet to obtain mAlexNet. In the mAlexNet architecture, the fully connected layer eight has 25 neurons instead of 1000 neurons. The features extracted by mAlexNet are given into the tree seed algorithm (TSA) ANN structure for classification. The experimental comparison shows that the performance of mAlexNet + TSA-ANN is more excellent. AlexNet and mAlexNet can also obtain good performance by hybridizing other structures or algorithms [166,167]. Different total number of layers [168,169] in ResNet leads to different performance results. Rahimzadeh et al. [170] modified ResNet50V2. This model can retain the data of small objects and improve the classification accuracy of images containing small important objects. By comparison, Loey et al. [171] found that among AlexNet, VGGNet16, VGGNet19, GoogleNet, and ResNet50, the ResNet50 was the most appropriate deep learning model for using the classical data augmentation and CGAN. Özdemir et al. [97] extended ResNet50 architecture with a feature-wise attention layer and used the mixup data augmentation technique. This architecture achieves higher accuracy. Mondal et al. [172] proposed the scheme of optimized InceptionResNetV2 for COVID-19 (CO-IRv2). It combines InceptionNet with

ResNet with hyperparameter tuning. DenseNet-121 is a network with 121 layers [173]. DenseNet-121 D solves the problem of vanishing gradients, allows better feature reuse, and reduces the number of parameters. It is more beneficial to the training of deep learning models [174]. Xiao et al. [175] improved DenseNet with a parallel attention module (PAM-DenseNet) which has spatial and channel attention modules. These make the net have better classification performance and patient-wise prediction performance. The VGG network alone [176], or together with other neural networks [177], has been used to classify chest CT images with high accuracy. Wang et al. [178] proposed a VGG-style base network (VSBN). Convolutional block attention module (CBAM) serves as the attention module of VSBN. In order to solve the problem of artificial intelligence (AI) model overfitting, VSBN uses an improved multiple-way data augmentation method. GoogleNet is retrained over COVID-CT-Dataset [179]. GoogleNet learned the variations present in diverse types of CT images. The system model GoogleNet-COD developed by Yu et al. [180] takes GoogleNet as the backbone network. It removes the last top-two layers and replaces them with four new layers, which include the dropout layer, two fully connected layers, and the output layer. Zhang et al. improved the deep convolutional neural network (DCNN) [181]: the pooling layer adopts stochastic pooling; construct a convolution block (CB), which is obtained by combining the convolution layer with the batch normalization layer; construct a fully connected block (FCB), which is obtained by combining the dropout layer with the fully connected layer. Pham [182] uses multiple CNNS to classify CT images collected from COVID-19 patients and non-COVID-19 subjects. Among them, the deepest net, DenseNet201, has the best performance. Transfer learning with the direct input of whole image slice and without using data augmentation provided better classification rates. JavadiMoghaddam et al. [183] proposed a convolutional neural network structure containing a wavelet and four convolutional layers. It optimizes convergence time using batch normalization (BN) and Mish Functions. The Haar wavelet transform occurs at the pooling layer. There are squeeze and Excitation blocks (SE blocks) after each dropout layer.

Table 2. Performances of CNN in COVID-19 Diagnosis using CT images.

Reference	Method	Performances	Datasets
[165]	mAlexNet	Accuracy: 97.92%, Sensitivity: 98.20%, Specificity: 97.68%, Precision: 97.32%, F1 score: 97.76%	SARS-CoV-2 Ct-Scan Dataset: 2482 chest CT scans
[165]	mAlexNet + TSA-ANN	Accuracy: 98.54%, Sensitivity: 97.75%, Specificity: 99.23%, Precision: 99.09%, F1 score: 98.41%	
[166]	mAlexNet + BiLSTM	Accuracy: 98.70%	COVID-19 Radiography Database
[167]	DC-Net-RVFL	Accuracy: 90.91%, Sensitivity: 85.68%, Specificity: 96.13%, Precision: 95.70%, F1 score: 90.41%	A Private Dataset: 296 lung window images
[168]	ResNet18	Accuracy: 86.70%, Precision: 80.80%, Recall: 81.50%, F1 score: 81.10%	A Private Dataset: 618 chest CT samples
[169]	ResNet50	Accuracy: 76%, Specificity: 61.50%, Recall: 81.10%, AUC: 0.8190	A Private Dataset: 495 chest CT images
[170]	Modified ResNet50V2	Accuracy: 98.49%, Recall: 96.83%	COVID-Ctset: 63849 chest CT images
[171]	ResNet50 + Data augmentations + CGAN	Accuracy: 82.91%, Sensitivity: 77.66%, Specificity: 87.62%.	COVID-19 CT Scan Digital Images Dataset: 742 chest CT images
[97]	ResNet50+Attention+mixup	Accuracy: 95.57%	COVID-CT Dataset: 1596 chest CT images

Table 2. *Cont.*

Reference	Method	Performances	Datasets
[172]	CO-IRv2 Adam	Accuracy: 94.97%, Specificity: 96.52%, Precision: 96.90%, F1-Score: 95.24%, Recall: 93.63%, Execution Time(sec): 717	A New Dataset: 2481 chest CT images
[172]	CO-IRv2 Nadam	Accuracy: 96.18%, Specificity: 95.08%, Precision: 95.35%, F1-Score: 96.28%, Recall: 97.23%, Execution Time(sec): 707	
[172]	CO-IRv2 RMSProp	Accuracy: 96.18%, Specificity: 99.18%, Precision: 99.16%, F1-Score: 96.13%, Recall: 93.28%, Execution Time(sec): 749	
[173]	DenseNet-121	Accuracy: 92%, Recall: 95%	A Real Patient Image Dataset: 2482 chest CT images
[175]	PAM-DenseNet	Accuracy: 94.29%, Sensitivity: 95.74%, Specificity: 96.77%, Precision: 93.75%	Dataset 1: A Lung CT Slices Dataset, 3530 chest CT slices Dataset 2: A Lung CT Scans Dataset, 280 chest CT scans
[176]	VGG-19	Accuracy: 94.52%	COVID-19 CT Dataset: 738 chest CT scan images
[177]	SRGAN +VGG16	Accuracy: 98.0%, Sensitivity: 99.0%, Specificity: 94.9%	COVID-CT-Dataset: 470 chest CT images
[178]	AVNC	The sensitivity, precision, F1 all above 95%	A Private Dataset: 1164 slice images
[179]	GoogleNet	Accuracy: 82.14%	COVID-CT-Dataset: 349 chest CT images
[180]	GoogleNet-COD	Accuracy: 87.50%, Sensitivity: 90.91%, Specificity: 84.09%	A Private COVID-19 Dataset: 148 chests CT images
[181]	5L-DCNN-SP-C	Accuracy: 93.64%, Sensitivity: 93.28%, Specificity: 94.00%, Precision: 93.96%, F1 score:93.62%	A Private Dataset: 320 chest CT images
[182]	AlexNet	Accuracy: 86.85%, Sensitivity: 80.25%, Specificity: 94.29%, F1 score: 0.85, AUC: 0.94	COVID-CT-Dataset: 349 chest CT images
[182]	GoogleNet	Accuracy: 93.83%, Sensitivity: 96.71%, Specificity: 90.57%, F1 score: 0.94, AUC: 0.96	
[182]	ResNet-18	Accuracy: 95.44%, Sensitivity: 98.99%, Specificity: 91.43%, F1 score: 0.96, AUC: 0.98	
[182]	ResNet-50	Accuracy: 93.62%, Sensitivity: 95.57%, Specificity: 91.43%, F1 score: 0.94, AUC: 0.98	
[182]	ResNet-101	Accuracy: 93.29%, Sensitivity: 96.20%, Specificity: 90.00%, F1 score: 0.94, AUC: 0.98	
[182]	Inception-ResNet-v2	Accuracy: 88.59%, Sensitivity: 89.24%, Specificity: 87.86%, F1 score: 0.89, AUC: 0.96	
[182]	VGG-16	Accuracy: 89.26%, Sensitivity: 92.83%, Specificity: 85.24%, F1 score: 0.90, AUC: 0.96	
[182]	VGG-19	Accuracy: 90.16%, Sensitivity: 87.34%, Specificity: 93.33%, F1 score: 0.90, AUC: 0.97	
[182]	DenseNet-201	Accuracy: 96.20%, Sensitivity: 95.78%, Specificity: 96.67%, F1 score: 0.96, AUC: 0.98	
[183]	WCNN4	Accuracy: 99.03%	COVID-19 CT Dataset: 19685 chest CT images

As can be seen from Table 2, there have been many suggestions, proposals, and implementations for applying CNN to COVID-19 diagnosis by analyzing chest CT images. For example, some researchers focus on adjusting the architecture of the network. Some focus on adjusting the number of layers of the network, some on improving existing algorithms, and some on mixing several structures.

Many schemes listed above have achieved accuracy higher than 90% when implemented. However, there are a lot of CNN networks that have a huge space for improvement:

(1) It can be concluded from the results that the modified AlexNet can obtain better accuracy. When combined with other structures, the network is improved so that the accuracy is higher for COVID-19 diagnosis.
(2) Different accuracy will be obtained from different depths of ResNet. Common ones include ResNet18, ResNet50, ResNet101, etc. However, according to the results of the above experiments, ResNet18 performs the best. ResNet can be modified to improve accuracy. The modified ResNet or ResNet combined with other networks could achieve more than 90% accuracy. ResNet50 is often used with other architectures or algorithms for COVID-19 diagnosis. Moreover, we found that the larger the dataset ResNet applied, the higher the accuracy.
(3) With DenseNet, the accuracy was close to 95%. DenseNet has a better performance than other networks on the same dataset. The dataset size does not affect DenseNet accuracy as much as the depth of the network. DenseNet201 shows excellent performance for COVID-19 diagnosis.
(4) Even on small datasets, VGG combined with other structures or algorithms can achieve more than 95% accuracy. VGG19 and VGG16 performed similarly.
(5) The accuracy of GoogleNet for COVID-19 diagnosis is not good enough, generally less than 90%. Try tweaking the architecture of GoogleNet or combining GoogleNet with other networks to improve classification accuracy.
(6) For almost all networks, accuracy increases as the dataset become larger. Private datasets are generally smaller than public datasets. When there are more than 1000 images in the dataset, almost all networks or models can achieve more than 95% accuracy.

6.2. CNN Applied to X-ray Images

The papers mentioned in Table 3 use the architecture or technology of convolutional neural networks (CNNs) to analyze chest X-ray images and establish various training models for COVID-19 diagnosis. Through optimizations, these automatic diagnosis systems have achieved better performance. Chest X-ray images were more readily available. So COVID-19 diagnosis systems could be more widely used by analyzing chest X-ray images in areas with inadequate medical systems.

Cortés et al. [184] applied to learn transfer to AlexNet and fine-tuned it. The first layer of AlexNet is replaced for images in a single intensity. Kaur et al. [185] used the improved AlexNet architecture. Strength Pareto evolutionary algorithm-II (SPEA-II) is used to optimize parameters. Narin et al. [186] compared different layers of ResNet to classify chest X-ray images. ResNet50 provides the highest classification performance. In the study of Chowdhury et al. [187], DenseNet201 performed well in classifying chest X-ray images with image augmentation. In the study, Hernandez et al. applied transfer learning through ResNet, DenseNet, and VGG and fine-tuned them, which achieved higher accuracy [188]. Sitaula et al. [189] added the attention module to the appropriate convolution layer of VGG16. Classification experiments were performed on three COVID-19 chest X-ray image datasets. Haritha et al. [190] used GoogleNet to classify X-ray images and predict COVID-19. Kaya et al. [191] first applied the angle transformation (AT) on X-ray images. Then these images are trained using GoogleNet combined with LSTM. Finally, they obtained a better accuracy rate.

Table 3. Performances of CNN in COVID-19 Diagnosis using X-ray images.

Reference	Method	Performances	Datasets
[184]	AlexNet	Accuracy: 96.5%, Sensitivity: 98.0%, Specificity: 91.7%	Six Public Databases: 11,312 chest X-ray images
[185]	mAlexNet + SPEA-II	Accuracy: 99.130%, Sensitivity: 99.476%, Specificity: 99.154%	Dataset 1: Kaggle Dataset, 3050 chest X-ray images Dataset 2: 1203 chest X-ray scans
[186]	ResNet50	Accuracy: 99.7%, Specificity: 99.8%, Precision: 98.3%, F1 score: 98.5%, Recall: 98.8%	Dataset 1: GitHub Dataset, 341 chest X-ray images Dataset 2: ChestX-ray8 Database, 2800 chest X-ray images Dataset 3: Kaggle Dataset, 4265 chest X-ray images
[186]	ResNet101	Accuracy: 94.7%, Specificity: 99.9%, Precision: 98.9%, F1 score: 68.6%, Recall: 52.5%	
[186]	ResNet152	Accuracy: 92.8%, Specificity: 98.0%, Precision: 75.7%, F1 score: 60.9%, Recall: 51.0%	
[187]	DenseNet201	Accuracy: 97.94%, Sensitivity: 97.94%, Specificity: 98.80%, Precision: 97.95%, F1 score: 97.94%	Dataset 1: COVID-19 Database, 423 chest X-ray images Dataset 2: 1579 normal chest X-ray images Dataset 3: 1485 viral pneumonia chest X-ray images
[188]	ResNet50 + fine tuning	Accuracy: 90.63%, Precision: 90.00%, F1 score: 90.72%, Recall: 91.67%	Italian Society of Medical and Interventional Radiology and ChexPert Dataset, 27000 chest X-ray images
[188]	DenseNet121 + fine tuning	Accuracy: 83.4%, Precision: 89%, F1 score: 76.19%, Recall: 67%	
[188]	VGG16 + fine tuning	Accuracy: 82.29%, Precision: 80.39%, F1 score: 82.82%, Recall: 85.41%	
[189]	VGG-16 + attention module + convolution module	Accuracy: 79.58% (Dataset 1), 85.43% (Dataset 2), 87.49% (Dataset 3)	Dataset 1: Public Databases, 1125 chest X-ray images Dataset 2: Public Databases, 1638 chest X-ray images Dataset 3: Public Databases, 2138 chest X-ray images
[190]	GoogleNet	Training Accuracy: 99%, Testing Accuracy: 98.5%	Public Dataset: 1824 chest X-ray images
[191]	AT + GoogleNet + LSTM	Accuracy: 98.97%	Mendeley Database: 1824 chest X-ray images

As can be seen from Table 3, there have also been many suggestions, proposals, and implementations for applying CNN to COVID-19 diagnosis by analyzing chest X-ray images. The researchers focused on tweaking the number of layers in the network and

mixing CNNS with other network structures. They divided datasets into several classes for training and validation. Many automatic diagnosis systems can achieve accuracy higher than 90%. However, there is a huge space for improvement.

7. Conclusions

In this paper, we reviewed and summarized convolutional neural networks in COVID-19 diagnosis. We introduced various technologies related to CNN and some mature CNN networks with excellent performance. Then, we analyzed and compared various suggestions of other researchers on the application of CNN for COVID-19 diagnosis. Here are a few conclusions and suggestions:

(1) At present, rapid and accurate COVID-19 diagnosis is vital. The classification method of chest CT or chest X-ray images based on CNN plays an important role.
(2) The current experiment has limited datasets. It is necessary to collect more data or explore better methods for small datasets.
(3) Most experiments do not consider the execution time problem. It is necessary to shorten the execution time with appropriate data preprocessing [192–195] strategies or GPU acceleration.
(4) The experiments discussed in this paper use chest CT or chest X-ray images as the input datasets of CNN and have achieved good performance. Although X-ray image is not as good as CT in performance, it has low cost, low radiation dose, and easy-to-operate in general hospitals [196]. Future research could consider more types of medical image forms. The application of the CNN method on medical images has potential value for COVID-19 diagnosis.

Author Contributions: X.H.: Conceptualization, Software, Investigation, Data Curation, Writing—Original Draft, Visualization. Z.H.: Methodology. Software, Formal analysis, Investigation, Writing—Review and Editing, Visualization, Project administration. S.W.: Methodology, Validation, Formal analysis, Resources, Writing—Review and Editing, Supervision, Project administration, Funding acquisition. Y.Z.: Conceptualization. Software, Validation, Resources, Writing—Original Draft, Writing—Review and Editing, Supervision, Project administration, Funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: The paper is partially supported by British Heart Foundation Accelerator Award, UK (AA/18/3/34220); Royal Society International Exchanges Cost Share Award, UK (RP202G0230); Hope Foundation for Cancer Research, UK (RM60G0680); Medical Research Council Confidence in Concept Award, UK (MC_PC_17171); Sino-UK Industrial Fund, UK (RP202G0289); Global Challenges Research Fund (GCRF), UK (P202PF11); LIAS Pioneering Partnerships award, UK (P202ED10); Data Science Enhancement Fund, UK (P202RE237); Fight for Sight, UK (24NN201).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Manigandan, S.; Wu, M.-T.; Ponnusamy, V.K.; Raghavendra, V.B.; Pugazhendhi, A.; Brindhadevi, K. A systematic review on recent trends in transmission, diagnosis, prevention and imaging features of COVID-19. *Process Biochem.* **2020**, *98*, 233–240. [CrossRef] [PubMed]
2. Guan, W.-J.; Ni, Z.-Y.; Hu, Y.; Liang, W.-H.; Ou, C.-Q.; He, J.-X.; Liu, L.; Shan, H.; Lei, C.-L.; Hui, D.S. Clinical characteristics of coronavirus disease 2019 in China. *New Engl. J. Med.* **2020**, *382*, 1708–1720. [CrossRef] [PubMed]
3. Nour, M.; Cömert, Z.; Polat, K. A novel medical diagnosis model for COVID-19 infection detection based on deep features and Bayesian optimization. *Appl. Soft Comput.* **2020**, *97*, 106580. [CrossRef] [PubMed]
4. Cui, F.; Zhou, H.S. Diagnostic methods and potential portable biosensors for coronavirus disease 2019. *Biosens. Bioelectron.* **2020**, *165*, 112349. [CrossRef] [PubMed]

5. Zhu, X.; Wang, X.; Han, L.; Chen, T.; Wang, L.; Li, H.; Li, S.; He, L.; Fu, X.; Chen, S. Multiplex reverse transcription loop-mediated isothermal amplification combined with nanoparticle-based lateral flow biosensor for the diagnosis of COVID-19. *Biosens. Bioelectron.* **2020**, *166*, 112437. [CrossRef]
6. Qiu, G.; Gai, Z.; Tao, Y.; Schmitt, J.; Kullak-Ublick, G.A.; Wang, J. Dual-functional plasmonic photothermal biosensors for highly accurate severe acute respiratory syndrome coronavirus 2 detection. *ACS Nano* **2020**, *14*, 5268–5277. [CrossRef]
7. Sengupta, J.; Hussain, C.M. Graphene-based field-effect transistor biosensors for the rapid detection and analysis of viruses: A perspective in view of COVID-19. *Carbon Trends* **2021**, *2*, 100011. [CrossRef]
8. Caruso, D.; Zerunian, M.; Polici, M.; Pucciarelli, F.; Polidori, T.; Rucci, C.; Guido, G.; Bracci, B.; De Dominicis, C.; Laghi, A. Chest CT features of COVID-19 in Rome, Italy. *Radiology* **2020**, *296*, 201237. [CrossRef]
9. Xie, X.; Zhong, Z.; Zhao, W.; Zheng, C.; Wang, F.; Liu, J. Chest CT for typical 2019-nCoV pneumonia: Relationship to negative RT-PCR testing. *Radiology* **2020**, *296*, 200343. [CrossRef]
10. Chan, J.F.-W.; Yuan, S.; Kok, K.-H.; To, K.K.-W.; Chu, H.; Yang, J.; Xing, F.; Liu, J.; Yip, C.C.-Y.; Poon, R.W.-S. A familial cluster of pneumonia associated with the 2019 novel coronavirus indicating person-to-person transmission: A study of a family cluster. *Lancet* **2020**, *395*, 514–523. [CrossRef]
11. Hussain, L.; Nguyen, T.; Li, H.; Abbasi, A.A.; Lone, K.J.; Zhao, Z.; Zaib, M.; Chen, A.; Duong, T.Q. Machine-learning classification of texture features of portable chest X-ray accurately classifies COVID-19 lung infection. *BioMedical Eng. OnLine* **2020**, *19*, 88. [CrossRef]
12. Górriz, J.M.; Ramírez, J.; Ortíz, A.; Martínez-Murcia, F.J.; Segovia, F.; Suckling, J.; Leming, M.; Zhang, Y.-D.; Álvarez-Sánchez, J.R.; Bologna, G. Artificial intelligence within the interplay between natural and artificial computation: Advances in data science, trends and applications. *Neurocomputing* **2020**, *410*, 237–270. [CrossRef]
13. Soomro, T.A.; Zheng, L.; Afifi, A.J.; Ali, A.; Yin, M.; Gao, J. Artificial intelligence (AI) for medical imaging to combat coronavirus disease (COVID-19): A detailed review with direction for future research. *Artif. Intell. Rev.* **2022**, *55*, 1409–1439. [CrossRef]
14. Yang, L.; Li, Z.; Ma, S.; Yang, X. Artificial intelligence image recognition based on 5G deep learning edge algorithm of Digestive endoscopy on medical construction. *Alex. Eng. J.* **2022**, *61*, 1852–1863. [CrossRef]
15. Gorriz, J.M.; Ramirez, J.; Suckling, J.; Martínez-Murcia, F.J.; Illán, I.; Segovia, F.; Ortiz, A.; Salas-Gonzalez, D.; Castillo-Barnes, D.; Puntonet, C.G. A semi-supervised learning approach for model selection based on class-hypothesis testing. *Expert Syst. Appl.* **2017**, *90*, 40–49. [CrossRef]
16. Pesapane, F.; Codari, M.; Sardanelli, F. Artificial intelligence in medical imaging: Threat or opportunity? Radiologists again at the forefront of innovation in medicine. *Eur. Radiol. Exp.* **2018**, *2*, 1–10. [CrossRef]
17. Górriz, J.M.; Ramírez, J.; Segovia, F.; Martínez, F.J.; Lai, M.-C.; Lombardo, M.V.; Baron-Cohen, S.; Consortium, M.A.; Suckling, J. A machine learning approach to reveal the neurophenotypes of autisms. *Int. J. Neural Syst.* **2019**, *29*, 1850058. [CrossRef]
18. Satapathy, S.; Loganathan, D.; Kondaveeti, H.K.; Rath, R. Performance analysis of machine learning algorithms on automated sleep staging feature sets. *CAAI Trans. Intell. Technol.* **2021**, *6*, 155–174. [CrossRef]
19. Lee, J.-G.; Jun, S.; Cho, Y.-W.; Lee, H.; Kim, G.B.; Seo, J.B.; Kim, N. Deep learning in medical imaging: General overview. *Korean J. Radiol.* **2017**, *18*, 570–584. [CrossRef]
20. Varoquaux, G.; Cheplygina, V. Machine learning for medical imaging: Methodological failures and recommendations for the future. *NPJ Digit. Med.* **2022**, *5*, 1–8. [CrossRef]
21. Elyan, E.; Vuttipittayamongkol, P.; Johnston, P.; Martin, K.; McPherson, K.; Moreno-García, C.F.; Jayne, C.; Sarker, M.K. Computer vision and machine learning for medical image analysis: Recent advances, challenges, and way forward. *Artif. Intell. Surg.* **2022**, *2*, 24–25. [CrossRef]
22. Gasparin, A.; Lukovic, S.; Alippi, C. Deep learning for time series forecasting: The electric load case. *CAAI Trans. Intell. Technol.* **2022**, *7*, 1–25. [CrossRef]
23. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
24. Mukherjee, S.; Sadhukhan, B.; Sarkar, N.; Roy, D.; De, S. Stock market prediction using deep learning algorithms. *CAAI Trans. Intell. Technol.* **2021**. *early view*. [CrossRef]
25. Serte, S.; Serener, A.; Al-Turjman, F. Deep learning in medical imaging: A brief review. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e4080. [CrossRef]
26. Tsuneki, M. Deep learning models in medical image analysis. *J. Oral Biosci.* **2022**, *64*, 312–320. [CrossRef]
27. Liu, X.; Yang, L.; Chen, J.; Yu, S.; Li, K. Region-to-boundary deep learning model with multi-scale feature fusion for medical image segmentation. *Biomed. Signal Process. Control* **2022**, *71*, 103165. [CrossRef]
28. Ning, F.; Delhomme, D.; LeCun, Y.; Piano, F.; Bottou, L.; Barbano, P.E. Toward automatic phenotyping of developing embryos from videos. *IEEE Trans. Image Process.* **2005**, *14*, 1360–1371. [CrossRef]
29. CireAan, D.; Meier, U.; Masci, J.; Schmidhuber, J. Multi-column deep neural network for traffic sign classification. *Neural Netw.* **2012**, *32*, 333–338. [CrossRef]
30. Abdel-Salam, R.; Mostafa, R.; Abdel-Gawad, A.H. RIECNN: Real-time image enhanced CNN for traffic sign recognition. *Neural Comput. Appl.* **2022**, *34*, 6085–6096. [CrossRef]
31. Karthika, R.; Parameswaran, L. A novel convolutional neural network based architecture for object detection and recognition with an application to traffic sign recognition from road scenes. *Pattern Recognit. Image Anal.* **2022**, *32*, 351–362. [CrossRef]

32. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv* **2014**, arXiv:1412.7062.
33. Du, H.; Shi, H.; Zeng, D.; Zhang, X.-P.; Mei, T. The elements of end-to-end deep face recognition: A survey of recent advances. *ACM Comput. Surv. (CSUR)* **2022**, *54*, 1–42. [CrossRef]
34. Large Scale Visual Recognition Challenge 2016 (ILSVRC2016) Results. Available online: http://image-net.org/challenges/LSVRC/2016/ (accessed on 1 April 2022).
35. Dhiman, G.; Juneja, S.; Viriyasitavat, W.; Mohafez, H.; Hadizadeh, M.; Islam, M.A.; El Bayoumy, I.; Gulati, K. A novel machine-learning-based hybrid CNN model for tumor identification in medical image processing. *Sustainability* **2022**, *14*, 1447. [CrossRef]
36. Tiwari, P.; Pant, B.; Elarabawy, M.M.; Abd-Elnaby, M.; Mohd, N.; Dhiman, G.; Sharma, S. Cnn based multiclass brain tumor detection using medical imaging. *Comput. Intell. Neurosci.* **2022**, *2022*, 1830010. [CrossRef]
37. Chen, S.; Gamechi, Z.S.; Dubost, F.; van Tulder, G.; de Bruijne, M. An end-to-end approach to segmentation in medical images with CNN and posterior-CRF. *Med. Image Anal.* **2022**, *76*, 102311. [CrossRef]
38. Aslan, M.F.; Sabanci, K.; Durdu, A.; Unlersen, M.F. COVID-19 diagnosis using state-of-the-art CNN architecture features and Bayesian Optimization. *Comput. Biol. Med.* **2022**, *142*, 105244. [CrossRef]
39. De Sousa, P.M.; Carneiro, P.C.; Oliveira, M.M.; Pereira, G.M.; da Costa Junior, C.A.; de Moura, L.V.; Mattjie, C.; da Silva, A.M.M.; Patrocinio, A.C. COVID-19 classification in X-ray chest images using a new convolutional neural network: CNN-COVID. *Res. Biomed. Eng.* **2022**, *38*, 87–97. [CrossRef]
40. Fan, X.; Feng, X.; Dong, Y.; Hou, H. COVID-19 CT image recognition algorithm based on transformer and CNN. *Displays* **2022**, *72*, 102150. [CrossRef]
41. Hounsfield, G.N. Computerized transverse axial scanning (tomography): Part 1. Description of system. *Br. J. Radiol.* **1973**, *46*, 1016–1022. [CrossRef]
42. Cellina, M.; Orsi, M.; Valenti Pittino, C.; Toluian, T.; Oliva, G. Chest computed tomography findings of COVID-19 pneumonia: Pictorial essay with literature review. *Jpn. J. Radiol.* **2020**, *38*, 1012–1019. [CrossRef] [PubMed]
43. Wang, J.; Xu, Z.; Feng, R.; An, Y.; Ao, W.; Gao, Y.; Wang, X.; Xie, Z. CT characteristics of patients infected with 2019 novel coronavirus: Association with clinical type. *Clin. Radiol.* **2020**, *75*, 408–414. [CrossRef] [PubMed]
44. Bernheim, A.; Mei, X.; Huang, M.; Yang, Y.; Fayad, Z.A.; Zhang, N.; Diao, K.; Lin, B.; Zhu, X.; Li, K. Chest CT findings in coronavirus disease-19 (COVID-19): Relationship to duration of infection. *Radiology* **2020**, *295*, 200463. [CrossRef] [PubMed]
45. Wong, H.Y.F.; Lam, H.Y.S.; Fong, A.H.-T.; Leung, S.T.; Chin, T.W.-Y.; Lo, C.S.Y.; Lui, M.M.-S.; Lee, J.C.Y.; Chiu, K.W.-H.; Chung, T.W.-H. Frequency and distribution of chest radiographic findings in patients positive for COVID-19. *Radiology* **2020**, *296*, E72–E78. [CrossRef] [PubMed]
46. Jokerst, C.; Chung, J.H.; Ackman, J.B.; Carter, B.; Colletti, P.M.; Crabtree, T.D.; de Groot, P.M.; Iannettoni, M.D.; Maldonado, F.; McComb, B.L. ACR Appropriateness Criteria® acute respiratory illness in immunocompetent patients. *J. Am. Coll. Radiol.* **2018**, *15*, S240–S251. [CrossRef]
47. Ball, L.; Vercesi, V.; Costantino, F.; Chandrapatham, K.; Pelosi, P. Lung imaging: How to get better look inside the lung. *Ann. Transl. Med.* **2017**, *5*, 294. [CrossRef]
48. Dennie, C.; Hague, C.; Lim, R.S.; Manos, D.; Memauri, B.F.; Nguyen, E.T.; Taylor, J. Canadian Society of Thoracic Radiology/Canadian Association of Radiologists consensus statement regarding chest imaging in suspected and confirmed COVID-19. *Can. Assoc. Radiol. J.* **2020**, *71*, 470–481. [CrossRef]
49. Oh, Y.; Park, S.; Ye, J.C. Deep learning COVID-19 features on CXR using limited training data sets. *IEEE Trans. Med. Imaging* **2020**, *39*, 2688–2700. [CrossRef]
50. Loraksa, C.; Mongkolsomlit, S.; Nimsuk, N.; Uscharapong, M.; Kiatisevi, P. Effectiveness of Learning Systems from Common Image File Types to Detect Osteosarcoma Based on Convolutional Neural Networks (CNNs) Models. *J. Imaging* **2021**, *8*, 2. [CrossRef]
51. Jin, K.H.; McCann, M.T.; Froustey, E.; Unser, M. Deep Convolutional Neural Network for Inverse Problems in Imaging. *IEEE Trans. Image Process.* **2017**, *26*, 4509–4522. [CrossRef]
52. Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A.P.; Bishop, R.; Rueckert, D.; Wang, Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1874–1883.
53. Zhao, Y.; Wang, L. The application of convolution neural networks in sign language recognition. In Proceedings of the 2018 Ninth International Conference on Intelligent Control and Information Processing (ICICIP), Wanzhou, China, 9–11 November 2018; pp. 269–272.
54. Rastegari, M.; Ordonez, V.; Redmon, J.; Farhadi, A. Xnor-net: Imagenet classification using binary convolutional neural networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 525–542.
55. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent advances in convolutional neural networks. *Pattern Recognit.* **2018**, *77*, 354–377. [CrossRef]
56. Jiang, X.; Satapathy, S.C.; Yang, L.; Wang, S.-H.; Zhang, Y.-D. A Survey on Artificial Intelligence in Chinese Sign Language Recognition. *Arab. J. Sci. Eng.* **2020**, *45*, 9859–9894. [CrossRef]
57. Wang, T.; Wu, D.J.; Coates, A.; Ng, A.Y. End-to-end text recognition with convolutional neural networks. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan, 11–15 November 2012; pp. 3304–3308.

58. Boureau, Y.-L.; Ponce, J.; LeCun, Y. A theoretical analysis of feature pooling in visual recognition. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 111–118.
59. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision, Zürich, Switzerland, 5–12 September 2014; pp. 818–833.
60. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, arXiv:1207.0580.
61. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
62. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
63. LeCun, Y.A.; Bottou, L.; Orr, G.B.; Müller, K.-R. Efficient backprop. In *Neural Networks: Tricks of the Trade*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 9–48.
64. Wiesler, S.; Ney, H. A convergence analysis of log-linear training. In Proceedings of the Advances in Neural Information Processing Systems 24 (NIPS 2011), Granada, Spain, 12–15 December 2011; Volume 24.
65. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 929–1958.
66. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
67. Gao, H.; Cai, L.; Ji, S. Adaptive convolutional relus. In Proceedings of the Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 3914–3921.
68. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the International Conference on Machine Learning (ICML 2013), Atlanta, GA, USA, 16–21 June 2013; p. 3.
69. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
70. Xu, B.; Wang, N.; Chen, T.; Li, M. Empirical evaluation of rectified activations in convolutional network. *arXiv* **2015**, arXiv:1505.00853.
71. Clevert, D.-A.; Unterthiner, T.; Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv* **2015**, arXiv:1511.07289.
72. Wu, H.; Gu, X. Towards dropout training for convolutional neural networks. *Neural Netw.* **2015**, *71*, 1–10. [CrossRef]
73. Cai, H.; Gan, C.; Wang, T.; Zhang, Z.; Han, S. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv* **2019**, arXiv:1908.09791.
74. Yildirim, O.; Baloglu, U.B.; Tan, R.-S.; Ciaccio, E.J.; Acharya, U.R. A new approach for arrhythmia classification using deep coded features and LSTM networks. *Comput. Methods Programs Biomed.* **2019**, *176*, 121–133. [CrossRef] [PubMed]
75. Zhao, R.; Song, W.; Zhang, W.; Xing, T.; Lin, J.-H.; Srivastava, M.; Gupta, R.; Zhang, Z. Accelerating binarized convolutional neural networks with software-programmable FPGAs. In Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, Monterey, CA, USA, 22–24 February 2017; pp. 15–24.
76. Murray, N.; Perronnin, F. Generalized max pooling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2473–2480.
77. He, Z.; Shao, H.; Zhong, X.; Zhao, X. Ensemble transfer CNNs driven by multi-channel signals for fault diagnosis of rotating machinery cross working conditions. *Knowl.-Based Syst.* **2020**, *207*, 106396. [CrossRef]
78. Singh, P.; Chaudhury, S.; Panigrahi, B.K. Hybrid MPSO-CNN: Multi-level particle swarm optimized hyperparameters of convolutional neural network. *Swarm Evol. Comput.* **2021**, *63*, 100863. [CrossRef]
79. Zafar, A.; Aamir, M.; Mohd Nawi, N.; Arshad, A.; Riaz, S.; Alruban, A.; Dutta, A.K.; Almotairi, S. A Comparison of Pooling Methods for Convolutional Neural Networks. *Appl. Sci.* **2022**, *12*, 8643. [CrossRef]
80. Li, Y.; Bao, J.; Chen, T.; Yu, A.; Yang, R. Prediction of ball milling performance by a convolutional neural network model and transfer learning. *Powder Technol.* **2022**, *403*, 117409. [CrossRef]
81. Zhang, W.; Li, C.; Peng, G.; Chen, Y.; Zhang, Z. A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. *Mech. Syst. Signal Process.* **2018**, *100*, 439–453. [CrossRef]
82. Wang, S.-H.; Attique, K.M.; Vishnuvarthanan, G. Deep rank-based average pooling network for COVID-19 recognition. *Comput. Mater. Contin.* **2022**, *70*, 2797–2813. [CrossRef]
83. Graham, B. Fractional max-pooling. *arXiv* **2014**, arXiv:1412.6071.
84. Wang, S.-H.; Satapathy, S.C.; Anderson, D.; Chen, S.-X.; Zhang, Y.-D. Deep fractional max pooling neural network for COVID-19 recognition. *Front. Public Health* **2021**, *9*, 726144. [CrossRef]
85. Yu, D.; Wang, H.; Chen, P.; Wei, Z. Mixed pooling for convolutional neural networks. In Proceedings of the International Conference on Rough Sets and Knowledge Technology, Shanghai, China, 24–26 October 2014; pp. 364–375.
86. Zhou, Q.; Qu, Z.; Cao, C. Mixed pooling and richer attention feature fusion for crack detection. *Pattern Recognit. Lett.* **2021**, *145*, 96–102. [CrossRef]
87. Nayak, D.R.; Dash, R.; Majhi, B. Automated Diagnosis of Multi-class Brain Abnormalities using MRI Images: A Deep Convolutional Neural Network based Method. *Pattern Recognit. Lett.* **2020**, *138*, 385–391. [CrossRef]
88. Deliège, A.; Istasse, M.; Kumar, A.; De Vleeschouwer, C.; Van Droogenbroeck, M. Ordinal pooling. *arXiv* **2021**, arXiv:2109.01561.

89. Skourt, B.A.; El Hassani, A.; Majda, A. Mixed-pooling-dropout for convolutional neural network regularization. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 4756–4762. [CrossRef]
90. Lee, C.-Y.; Gallagher, P.W.; Tu, Z. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In Proceedings of the Artificial Intelligence and Statistics, Cadiz, Spain, 9–11 May 2016; pp. 464–472.
91. Bello, M.; Nápoles, G.; Sánchez, R.; Bello, R.; Vanhoof, K. Deep neural network to extract high-level features and labels in multi-label classification problems. *Neurocomputing* **2020**, *413*, 259–270. [CrossRef]
92. Blonder, B.; Both, S.; Jodra, M.; Xu, H.; Fricker, M.; Matos, I.S.; Majalap, N.; Burslem, D.F.; Teh, Y.A.; Malhi, Y. Linking functional traits to multiscale statistics of leaf venation networks. *New Phytol.* **2020**, *228*, 1796–1810. [CrossRef]
93. Xu, Q.; Zhang, M.; Gu, Z.; Pan, G. Overfitting remedy by sparsifying regularization on fully-connected layers of CNNs. *Neurocomputing* **2019**, *328*, 69–74. [CrossRef]
94. Chen, Y.; Ming, D.; Lv, X. Superpixel based land cover classification of VHR satellite image combining multi-scale CNN and scale parameter estimation. *Earth Sci. Inform.* **2019**, *12*, 341–363. [CrossRef]
95. Wang, S.-H.; Zhang, Y.; Cheng, X.; Zhang, X.; Zhang, Y.-D. PSSPNN: PatchShuffle stochastic pooling neural network for an explainable diagnosis of COVID-19 with multiple-way data augmentation. *Comput. Math. Methods Med.* **2021**, *2021*, 6633755. [CrossRef]
96. Zhang, Y.-D.; Satapathy, S.C.; Zhu, L.Y.; Górriz, J.M.; Wang, S. A seven-layer convolutional neural network for chest CT-based COVID-19 diagnosis using stochastic pooling. *IEEE Sens. J.* **2022**, *22*, 17573–17582. [CrossRef]
97. Özdemir, Ö.; Sönmez, E.B. Attention mechanism and mixup data augmentation for classification of COVID-19 Computed Tomography images. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 6199–6207. [CrossRef]
98. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
99. Yan, J.; Feng, K.; Zhao, H.; Sheng, K. Siamese-Prototypical Network with Data Augmentation Pre-training for Few-shot Medical Image Classification. In Proceedings of the 2022 2nd International Conference on Frontiers of Electronics, Information and Computation Technologies (ICFEICT), Wuhan, China, 19–21 August 2022; pp. 387–391.
100. Monshi, M.M.A.; Poon, J.; Chung, V.; Monshi, F.M. CovidXrayNet: Optimizing data augmentation and CNN hyperparameters for improved COVID-19 detection from CXR. *Comput. Biol. Med.* **2021**, *133*, 104375. [CrossRef] [PubMed]
101. Barshooi, A.H.; Amirkhani, A. A novel data augmentation based on Gabor filter and convolutional deep learning for improving the classification of COVID-19 chest X-Ray images. *Biomed. Signal Process. Control* **2022**, *72*, 103326. [CrossRef] [PubMed]
102. Banerjee, S.; Chipman, R.; Otani, Y. Simultaneous balancing of geometric transformation and linear polarizations using six-fold-mirror geometry over the visible region. *Opt. Lett.* **2020**, *45*, 2510–2513. [CrossRef] [PubMed]
103. D'Cunha, N.W.; Birajdhar, S.A.; Manikantan, K.; Ramachandran, S. Face recognition using Homomorphic Filtering as a pre-processing technique. In Proceedings of the 2013 International Conference on Emerging Trends in Communication, Control, Signal Processing and Computing Applications (C2SPCA), Bangalore, India, 10–11 October 2013; pp. 1–6.
104. Morita, M.; Fujii, Y.; Sato, T. The Width Underestimation of 3D Objects with Image Rotation. In *I-Perception, Proceedings of the 15th Asia-Pacific Conferenceon Vision (APCV), Osaka, Japan, 29 July–1 August 2019*; Sage Publications Ltd.: London, UK; p. 43.
105. Wang, S.; Celebi, M.E.; Zhang, Y.-D.; Yu, X.; Lu, S.; Yao, X.; Zhou, Q.; Miguel, M.-G.; Tian, Y.; Gorriz, J.M.; et al. Advances in Data Preprocessing for Biomedical Data Fusion: An Overview of the Methods, Challenges, and Prospects. *Inf. Fusion* **2021**, *76*, 376–421. [CrossRef]
106. Gawedzinski, J.; Schmeler, K.M.; Milbourne, A.; Ramalingam, P.; Moghaddam, P.A.; Richards-Kortum, R.; Tkaczyk, T.S. Toward development of a large field-of-view cancer screening patch (CASP) to detect cervical intraepithelial neoplasia. *Biomed. Opt. Express* **2019**, *10*, 6145–6159. [CrossRef]
107. Leyh-Bannurah, S.-R.; Wolffgang, U.; Schmitz, J.; Ouellet, V.; Azzi, F.; Tian, Z.; Helmke, B.; Graefen, M.; Budäus, L.; Karakiewicz, P.I. MP19-20 State-of-the-Art Weakly Supervised Automated Classification of Prostate Cancer Tissue Microarrays via Deep Learning: Can Sufficient Accuracy Be Achieved without Manual Patch Level Annotation? *J. Urol.* **2020**, *203*, e306. [CrossRef]
108. Pandian, J.A.; Geetharamani, G.; Annette, B. Data augmentation on plant leaf disease image dataset using image manipulation and deep learning techniques. In Proceedings of the 2019 IEEE 9th International Conference on Advanced Computing (IACC), Tiruchirappalli, India, 13–14 December 2019; pp. 199–204.
109. Tada, Y.; Hagiwara, Y.; Tanaka, H.; Taniguchi, T. Robust understanding of robot-directed speech commands using sequence to sequence with noise injection. *Front. Robot. AI* **2020**, *6*, 144. [CrossRef]
110. Zhang, Y.-D.; Dong, Z.; Chen, X.; Jia, W.; Du, S.; Muhammad, K.; Wang, S.-H. Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimed. Tools Appl.* **2019**, *78*, 3613–3632. [CrossRef]
111. DeVries, T.; Taylor, G.W. Dataset augmentation in feature space. *arXiv* **2017**, arXiv:1702.05538.
112. Xie, L.; Wang, J.; Wei, Z.; Wang, M.; Tian, Q. Disturblabel: Regularizing cnn on the loss layer. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4753–4762.
113. Sebastian, S.; Manimekalai, M. Color image compression Using JPEG2000 with adaptive color space transform. In Proceedings of the 2014 International Conference on Electronics and Communication Systems (ICECS), Coimbatore, India, 13–14 February 2014; pp. 1–5.
114. Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; Yang, Y. Random erasing data augmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 13001–13008.

115. Singhal, P.; Verma, A.; Garg, A. A study in finding effectiveness of Gaussian blur filter over bilateral filter in natural scenes for graph based image segmentation. In Proceedings of the 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 January 2017; pp. 1–6.
116. Blessy, P.S.S.A.; Sulochana, H.C. Enhanced Homomorphic Unsharp Masking method for intensity inhomogeneity correction in brain MR images. *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.* **2020**, *8*, 40–48.
117. Inoue, H. Data augmentation by pairing samples for images classification. *arXiv* **2018**, arXiv:1801.02929.
118. Summers, C.; Dinneen, M.J. Improved mixed-example data augmentation. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 7–11 January 2019; pp. 1262–1270.
119. Takahashi, R.; Matsubara, T.; Uehara, K. Data augmentation using random image cropping and patching for deep CNNs. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 2917–2931. [CrossRef]
120. Bowles, C.; Chen, L.; Guerrero, R.; Bentley, P.; Gunn, R.; Hammers, A.; Dickie, A.D.; Valdés Hernández, M.; Wardlaw, J.; Rueckert, D. GAN augmentation: Augmenting training data using generative adversarial networks. *arXiv* **2018**, arXiv:1810.10863.
121. Gatys, L.A.; Ecker, A.S.; Bethge, M. A neural algorithm of artistic style. *arXiv* **2015**, arXiv:1508.06576. [CrossRef]
122. Zoph, B.; Le, Q.V. Neural architecture search with reinforcement learning. *arXiv* **2016**, arXiv:1611.01578.
123. Perez, L.; Wang, J. The effectiveness of data augmentation in image classification using deep learning. *arXiv* **2017**, arXiv:1712.04621.
124. Lemley, J.; Bazrafkan, S.; Corcoran, P. Smart Augmentation Learning an Optimal Data Augmentation Strategy. *IEEE Access* **2017**, *5*, 5858–5869. [CrossRef]
125. Ekin, D.C.; Barret, Z.; Dandelion, M.; Vijay, V.; Quoc, V.L. AutoAugment: Learning augmentation policies from data. *arXiv* **2019**, arXiv:1805.09501.
126. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
127. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [CrossRef] [PubMed]
128. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
129. Saxe, A.M.; McClelland, J.L.; Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv* **2013**, arXiv:1312.6120.
130. He, K.; Sun, J. Convolutional neural networks at constrained time cost. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5353–5360.
131. Srivastava, R.K.; Greff, K.; Schmidhuber, J. Highway networks. *arXiv* **2015**, arXiv:1505.00387.
132. 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, USA, 27–30 June 2016; pp. 770–778.
133. Huang, G.; Liu, Z.; Laurens, V.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
134. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
135. Weiss, K.; Khoshgoftaar, T.M.; Wang, D. A survey of transfer learning. *J. Big Data* **2016**, *3*, 9. [CrossRef]
136. Wang, C.; Mahadevan, S. Heterogeneous domain adaptation using manifold alignment. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Spain, 16–22 July 2011.
137. Duan, L.; Xu, D.; Tsang, I. Learning with augmented features for heterogeneous domain adaptation. *arXiv* **2012**, arXiv:1206.4660.
138. Kulis, B.; Saenko, K.; Darrell, T. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In Proceedings of the IEEE 2011 Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011.
139. Zhu, Y.; Chen, Y.; Lu, Z.; Pan, S.J.; Xue, G.-R.; Yu, Y.; Yang, Q. Heterogeneous transfer learning for image classification. In Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 7–11 August 2011; Volume 2.
140. Harel, M.; Mannor, S. Learning from multiple outlooks. *arXiv* **2010**, arXiv:1005.0027.
141. Nam, J.; Kim, S. Heterogeneous defect prediction. In Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, Bergamo, Italy, 30 August–4 September 2015; pp. 508–519.
142. Peter, P.; Benno, S. Cross-language text classification using structural correspondence learning. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, 11–16 July 2010; pp. 1118–1127.
143. Zhou, J.; Pan, S.; Tsang, I.; Yan, Y. Hybrid heterogeneous transfer learning through deep learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; pp. 2213–2219.
144. Zhou, J.; Tsang, I.; Pan, S.; Tan, M. Heterogeneous domain adaptation for multiple classes. In Proceedings of the 17th International Conference on Artificial Intelligence and Statistics, Reykjavik, Iceland, 22–25 April 2014; pp. 1095–1103.
145. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [CrossRef]
146. Sarkar, D.; Bali, R.; Ghosh, T. *Hands-On Transfer Learning with Python: Implement Advanced Deep Learning and Neural Network Models Using TensorFlow and Keras*; Packt Publishing Ltd.: Birmingham, UK, 2018.

147. Guo, Y.; Shi, H.; Kumar, A.; Grauman, K.; Rosing, T.; Feris, R. Spottune: Transfer learning through adaptive fine-tuning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15 June–20 June 2019; pp. 4800–4809.
148. Caruana, R. Multitask learning. *Mach. Learn.* **1997**, *28*, 41–75. [CrossRef]
149. Daume III, H.; Marcu, D. Domain adaptation for statistical classifiers. *J. Artif. Intell. Res.* **2006**, *26*, 101–126. [CrossRef]
150. Zadrozny, B. Learning and evaluating classifiers under sample selection bias. In Proceedings of the Twenty-First International Conference on Machine Learning, Banff, AB, Canada, 4–8 July 2004; p. 114.
151. Shimodaira, H. Improving predictive inference under covariate shift by weighting the log-likelihood function. *J. Stat. Plan. Inference* **2000**, *90*, 227–244. [CrossRef]
152. Huang, J. Correcting sample selection bias by unlabeled data. *Adv. Neural Inf. Processing Systems* **2007**, *19*, 601–608.
153. Jakob, N.; Gurevych, I. Extracting opinion targets in a single and cross-domain setting with conditional random fields. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, MA, USA, 9–11 October 2010; pp. 1035–1045.
154. Moreno-Torres, J.G.; Raeder, T.; Alaiz-Rodríguez, R.; Chawla, N.V.; Herrera, F. A unifying view on dataset shift in classification. *Pattern Recognit.* **2012**, *45*, 521–530.
155. Pan, S.J.; Kwok, J.T.; Yang, Q. Transfer learning via dimensionality reduction. In Proceedings of the AAAI, Chicago, IL, USA, 13–17 July 2008; pp. 677–682.
156. Blitzer, J.; McDonald, R.; Pereira, F. Domain adaptation with structural correspondence learning. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Sydney, Australia, 22–23 July 2006; pp. 120–128.
157. Ribani, R.; Marengoni, M. A survey of transfer learning for convolutional neural networks. In Proceedings of the 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T), Rio de Janeiro, Brazil, 28–31 October 2019; pp. 47–57.
158. Gao, J.; Fan, W.; Jiang, J.; Han, J. Knowledge transfer via multiple model local structure mapping. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 283–291.
159. Bonilla, E.V.; Chai, K.; Williams, C. Multi-task Gaussian process prediction. In Proceedings of the 20th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 3–6 December 2007; pp. 153–160.
160. Evgeniou, T.; Pontil, M. Regularized multi-task learning. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004; pp. 109–117.
161. Liang, R.-Z.; Xie, W.; Li, W.; Wang, H.; Wang, J.J.-Y.; Taylor, L. A novel transfer learning method based on common space mapping and weighted domain matching. In Proceedings of the 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), San Jose, CA, USA, 6–8 November 2016; pp. 299–303.
162. Li, F.; Pan, S.J.; Jin, O.; Yang, Q.; Zhu, X. Cross-domain co-extraction of sentiment and topic lexicons. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Jeju Island, Korea, 8–14 July 2012; pp. 410–419.
163. Mihalkova, L.; Mooney, R.J. Transfer learning from minimal target data by mapping across relational domains. In Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, CA, USA, 11–17 July 2009; pp. 1163–1168.
164. Chang, Z.; Zhan, Z.; Zhao, Z.; You, Z.; Liu, Y.; Yan, Z.; Fu, Y.; Liang, W.; Zhao, L. Application of artificial intelligence in COVID-19 medical area: A systematic review. *J. Thorac. Dis.* **2021**, *13*, 7034–7053. [CrossRef]
165. Aslan, M.F.; Sabanci, K.; Ropelewska, E. A New Approach to COVID-19 Detection: An ANN Proposal Optimized through Tree-Seed Algorithm. *Symmetry* **2022**, *14*, 1310. [CrossRef]
166. Aslan, M.F.; Unlersen, M.F.; Sabanci, K.; Durdu, A. CNN-based transfer learning–BiLSTM network: A novel approach for COVID-19 infection detection. *Appl. Soft Comput.* **2021**, *98*, 106912. [CrossRef]
167. Zhang, X.; Lu, S.; Wang, S.-H.; Yu, X.; Wang, S.-J.; Yao, L.; Pan, Y.; Zhang, Y.-D. Diagnosis of COVID-19 pneumonia via a novel deep learning architecture. *J. Comput. Sci. Technol.* **2022**, *37*, 330–343. [CrossRef]
168. Xu, X.; Jiang, X.; Ma, C.; Du, P.; Li, X.; Lv, S.; Yu, L.; Ni, Q.; Chen, Y.; Su, J. A deep learning system to screen novel coronavirus disease 2019 pneumonia. *Engineering* **2020**, *6*, 1122–1129. [CrossRef]
169. Wu, X.; Hui, H.; Niu, M.; Li, L.; Wang, L.; He, B.; Yang, X.; Li, L.; Li, H.; Tian, J. Deep learning-based multi-view fusion model for screening 2019 novel coronavirus pneumonia: A multicentre study. *Eur. J. Radiol.* **2020**, *128*, 109041. [CrossRef]
170. Rahimzadeh, M.; Attar, A.; Sakhaei, S.M. A fully automated deep learning-based network for detecting COVID-19 from a new and large lung CT scan dataset. *Biomed. Signal Process. Control* **2021**, *68*, 102588. [CrossRef]
171. Loey, M.; Manogaran, G.; Khalifa, N.E.M. A deep transfer learning model with classical data augmentation and CGAN to detect COVID-19 from chest CT radiography digital images. *Neural Comput. Appl.* **2020**, *32*, 1–13. [CrossRef]
172. Mondal, M.R.H.; Bharati, S.; Podder, P. CO-IRv2: Optimized InceptionResNetV2 for COVID-19 detection from chest CT images. *PLoS ONE* **2021**, *16*, e0259179. [CrossRef] [PubMed]
173. Hasan, N.; Bao, Y.; Shawon, A.; Huang, Y. DenseNet convolutional neural networks application for predicting COVID-19 using CT image. *SN Comput. Sci.* **2021**, *2*, 389. [CrossRef] [PubMed]
174. Roy, S.; Kiral-Kornek, I.; Harrer, S. ChronoNet: A deep recurrent neural network for abnormal EEG identification. In Proceedings of the 17th Conference on Artificial Intelligence in Medicine, AIME 2019, Poznan, Poland, 26–29 June 2019; pp. 47–56.

175. Xiao, B.; Yang, Z.; Qiu, X.; Xiao, J.; Wang, G.; Zeng, W.; Li, W.; Nian, Y.; Chen, W. PAM-DenseNet: A deep convolutional neural network for computer-aided COVID-19 diagnosis. *IEEE Trans. Cybern.* **2021**, *52*, 12163–12174. [CrossRef] [PubMed]
176. Shah, V.; Keniya, R.; Shridharani, A.; Punjabi, M.; Shah, J.; Mehendale, N. Diagnosis of COVID-19 using CT scan images and deep learning techniques. *Emerg. Radiol.* **2021**, *28*, 497–505. [CrossRef] [PubMed]
177. Tan, W.; Liu, P.; Li, X.; Liu, Y.; Zhou, Q.; Chen, C.; Gong, Z.; Yin, X.; Zhang, Y. Classification of COVID-19 pneumonia from chest CT images based on reconstructed super-resolution images and VGG neural network. *Health Inf. Sci. Syst.* **2021**, *9*, 10. [CrossRef] [PubMed]
178. Wang, S.-H.; Fernandes, S.L.; Zhu, Z.; Zhang, Y.-D. AVNC: Attention-based VGG-style network for COVID-19 diagnosis by CBAM. *IEEE Sens. J.* **2021**, *22*, 17431–17438. [CrossRef]
179. Alsharman, N.; Jawarneh, I. GoogleNet CNN neural network towards chest CT-coronavirus medical image classification. *J. Comput. Sci.* **2020**, *16*, 620–625. [CrossRef]
180. Yu, X.; Wang, S.-H.; Zhang, X.; Zhang, Y.-D. Detection of COVID-19 by GoogLeNet-COD. In *Intelligent Computing Theories and Application: 16th International Conference, ICIC 2020, Bari, Italy, 2–5 October 2020, Proceedings, Part I*; Springer: Cham, Switzerland, 2020; pp. 499–509.
181. Zhang, Y.D.; Satapathy, S.C.; Liu, S.; Li, G.R. A five-layer deep convolutional neural network with stochastic pooling for chest CT-based COVID-19 diagnosis. *Mach. Vis. Appl.* **2021**, *32*, 14. [CrossRef]
182. Pham, T.D. A comprehensive study on classification of COVID-19 on computed tomography with pretrained convolutional neural networks. *Sci. Rep.* **2020**, *10*, 16942. [CrossRef]
183. JavadiMoghaddam, S.; Gholamalinejad, H. A novel deep learning based method for COVID-19 detection from CT image. *Biomed. Signal Process. Control* **2021**, *70*, 102987. [CrossRef]
184. Cortés, E.; Sánchez, S. Deep Learning Transfer with AlexNet for chest X-ray COVID-19 recognition. *IEEE Lat. Am. Trans.* **2021**, *19*, 944–951. [CrossRef]
185. Kaur, M.; Kumar, V.; Yadav, V.; Singh, D.; Kumar, N.; Das, N.N. Metaheuristic-based deep COVID-19 screening model from chest X-ray images. *J. Healthc. Eng.* **2021**, *2021*, 8829829. [CrossRef]
186. Narin, A.; Kaya, C.; Pamuk, Z. Automatic detection of coronavirus disease (COVID-19) using x-ray images and deep convolutional neural networks. *Pattern Anal. Appl.* **2021**, *24*, 1207–1220. [CrossRef]
187. Chowdhury, M.E.; Rahman, T.; Khandakar, A.; Mazhar, R.; Kadir, M.A.; Mahbub, Z.B.; Islam, K.R.; Khan, M.S.; Iqbal, A.; Al Emadi, N. Can AI help in screening viral and COVID-19 pneumonia? *IEEE Access* **2020**, *8*, 132665–132676. [CrossRef]
188. Hernandez, D.; Pereira, R.; Georgevia, P. COVID-19 detection through X-Ray chest images. In Proceedings of the 2020 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 1–3 October 2020; pp. 1–5.
189. Sitaula, C.; Hossain, M.B. Attention-based VGG-16 model for COVID-19 chest X-ray image classification. *Appl. Intell.* **2021**, *51*, 2850–2863. [CrossRef]
190. Haritha, D.; Swaroop, N.; Mounika, M. Prediction of COVID-19 Cases Using CNN with X-rays. In Proceedings of the 2020 5th International Conference on Computing, Communication and Security (ICCCS), Patna, India, 14–16 October 2020; pp. 1–6.
191. Kaya, Y.; Yiner, Z.; Kaya, M.; Kuncan, F. A new approach to COVID-19 detection from X-ray images using angle transformation with GoogleNet and LSTM. *Meas. Sci. Technol.* **2022**, *33*, 124011. [CrossRef]
192. Khanday, N.Y.; Sofi, S.A. Deep insight: Convolutional neural network and its applications for COVID-19 prognosis. *Biomed. Signal Process. Control* **2021**, *69*, 102814. [CrossRef]
193. Abdulah, H.; Huber, B.; Abdallah, H.; Palese, L.L.; Soltanian-Zadeh, H.; Gatti, D.L. A Hybrid Pipeline for COVID-19 Screening Incorporating Lungs Segmentation and Wavelet Based Preprocessing of Chest X-Rays. *medRxiv* **2022**. [CrossRef]
194. Georgiadis, A.; Babbar, V.; Silavong, F.; Moran, S.; Otter, R. ST-FL: Style transfer preprocessing in federated learning for COVID-19 segmentation. In Proceedings of the Medical Imaging 2022: Imaging Informatics for Healthcare, Research, and Applications, SPIE Medical Imaging, San Diego, CA, USA, 4 April 2022; pp. 13–27.
195. Maity, A.; Nair, T.R.; Chandra, A. Image Pre-processing techniques comparison: COVID-19 detection through Chest X-Rays via Deep Learning. *Int. J. Sci. Res. Sci. Technol.* **2020**, *7*, 113–123. [CrossRef]
196. Heidari, M.; Mirniaharikandehei, S.; Khuzani, A.Z.; Danala, G.; Qiu, Y.; Zheng, B. Improving the performance of CNN to predict the likelihood of COVID-19 using chest X-ray images with preprocessing algorithms. *Int. J. Med. Inform.* **2020**, *144*, 104284. [CrossRef]