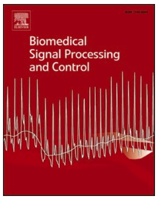
Biomedical Signal Processing and Control 69 (2021) 102814

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc



Deep insight: Convolutional neural network and its applications for COVID-19 prognosis

Nadeem Yousuf Khanday *, Shabir Ahmad Sofi

National Institute of Technology Srinagar, Jammu and Kashmir 19006, India

ARTICLE INFO

Keywords:
Convolutional neural network
Pandemic
COVID-19
Applications
Challenges

ABSTRACT

Background and objective: SARS-CoV-2, a novel strain of coronavirus' also called coronavirus disease 19 (COVID-19), a highly contagious pathogenic respiratory viral infection emerged in December 2019 in Wuhan, a city in China's Hubei province without an obvious cause. Very rapidly it spread across the globe (over 200 countries and territories) and finally on 11 March 2020 World Health Organisation characterized it as a "pandemic". Although it has low mortality of around 3% as of 18 May 2021 it has already infected 164,316,270 humans with 3,406,027 unfortunate deaths. Undoubtedly the world was rocked by the COVID-19 pandemic, but researchers rose to all manner of challenges to tackle this pandemic by adopting the shreds of evidence of ML and AI in previous epidemics to develop novel models, methods, and strategies. We aim to provide a deeper insight into the convolutional neural network which is the most notable and extensively adopted technique on radiographic visual imagery to help expert medical practitioners and researchers to design and finetune their state-of-the-art models for their applicability in the arena of COVID-19.
Method: In this study, a deep convolutional neural network, its layers, activation and loss functions, regularization techniques, tools, methods, variants, and recent developments were explored to find its applications for COVID-19 prognosis. The pipeline of a general architecture for COVID-19 prognosis has also been proposed.
Result: This paper highlights recent studies of deep CNN and its applications for better prognosis, detection, classification, and screening of COVID-19 to help researchers and expert medical community in multiple directions. It also addresses a few challenges, limitations, and outlooks while using such methods for COVID-19 prognosis.
Conclusion: The recent and ongoing developments in AI, MI, and deep learning (Deep CNN) has shown promising results and significantly improved performance metrics for screening, prediction, detection, classification, forecasting, medication, treatment, contact tracing, etc. to curtail the manual intervention in medical practice. However, the research community of medical experts is yet to recognize and label the benchmark of the deep learning framework for effective detection of COVID-19 positive cases from radiology imagery.

Preface

This overview is the preprint of convolutional neural networks, its layers along with its variants and applications to combat COVID-19 under the umbrella of AI and ML. One of our motive is to accredit those who committed and contributed to the current state-of-the-art. And our objective is to provide the researchers of visual imagery with a concise summary of state-of-the-art CNN, its modules, tools and applications to get deeper insight for further exploration to detect, classify and combat COVID-19. I accept the limitations of attempting to achieve this goal. To cover most relevant and important publications we started from early works and relevant ideas and finally ending up with recent developments hierarchically. The present preprint contains good collection of references. Nevertheless, through an expert selection bias we may have unnoticed the essential work. Any corrections and suggestions to improve the work are most welcome and our pleasure.

1. Introduction

Machine learning is a field of study in which computers are enriched with the capability of acting without being explicitly programmed. Science and engineering are professionally fused in machine learning. As a science, it is concerned with achieving high-level interpretation from

* Corresponding author.
E-mail addresses: nadeem_01phd18@nitsri.net (N.Y. Khanday), shabir@nitsri.net (S.A. Sofi).

https://doi.org/10.1016/j.bspc.2021.102814
Received 15 February 2021; Received in revised form 18 May 2021; Accepted 23 May 2021
Available online 28 May 2021

Acronyms with definitions in lexical order

AI	artificial intelligence
ANN	artificial neural network
BM	Boltzmann machine
BoW	bag of words
CAE	convolutional autoencoder
CAP	community acquired pneumonia
CAT	computerized axial tomography
CNN	convolutional neural network
CRBM	conditional RBM
CRBM	convolutional RBM
CT	computed tomography
CV	computer vision
DBM	deep Boltzmann machine
DBN	deep belief network
ELU	exponential linear unit
FCN	fully convolutional networks
FFT	Fast Fourier Transformation
GAN	Generative Adversarial Network
GD	gradient descent
GPU	graphics processing unit
MAE	Mean Absolute Error
ML	machine learning
MLP	multilayer perceptron
MRI	magnetic resonance imaging
MRF	Markov random fields
MSE	Mean Squared Error
MSLE	mean squared logarithm error
NIN	network in network
RBM	restricted Boltzmann's machine
RCNN	recursive CNN
R-CNN	region based CNN
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SL	supervised learning
SVM	support vector machine
UL	unsupervised learning
WHO	World Health Organisation
YOLO	You Only Look Once

video sequences, digital imagery, multi-view camera geometry, or multidimensional data [1]. As an engineering and technological discipline, it is concerned with the construction of learning systems and applications by applying its theories and models. Hence, the main aim and focus is to develop computer models that make computers able to learn independently, after being fed with supervised training labels, generalize and realize signatures and relations. Increased chip computation like GPU units, improvement and development of highly efficient learning algorithms, and cheaper computer hardware are the main reasons for the blooming of machine learning today [2]. Over the years, due to the work of enthusiastic researchers a number of overview articles have explored aspects of learning models and their learning algorithms. Continuous improvements and modifications have been carried out to boost the performance metrics of different machine learning models and algorithms.

However, with dynamic phenomenons of nature, new problems arise from time to time which challenges human intelligence and experience. But without breaking down, each time humans try to resolve the problems by accepting challenges and came up with the best possible solutions. Wuhan, the city of China gave birth to such a problem when a new coronavirus in late 2019 called Coronavirus 2019 (COVID-19) crop up which is actually a contagious respiratory infectious disease induced by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). Very rapidly it spread across the globe and finally on 11-March 2020, WHO characterizes it as a ***Pandemic*** [3], which is the first-ever pandemic due to any class of coronavirus because it was having alarming levels of spread, severity, and inaction. After its outbreak, the count of cases started to upsurge exponentially in all the countries and shock the world with the number of deaths and newly infected cases on daily basis. By 18 May 2021, it has already infected 164,316,270 individuals across the globe with 3,406,027 unfortunate deaths. Not only it has crushed human lives, rattled markets, and disclosed the competence of governments but it could be the straw that breaks the camel's back of economic globalization and results in permanent shifts in economic and political power.

World Health Organisation responds with the best of its capabilities across the world on prevention, surveillance, containment, treatment, and coordination to combat COVID-19. Apart from WHO, every other country including NGO's adopted their national health missions, policies and utilized their research laboratories, resources to tackle the situation. At the same time, enthusiastic researchers left no stone unturned in their search for new strategies, ideas, techniques, and system models to combat COVID-19. The recent and ongoing progress in AI and ML has improved significantly the performance metrics to detect coronavirus from a CT scan or X-ray images at early stages. It has also improved analysis, medication, prediction, contact tracing, and vaccine/drug evolution process to curtail the human intervention in medical practice. Although in this global urgency, machine learning and artificial intelligence models play a vital role to help medical practitioners and researchers in the detection and prediction of coronavirus but still it is far away to shoot-out it completely. But to combat COVID-19 researchers are tirelessly working and putting in their effort, time and energy.

Different state-of-the-art machine learning algorithms like support vector machines (SVM) [4–6], Relevance vector machines, Random forest [7,8], Extreme machine learning, AdaBoost [9], different variants of regression [4], LSTM [10,11], and Deep neural networks [5,12,13,10,14–16] including deep convolutional neural networks and deep belief networks are adopted for automatic diagnosis, detection and classification (SARS, MERS, COVID-19) purposes and gained popularity by creating end-to-end modeling to achieve promised results. But COVID-19's alarming levels of spread and severity have necessitated the need for expertise. AI-based detection systems find great need and demand to assist radiologists to obtain an accurate diagnosis. Efforts are being continuously made to improve the performance metrics of learning models regarding COVID-19. It is hard to cover the contribution of machine learning or AI towards COVID-19 in a single paper. However, this work aims at highlighting the comprehensive overview and contribution of only convolutional neural networks and their applications for COVID-19. A substantial explanation of CNN has been provided in the following sections to clearly understand the CNN and its modules and finally how it can be used to automatic prognosis of COVID-19.

The rest organisation of this paper is structured as follows. Section 2 expounds the convolutional neural network and its three layers specifically convolutional layer (Section 2.1) and its types, pooling layer (Section 2.2) and its types and finally fully connected layer (Section 2.3). Different activation functions (Section 2.4), loss functions (Section 2.5) and regularization techniques (Section 2.6) for CNNs are also illustrated. Section 3 briefly explores alternate deep learning techniques. In Section 4 the pipeline of general architecture for COVID-19 prognosis has been proposed. Section 5 highlights applications of Deep CNNs for COVID-19. Finally, Section 6 covers challenges, limitations and outlooks to be explored in the future.

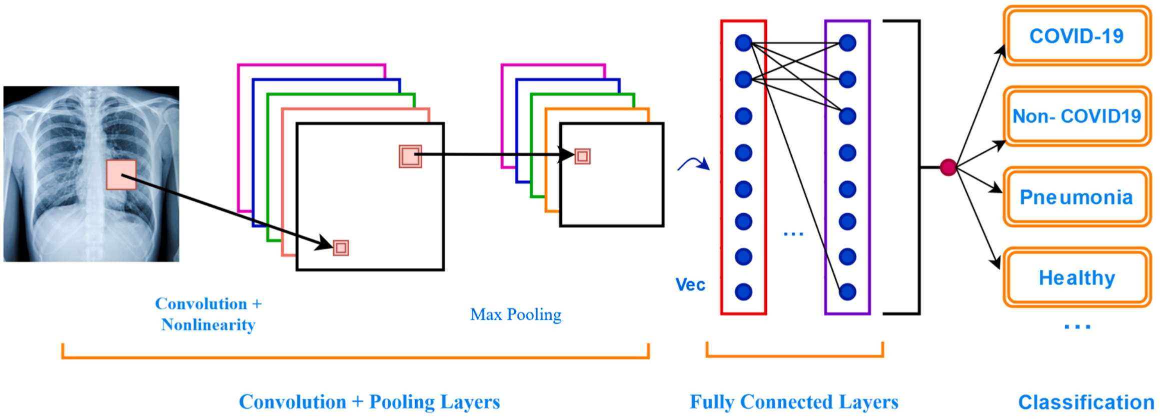


Fig. 1. The pipeline of general architecture of CNN.

2. Convolutional neural network

In diverse computer vision applications, the convolutional neural network is the most notable visual learning algorithm with acceptable performance in processing 2D data with grid-like topology like images and videos. The field of CNN is encouraged by receptive fields in the visual cortex of animals by [17] and later with multilayer artificial neural network LeNet-5 by [18–20]. CNN differs from Fukushima's neocognitron [21] by sharing weights in temporal dimensions to reduce computation and complexity using time-delay neural networks. There exists three main layers in any CNN model namely convolutional layer, pooling layer, and fully connected layer, each having different functions. CNN model is trained like any standard neural network using back-propagation and involves two steps, a forward step, and a backward step. In the forward step; in each layer, the input image is represented with parameters like bias and weights to finally compute loss cost, and in the backward step; chain rules are applied to compute gradients of each parameter to fine-tune the network. Various works have been proposed since 2006 to solve the problems in training deep CNNs [22–25] and finally [26] proposed CNN architecture, AlexNet [25,27], similar to LeCun's LeNet-5 but with deeper structure to reduce the training problems and has been strongly applied to diverse computer vision tasks [26,28–34]. Later improved and advanced developments like VGGNet [23], ZFNet [35], GoogleNet [24], ResNet [36], Inception [37], and SENet [38] raised the performance of CNNs to new heights. The general architecture of convolutional neural network is shown in Fig. 1.

Computer vision has achieved success with the improvements and variants of CNN. Recursive CNN (RCNN) was introduced by Eigen et al. [39] in 2013 which has tied filter weights across layers and the same number of feature maps in all layers. Jarret et al. [40] proposed a novel model by fusing convolution with an autoassociator for feature extraction which is useful for object recognition. Advanced stacked convolutional autoencoder was developed by Masci et al. [41] for unsupervised feature learning which achieved satisfying CNN initializations by averting the local minima of highly non-convex objective functions. Desjardins and Benjio [42] used RBM as a kernel to compute convolution in a CNN which composed convolutional restricted Boltzmann machine (CRBM). They achieved a higher convergence rate and unlike RBMs, CRBM's complexity depends only on the count of features to be extracted and the magnitude of the receptive fields. For hierarchical representations convolutional deep belief neural network (CDBN) are developed [43] to be applied on scaled unsupervised learning.

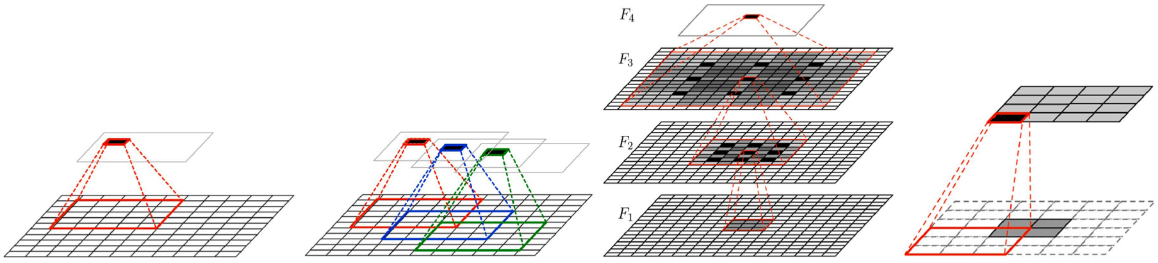
To avoid over-fitting, the need is to train the model with more and more training labels and to deal with large datasets, Mathieu and Henaff [44], proposed an algorithm using Fast Fourier transform (FFT) to quicken the training process with a significance factor by determining convolutions as products in the Fourier domain. The cuDNN [45] and fbfft [46] are some GPU based libraries that are introduced to increase the speed in the training and testing process. Using data management, data mining, data evaluation, data dispersion components, a novel dynamic graph convolutional network GCN is proposed which presents a never-ending learning platform called CUImage [47].

2.1. Convolution layer

A convolution operator is a generalized linear model and is considered as the central building block of convolutional neural networks which makes networks capable to construct informative features within local receptive fields at each layer by combining both spatial and channel-wise information [38]. It involves the multiplication between a 2D array of weights and an array of input data, called a kernel or a filter to convolve the full image as well as the intervening feature maps to generate various complete feature maps by different kernels. Mathematically, in ith feature map of nth layer, the feature value $V^{l}_{x,y,k}$ at location (x,y), is calculated by:

$$V^{l}_{x,y,i} = W^{l^T}_{i} Y^{l}_{i,j} + b^{l}_{i} \tag{1}$$

where b and W are the bias term and weight vector of the ith filter of the



(a) Convolution (b) Tiled Convolution (c) Dilated Convolution (d) Deconvolution

Fig. 2. [67] illustration of (a) convolution, (b) tiled convolution, (c) dilated convolution, and (d) deconvolution.

nth layer respectively, and Y is the input patch centred at location (x, y) of the nth layer. Activation functions (typically Sigmoid, *tanh* [48], ReLU [49]) of activation value $z^{l}_{x,y,i}$ of convolutional feature $V^{l}_{x,y,i}$ are applied to introduce non-linearity to CNN which can detect non-linear features desirable to multi-layered networks. The kernel is kept intentionally smaller than the input data. This allows it to remultiply the set of weights number of times with input array at different points on the input hence gives the capability of translational invariance to the model, which is not possible otherwise if the dimensionality of the filter output equals that of the inputs. The convolutional operation has many advantages [50] like (i) Parameter sharing mechanism to learn one set of parameters over the entire image instead of an isolated set of parameters at each location to scale down the number of parameters per response, hence reduces the overall complexity of the model and makes the training easier. (ii) Sparse interaction to reduce computational burden by learning correlations between neighboring pixels. (iii) Invariance translation or equivariance, a useful property to care about the presence of some feature than exactly where it is or to shift the responses according to the shifting of an object in an input image.

Some supporting works to enhance the representational ability of the convolutional layer are also introduced and showed in Fig. 2.

2.1.1. Tiled convolution

Tiled convolutional neural network [51] is a variation of CNN which with multiple maps gives the advantage of representing complex invariances like rotational and scale-invariant features by pooling over untied weights and decreases the number of learnable parameters.

2.1.2. Transposed convolution

Switching the backward and forward passes of a convolution, constitutes transposed convolution also called fractionally strided convolutions [52] or deconvolution [35,53–55] which identifies a single activation with numerous output activations. For the input feature map, dilation factor is given by the stride of deconvolution with padding to upsample the input and then perform the convolution operation on it. Application areas of deconvolution are visualization [35], recognition [56,57], super-resolution [58], semantic segmentation [59] and localization [60].

2.1.3. Dilated convolution

By introducing one or more hyper-parameter to the convolutional layer, a new variant called dilated convolutional neural network [61] has been developed. The receptive field of output units is cheaply increased without increasing the kernel size which allows the network to gain more relevant information for making better predictions. Scene segmentation [61], speech synthesis [62], recognition [63] and machine translation [64] are some highlighted tasks achieved with dilated convolution.

2.1.4. Network in network

Lin et al. [65] introduced a general network structure to increase the representational power of the neural network by replacing the linear filter of the convolutional layer with micro-network like multi-layer perceptron convolution to boost model discriminability for local patches within the receptive fields. Subsampling layers can be added in between mlpconv layers and stacking of such mlpconv layers gives the overall structure of network in network (NIN), on top of which lie the objective cost layer and the global average pooling.

2.1.5. Inception module

Using the structure of the network in networks [65], Szegedy et al. [24] proposed inception module, which uses 1 × 1 convolutions to increase both depth and width of their network without a significant performance penalty. Using the inception module, the drastic dimensional reduction is achieved by bringing down the number of network parameters to 5 million which are much less than those of AlexNet (60M) and ZFNet (75M), hence reduces the computational requirements. Architectures of [28] and [66] adopts inception module as the base network which helps them in effective object detection and localization.

2.2. Pooling layer

Generally, a pooling layer is a new layer added after the convolutional layer and is successfully used for the dimensional reduction of feature maps and network parameters. As pooling operation involves neighboring pixels for computation, so pooling layer is also translation-invariant like the convolutional layer. Boureau et al. [68] and Scherer et al. [69] provide details about the most commonly used pooling strategies, Average pooling, and Max pooling. Based on their analytics it was concluded that Max pooling can lead to faster convergence, improves generalization, and is utilized by most of the GPU implementations of various CNN variants [24,70].

There are several variations related to the pooling layers each with a different purpose, some are mentioned below.

2.2.1. L_p pooling

Inspired by [71] and theoretically analysed by [72], suggests that L_p pooling contributes satisfactory generalization than max pooling. L_p pooling is shown as:

$$Z_{p,q,i} = \Big[\sum_{a,b\epsilon R} (f_{a,b,i})^{n}\Big]^{1/n} \tag{2}$$

where $Z_{p,q,i}$ is result of the pooling operator at position (p, q) in i_{th} feature map, and $f_{a,b,i}$ is the feature value at position (a, b) within the pooling region $R_{p,q}$ in ith feature map. Max and Avg. pooling can be derived from L_p pooling by substituting $n = 1$, L_p corresponds to Avg. pool and $n = \infty$, L_p reduces to Max. pooling. A new variant of L_p unit called L_p-norm is proposed by [73] which replaces the max operator in a maxout unit and is more able in depicting complex, non-linear separating boundaries.

2.2.2. Stochastic pooling

Stochastic pooling [74], proposed by Zeiler et al., is a dropout-inspired pooling method that replaces the traditional deterministic pooling operations with a stochastic procedure, using multinomial distribution within each pooling region by selecting the activation randomly. Unlike Max. pooling, stochastic pooling avoids over-fitting because of the stochastic component.

2.2.3. Spectral pooling

Spectral pooling [75] achieves reduction in dimensionality by truncating the representation of input onto the frequency basis set. For the same number of parameters, spectral pooling preserves considerably more information and structure features for network discriminability than any other pooling strategy like max-pooling by exploiting the non-uniformity of particular inputs using linear low-pass filtering operation in the frequency domain. As spectral pooling only requires matrix truncation, which provides the advantage of being implemented at a negligible additional computational cost in CNNs that use Fast Fourier Transformations for convolution filters. A new approach of pooling called Hartley spectral pooling [76], is introduced for less lossy in the dimensionality reduction and reduces the computation by avoiding the use of complex arithmetic for frequency representation.

2.2.4. Spatial pooling

He et al. [77] introduced spatial pyramid pooling (SPP) which takes an image of any size as input hence provides flexibility by not only allowing arbitrary aspect ratio but also arbitrary scales [78,79] to generate fixed-length representation. Unlike sliding window pooling,

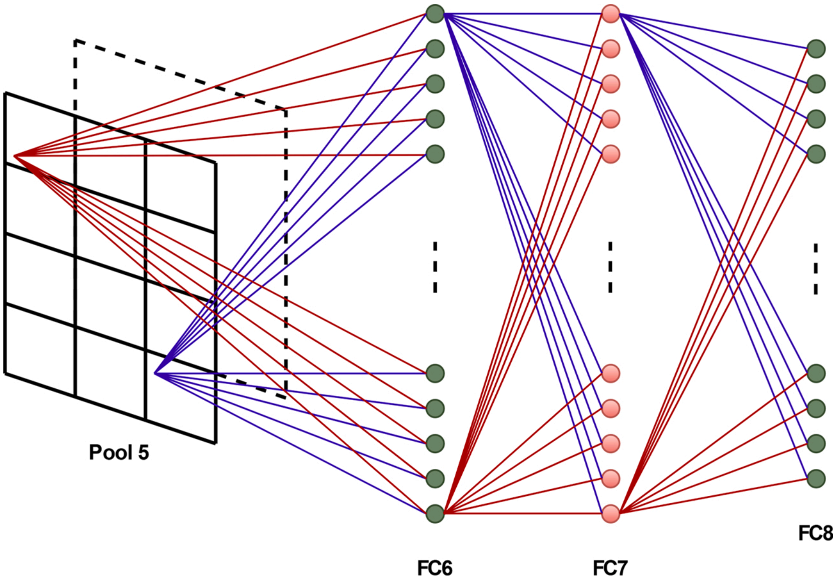


Fig. 3. The operation of fully-connected layer.

SPP improves Bag-of-words (BoW) by pooling input feature maps in local spatial bins with size proportional to the image size. Hence regardless of the image size, the number of bins will be fixed.

2.2.5. Def pooling

Computer vision tasks like object detection and recognition faces big challenge of handling deformations. Although Max pooling and average pooling handles deformations somehow fail to handle the geometrical model of object parts and deformation constraint. Ouyang et al. [80] offers a new deformation limited pooling layer called def pooling which handles deformation more efficiently by enabling the deep model to learn the deformation of visual signatures. As deformation stability at each layer changes significantly throughout training and often decreases, however, filter smoothness contributes significantly to achieving deformation stability in CNN's [81] and the level of learned deformation stability is determined by joint distribution of outputs and inputs.

The pooling approaches are designed for different purposes and involve different procedures, so diverse pooling methods could be fused to enhance the performance metrics of a convolutional neural network.

2.3. Fully connected layer

Certain fully-connected layers follow the extreme pooling layer of the network as shown in Fig. 1. In FCL all the inputs from one layer are connected to every activation unit of the next coming layer by converting the 2D feature maps into a 1D feature vector, which could be fed forward into a certain number of classes for classification [26]. 1D feature vector could be considered for further processing [28] like a compilation of the data extracted from previous layers to produce the final output. Few fully connected layers are compulsory in shallow CNN models where the final convolutional layer generated the features corresponding to a portion of the input image.

Fully connected layers perform like a traditional neural network and comprise most of the parameters (90%) of the network (CNN) to fit complex non-linear discriminant functions in the feature domain into which the input data parts are mapped. The operation of the fully-connected layer is illustrated by Fig. 3. AlexNet [27] has 60M parameters among which 58M parameters are in fully connected layers. Similarly, VGGNet [26] has total 138M parameters, among which 123M are in FC layers. The existence of a large number of parameters may end up in over-fitting the CNN model. However, techniques like stochastic pooling, dropout, data augmentation are introduced to avoid over-fitting. Moreover, [82] introduces a CNN architecture called SparseConnect which sparses the links between FC layers to reduce the problem of over-fitting.

Normally the structure and design of the fully connected layer is not changed, however, in the transferred learning method of image representation [83], the FC layers of the CNN model produce a means to learn rich mid-level image features transferable to diverse visual recognition tasks by preserving the learned parameters.

2.4. Activation function for CNN

The performance of a CNN for particular vision tasks can be significantly improved by a proper activation function. Some recently

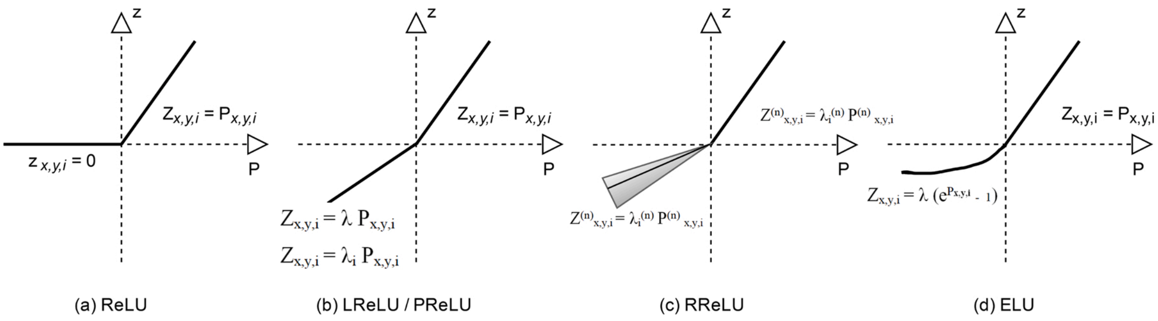


Fig. 4. The analogy among ReLU, LReLU, PReLU, RReLU and ELU.

introduced activation functions in CNNs are mentioned below.

2.4.1. ReLU

Rectified Linear Unit (ReLU) [84] is most evident unsaturated and successfully used activation function [85,86] in various tasks of CNN including image classification [24,26,87], speech recognition, natural language processes, and game intelligence [88], to name a few. For $P_{x,y,i}$ as input to activation function on the ith channel at location (x, y) (see Fig. 4(a)), the activation function for ReLU is calculated as:

$$z_{x,y,i} = \mathrm{Max}(P_{x,y,i}, 0) \tag{3}$$

ReLU is linear for all positive values, and zero for all negative values. Compared to Sigmoid and tanh activations, ReLU empirically works better [89,90] and is much faster to compute, makes the model to train even without pre-training [25] or run in less time and allows the network to converge very quickly due to linearity. Since ReLU prunes the negative part to zero which allows the network to be easily sparsely activated.

2.4.2. Dying ReLU

The problem of ReLU neurons is becoming inactive by having zero gradients whenever the unit is not active, refers to dying ReLU which drastically slows down the training process of the model. Once the ReLU neuron becomes inactive, it remains in a dead state for the whole procedure because gradient-based optimization never allows adjusting their weights. Deep ReLU network finally dies in probability, as the depth goes infinite. To alleviate this problem, recently Lu et al. [91] proposed a randomized asymmetric initialization procedure with erotically designed parameters to effectively prevent the dying ReLU.

Two other variants, called ***Leaky ReLU*** (LReLU) and ***Parametric ReLU*** (PReLU) introduced by Maas et al. [89] and He et al. [92] respectively (see Fig. 4(b)), are proposed to resolve the problem of dying ReLU.

Leaky ReLU is defined as:

$$z_{x,y,i} = \mathrm{Max}(P_{x,y,i}, 0) + \lambda \mathrm{Min}(P_{x,y,i}, 0) \tag{4}$$

where λ is predefined and ranges in (0,1). Leaky ReLU has a small gradient for negative values instead of mapping it to constant 0 which enables backpropagation. This not only fixes the dying problem but makes the activation function more balanced to speed-up the training process.

Randomized ReLU(RReLU) [93] is variant of Leaky ReLU, proposed and successfully used in Kaggle NDSB competition, where a random number λ_{xy} is taken from a uniform distribution [94,95] in training and then applied in testing to avoid zero gradients in negative part. Mathematically;

$$z_{x,y,i} = \begin{cases} P_{x,y,i}; & \text{if } P_{x,y,i} \geq 0 \\ \lambda_{xy} P_{x,y,i}; & \text{Otherwise} \end{cases} \tag{5}$$

Where $P_{x,y,i}$ denotes the input, at location (x, y) on ith channel, of activation function (see Fig. 4(c)). To get deterministic and stable outputs, method of dropout [96] can be used by taking average of all λ_{xy} in test phase only. Apart from improving performance, over-fitting can also be reduced due to randomized sampling of parameters.

Parametric Rectified Linear Unit (PReLU) boosts accuracy at negligible extra computational cost by adaptively learning the parameters of the rectifiers. PReLU function is computed as:

$$z_{x,y,i} = \mathrm{Max}(P_{x,y,i}, 0) + \alpha \mathrm{Min}(P_{x,y,i}, 0) \tag{6}$$

where α is the learned parameter common to all channels of one layer. The risk of over-fitting is eliminated by PReLU as it introduces very few parameters proportionate to the total number of channels. Backpropagation is used for training [19] and is optimized simultaneously with other layers as well.

2.4.3. ELU

Exponential linear unit (ELU) proposed by Clevert et al. [97] for faster and more accurate learning in CNN. Apart from fast learning, ELUs also lead to significantly better generalization performance than other variants of ReLUs. ELUs avoid vanishing gradient problem like ReLU, Leaky ReLU and Parametric ReLU do, by setting the positive part to identity, so their derivative part is one and not contractive. Unlike ReLU, the negative part is present in ELUs, which is responsible for faster learning as it decreases the gap between the normal gradient and the unit natural gradient likewise in batch normalization. ELUs saturate to a negative value when the argument gets smaller, hence decreases the information propagation and variation to the next layer, unlike Leaky ReLU and Parametric ReLU. Mathematically ELU is computed as:

$$f(p) = \begin{cases} P_{x,y,i}; & \text{if } P_{x,y,i} > 0 \\ \lambda(e^{P_{x,y,i}} - 1); & \text{if } \leq 0 \end{cases} \tag{7}$$

where λ is a predefined hyperparameter for regulating the value to which an ELU saturates for negative inputs (see Fig. 4(d)). ***Parametric ELU*** (PELU) [98], variant of ELU, proposed that learning a parameterization of ELU improves its performance. Using gradient-based framework on ImageNet [26] with different network architectures, PELU has relative error improvements over ELU.

2.4.4. Maxout

The maxout model is feed-forward architecture, using maxout unit as activation function [99] which returns the maximum across K affine feature maps, designed to be used in conjunction with the dropout which eases the training and makes it robust to achieve excellent performance. Maxout activation function is computed as;

$$z_{x,y,i} = \max_{k\epsilon[1,i]} P_{x,y,i} \tag{8}$$

Where $P_{x,y,i}$ is the input on ith channel of the feature map.

ReLU and its variants like Leaky ReLU are specific cases of maxout, hence it relishes all the profits of ReLU family without facing the dying ReLU problem, however number of parameters for every neuron increases. [100] proposed a new type of maxout, called ***Probout***, by replacing the max operation with a probabilistic sampling procedure, which is partially invariant to changes in its input. Probout successfully achieved the balance between retaining the tempting properties of maxout units and boosting their invariance properties.

2.5. Loss function

Models learn through loss function. It calculates the penalty by comparing the actual value and predicted value hence helps in optimizing the parameters of a neural network. For a specific task selecting an appropriate loss function is important. For regression problems, Mean Squared Error (MSE) is a commonly adopted loss function and is computed as the mean of the squared differences between the actual and predicted values [101]. Mean Squared Logarithmic Error (MSLE) is used when the model is forecasting unscaled quantities directly. Other variants Mean Absolute Error Loss (MAE) is used when the target variable is associated with outliers. For binary classification problems, cross-entropy is the default loss function. It computes the value that compiles the mean difference between the real and expected probability distributions for predicting the particular class. Variant of cross-entropy for binary classification is called ***Hinge loss***, primarily utilized for training large margin classifiers most notably for Support Vector Machines [102,103]. Hinge loss is a convex function and has many extensions. For an expected result $t = \pm 1$ and a classifier score Y, then Hinge loss can be mathematically represented as:

$$\mathrm{Hinge}_{\mathrm{Loss}} = 1 \Big/ \mathrm{N} \sum_{1}^{N} \sum_{1}^{K} [0, (1 - t).Z]^{n} \tag{9}$$

Where $Z = Y^T X_i + b$ and Y is weight vector of classifier. If $n = 1$, then it is simple Hinge-Loss (L_1-Loss); while if $n = 2$, then it is squared Hinge-Loss (L_2-Loss) [104], which on investigation [105] proves to be more effective than Softmax on MNIST [106]. Robustness is also significantly improved as demonstrated by [107].

Cross-entropy also called Logarithmic loss are also used for multi-class classification problems which are fundamental in computer vision, where the expected values are in the set $0,1,2,3,\ldots,n$, where each class is accredited an exclusive integer value. Multi-classification may suffer from space complexity because of one hot encoded vector for the expected element of each training example. However, ***sparse-cross entropy***, variant of cross-entropy resolves this problem by computing loss without requiring one-hot encoding for the target variable prior to training.

Kullback Leibler Divergence is a non-symmetrical measure of how a particular probability distribution *P(x)* varies from the baseline distribution *q(x)* over the similar values of discrete variable *x*. KL divergence is most notably used in autoencoders [108–110].

The symmetrical form of KLD is called Jensen-Shannon Divergence (JSD) which determines the analogy between *P(x)* and *q(x)*, by minimizing which two distributions can be as close as possible, to be used in Generative Adversarial Networks (GANs) [111–113].

Softmax Loss: Due to probabilistic interpretation and simplicity of softmax function, Softmax loss fuses it with a multinomial logistic loss to form arguably one of the most commonly used components in CNN architectures. Recently, [114] propose Large-Margin Softmax (L-Softmax) loss, which defines a flexible learning task with hyperparameter to adjust the margin which can effectively avoid the problem of over-fitting.

Contrastive Loss: To train Siamese network [115–118], contrastive loss function is most commonly used, which is distance-based loss function for learning a correlation measure from pairs of data points representation tagged as matching or non-matching. Lin et al. [119] proposed double margin loss function using additional margin parameter which can be interpreted as learning local large-margin classifiers to differentiate matching and non-matching elements.

Triplet Loss: Triplet loss [120] considers three instances namely an anchor instance (x_a), positive instance (x_p) from the same class of (x_a) as well and a negative instance (x_n).

Mathematically triplet loss is computed as:

$$\mathrm{Loss}_{\mathrm{Triplet}} = \sum_{i}^{N}[\| |\mathrm{f}(x_i^a) - \mathrm{f}(x_i^p) \| |_2^2 - \| |\mathrm{f}(x_i^a) - \mathrm{f}(x_i^n) \| |_2^2 + \lambda] \tag{10}$$

Where $\mathrm{f}(x)$ takes x as input, λ is a margin that is enforced between positive and negative pairs, i denotes ith input, a,p,n denotes the anchor, positive and negative instances respectively.

The primary goal of triplet loss is to decrease the distance between an anchor and a positive both having the same identity and maximize the distance between the negative and an anchor of a different identity. Liu et al. [121] proposed Coupled Cluster (CC) which is characterized over the negative set and the positive set. CC works well in backpropagation as well by changing the randomly selected anchor with the cluster center as it aggregates the samples in the negative set and samples in the positive set to increase the reliability.

A new loss function for CNN classifiers is introduced by Zhu et al. [122], inspired by predefined evenly-distributed class centroids (PEDCC), called ***PEDCC-loss***, which makes inter-class distance maximum and intra-class distance short enough in hidden feature space by replacing the classification linear layer weight with PEDCC weights in CNNs achieved best results in image classification tasks, improves stability in network training, and results in fast convergence.

2.6. Regularization

Regularization is one of the key elements of machine learning and a crucial ingredient of CNNs, which makes the model generalize better as it intends to reduce the test error but not training error. As the introduction of a large number of parameters often leads to over-fitting, which can be effectively reduced by regularization. Various variants of regularization techniques have emerged in defense of over-fitting e.g. sparse pooling, Large-Margin softmax, L_p-norm, Dropout, Dropconnect, data augmentation, transfer learning, batch normalization, and Shakeout are notable ones.

2.6.1. L_p-norm

Normally regularization adds term $\lambda\,\mathrm{R}(\theta)$ to the objective function which keeps and considers all the features but reduces their magnitude of parameters θ_j to penalize the model complexity. Mathematically, if the loss function is $\mathrm{J}(\theta,x,y)$, then the regularized loss will be:

$$\mathrm{J}_{\mathrm{Reg}}(\theta,x,y) + \lambda\mathrm{R}(\theta) \tag{11}$$

where $\mathrm{R}(\theta)$ is regularization term. λ is regularization parameter (to be chosen carefully).

For L_p-norm, regularization function is computed as:

$$\mathrm{R}(\theta) = \sum_{j} \| \theta_j\|_p^p \tag{12}$$

when $p \geq 1$, the L_p-norm is convex and has been widely used to induce structured sparsity in the solutions to various optimization problems [123,124]. For $p = 2$, it is referred to as weight decay.

2.6.2. Dropout

Hinton et al. [125] introduced Dropout, and expounded in-depth by Baldi et al. [126], which reduces over-fitting effectively by randomly dropping units (hidden and visible). The links are also released from the neural network during training to avoid complicated co-adaptations [125] on the training data and boosts the generalization ability. In Dropout the probability P is assigned to each element of a layer's output, otherwise is set to zero with probability $(1 - P)$. Similar to standard neural networks, Dropout is trained using stochastic gradient descent and provide a means to combine exponentially many heterogeneous neural network architectures effectively. Dropout holds the accuracy even though some information is missing and makes the network independent from anyone or a small fusion of neurons. Several variants of Dropout have been introduced for further improvements including the fast Dropout method by Wang et al. [127], adaptive Dropout by Ba et al. [128], and their detailed representation in [129–132]. Spatial Dropout [133] which expands the Dropout values across the feature map and works efficiently when the size of training data is limited.

2.6.3. Dropconnect

Inspired by Dropout, Dropconnect provides the generalization of Dropout by randomly setting the elements of weight matrix W to zero, rather than randomly setting each output unit to zero. In other words, instead of activations it randomly drop the weights. Although relatively slower Dropconnect achieves satisfactory results on a variety of standard benchmarks. Dropconnect also introduces dynamic sparsity on the weights W instead of output vectors of a layer. Additionally, biases are also masked out during training.

2.6.4. Shakeout

Recently, Guoliang Kang et al. [134] introduced a new variant of regularization, called Shakeout, by slightly modifying the concept of Dropout. Unlike Dropout, which randomly discards units, Shakeout randomly chooses to boost or reverse each unit's improvement to the next layer in the training stage. Regularizer of Shakeout enjoys the benefits of L_0, L_1 and L_2 regularization terms adaptively. Shakeout performs better than Dropout when the data is deficient. Much sparser weights [135] are obtained by Shakeout, and it also leads to better generalization. The instability of the training process is reduced by

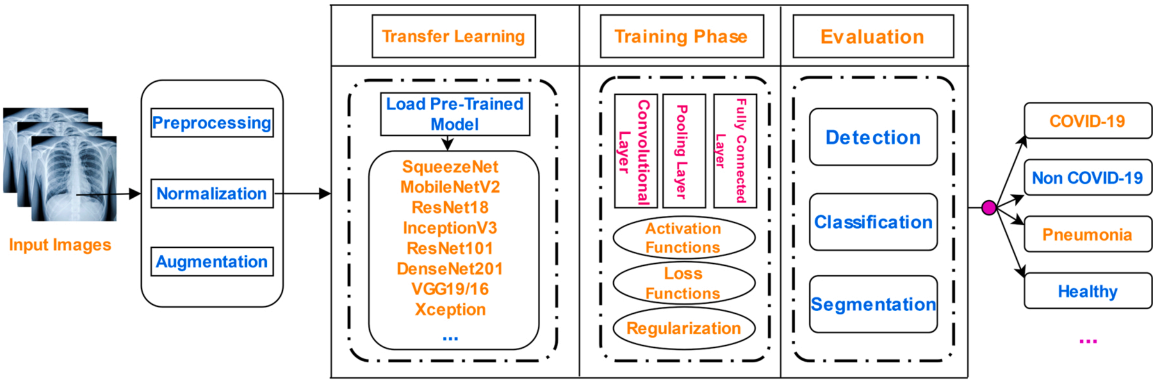


Fig. 5. The pipeline of general architecture for COVID-19 prognosis.

Shakeout for deeper architectures.

2.6.5. Data augmentation

Data augmentation, a data-space solution to an issue of deficient data by transforming existing data into advanced data without changing their natures, is applied notably for various computer vision tasks. It prevents over-fitting by modifying limited training dataset [136]. Popular data augmentation methods include geometrical transformations such as mirroring [137], sampling [25], shifting [138], rotating [139], color space augmentations, color transformations, random erasing, kernel filters, mixing, feature space augmentation, adversarial training, GANs, meta-learning and various photometric transformations [140]. To select the best suitable transformation from a bag of candidate transformations is done by greedy strategy proposed by Paulin et al [141]. Dosovitskiy et al. [142] proposed data augmentation based unsupervised feature learning, [143] and [144] introduces a way of gathering images from online sources to improve learning in different visual recognition tasks.

Apart from above-described regularization methods, there are some other methods like weight decay, weight tying and many more in [145] which avoids over-fitting by decreasing the number of parameters by having better representations of the input data [51] or help to have better generalization [26]. The important thing about these regularization techniques is that they can be fused to boost performance as they are not mutually exclusive.

3. Alternative deep learning techniques

Although CNN is the prominent deep learning technique to be used for different computer vision applications [193] like detection, classification, etc. However, there exist other state-of-the art techniques which justify prominent results for these applications in different domains. Some of these techniques are briefly mentioned below.

- ***Deep Restricted Boltzmann Machine:*** Hinton et al. [146,147] proposed a generative stochastic neural network which is an energy-based model and primary variant of Boltzmann machine [148], called Restricted Boltzmann machine (RBM) [149,150]. RBM is a special variant of BM with restriction of forming bipartite graph between hidden and visible units. Due to flexibility of RBMs, they play vital role in various applications such as dimensional reduction, topic-modelling, collaborative filtering, classification and feature learning. Using RBMs as basic building block, novel models can be build which include: ***Deep Belief Networks:*** (DBN) [151,152], ***Deep Boltzmann Machine:*** (DBM) [153]; and ***Deep Energy models:*** (DEM) [154]. Although RBMs does not promise better performance than CNNs related to computer vision tasks, however the developments and variants which adopt RBMs as building blocks leads to better performance.
- ***Autoencoder:*** An autoencoder (AE, also called Autoassociator) is a typical unsupervised learning algorithm of artificial neural network that learns a non-linear mapping function between data and its feature space. AE is used for learning efficient encodings [155] mainly for the purpose of dimensional reduction [156]. The output vectors dimensionality is identical to input vector because an autoencoder is trained to rebuild its own input X than to predict some target value Y.
- ***Recurrent Neural Network:*** The basic essence of RNNs [157] is to consider the influence of past information for generating the output. LSTM, BiLSTM, and GRU are prominent and powerful RNN models that are efficient for time-dependent in time-series data.
- ***Sparse coding:*** Sparse coding [135,158] is powerful and an effective data representation method. Its representations are considered as a strong technique for modelling high dimensional data. Due to applications of sparse coding in image classification, image restoration, image retrieval, image denoising, image clustering, recognition tasks, image and video processing and much more it has attracted considerable attention in recent years.
- ***Transfer Learning:*** Transfer learning aims to provide a framework to utilize previously-acquired knowledge to solve new but similar problems much more quickly and effectively [159]. Taking advantage of a pre-trained model (CNN) on a huge database to help with the learning of the target task (prognosis of COVID-19) that has limited training data.
- Random Forest algorithms [7,8], Ensemble techniques (ensemble-stacking, bagging, boosting), GANs are other prominent deep learning techniques which have been adopted in different segments for COVID-19 prognosis.

4. General architecture for COVID-19 prognosis

Although the world was rocked by the COVID-19 pandemic, but researchers rose to all manner of challenges to tackle this pandemic by adopting the shreds of evidence of ML and AI in previous epidemics to develop novel models, methods, and strategies. Enthusiastic researchers left no stone unturned in their search for new strategies, ideas, techniques, and system models to combat COVID-19. But due to dynamic behaviour and mutations of SARS-CoV-2 different variants have emerged and more are expected to occur over time. Moreover, asymptomatic cases and persons with newly found syndromes like joint pain, gastrointestinal symptoms, eyeball stress, etc. left medical practitioners baffled. Because of the development of new strains due to mutations in the virus, the predominance of organ involvement can change. So this makes it more challenging for researchers to combat COVID-19. Despite the fact that ML, AI, and deep learning has shown promising results and significantly improved the performance metrics for COVID-19 prognosis but the research community of medical experts is yet to recognize and label the benchmark of the deep learning framework for effective detection of COVID-19 positive cases from radiology imagery. So there is need to enhance the architecture of learning models which can provide flexibility towards newly variants to be detected and can boost the performance metrics as well. The pipeline of such general architecture which can be adopted for COVID-19 prognosis has been proposed and shown in Fig. 5.

The above architecture illustrates the general approach to be adopted for COVID-19 prognosis, however the techniques, tools and methods may vary from one domain to another. Preprocessing involves noise filtering, region of interest, resizing, and conversion from grey to RGB or viceversa. Normalization helps to learn faster and improves convergence stability. Although methods like rotation, flipping, scaling, and Gaussian noise helps in data augmentation to increase the volume of data but still learning models need more data to get trained efficiently. Hence, transfer learning technique takes advantage of a pre-trained model on a huge database to help with the learning of the target task that has limited training data. Layers of adopted learning model also needs to be adjusted with proper activation functions, loss functions and regularization techniques for optimized results. Lastly in evaluation phase, detected part needs to be classified, segmented and analysed for better prognosis.

5. Applications

In this portion, we survey some recent works that leveraged deep learning methods particularly convolutional neural network and its types to achieve state-of-the-art performance in diverse tasks to combat COVID-19 such as CT Scan/X-ray image classification (SARS, MERS, COVID-19), detection and recognition of coronavirus from imagery (CT or X-ray), instance segmentation of infected part in medical images, forecasting, prediction and contact tracing to name a few. The deep learning techniques adopts different types of visual radiography images for training and testing purpose, hence all are categorized as supervised learning class. Much valuable state-of-the-art of machine learning and artificial intelligence with novel models and algorithms were proposed by different researchers which show promising results in many applications regarding COVID-19 [160] and augmented the researchers on multiple angles to fight against the novel coronavirus outbreak.

In earlier stages of pandemic (COVID-19) [3,161–167] contribute a lot more about the earlier insight of COVID-19, however with time and deeper insight of coronavirus like its symptoms, possible anatomy, behavior, more visual chest CT/X-ray imagery of COVID infected patients, etc. novel algorithms and models were introduced by noble researchers. Using AI techniques, [14] distinguished the COVID-19 from Non-COVID pneumonia by adopting 10 well known CNNs including AlexNet, SqueezeNet, GoogleNet, VGG-19, VGG-16, ResNet-101, ResNet-50, ResNet-18, MobileNet-V2, and Xception. Among all networks, the highest performance in classifying or categorizing COVID-19 and Non-COVID-19 infections was obtained in Xception but it did not have the best sensitivity while ResNet-101 has the highest sensitivity although lower specificity. [13] introduced a model called ***DarkCovidNet*** adopting DarkNet and YOLO models for automated detection of COVID-19 from X-ray images of chest using deep neural network. They provide binary classification (COVID vs. No findings) with 98.08% accuracy and multi-class classification (COVID vs. No finding vs. pneumonia) with 87.02% accuracy. Lin Li et al. [168] developed a 3-Dimensional deep learning framework called ***COVNet*** consists of ResNet-50 as backbone for the detection of COVID-19 from CT scans of chest. It extracts both 3D global and 2D local representative features effectively from CT scans of the chest to differentiate COVID-19 and Community-acquired-pneumonia (CAP) with high sensitivity of 90% and specificity 96%. Based on SqueezeNet, a light CNN design was proposed by [12] for the efficient detection of COVID-19 from CT scan images.

Amongst a pool of deep CNNs (MobileNet, Xception, DenseNet, ResNet, Inception V3, InceptionResNet v2, NASNet, VGGNet) and different machine learning algorithms, the best combination of feature extractor and high-performance metric learners were obtained for the computer-aided prognosis of COVID-19 Pneumonia in [169]. From their work, DenseNet121 feature extractor equipped with bagging tree classifier gains a high performance with 99% accuracy in classification followed by a hybrid of the ResNet50 feature extractor trained by LightGBM with an accuracy metric of 98% on their dataset. On chest X-ray imagery, [170] proposed 5 pre-trained CNN based models (ResNet152, ResNet101, ResNet50, Inception V3 and Inception-ResNet V2) for effective detection of coronavirus pneumonia patient. According to their work, the ResNet50 model achieves the highest performance in classification on different datasets. Integrated stacking ***InstaCovNet-19*** model was proposed by [171], utilized different pre-trained models like Xception v3, ResNet101, etc. to compensate for a relatively small amount of training data. They achieved binary classification (COVID-19, Non- Covid) with 99.53% accuracy and 3-class (COVID-19, normal, pneumonia) classification with 99.08% accuracy. Inspired by AlexNet, a deep Bayes-squeezeNet decision-making system is used as backbone model by [172] for the COVID-19 prognosis from X-ray images. Bayesian optimization augmented dataset and fine-tuned hyperparameters boost the network performance in diagnosis and outperform its competitors.

CoroNet, a deep CNN model was proposed by [173] which without any human intervention detect COVID-19 viral infection from chest X-ray images. CoroNet adopts Xception architecture which is pre-trained on the ImageNet dataset and gains an overall accuracy of 89.6% for 4-class classification (covid vs. bacterial pneumonia vs. viral pneumonia vs. Normal) and 95% accuracy for 3-class classification (covid vs. Normal vs. pneumonia). Making use of a lightweight residual projection-expansion-projection-extension (PEPX) design pattern, [174] proposed a diverse architecture called ***COVID-Net*** which enhanced representational capacity while reducing computational complexity to generate a network for the prognosis of COVID-19 cases from chest X-ray images. Inspired by capsule networks, a new framework called ***COVID-CAPS*** was proposed by [175] which overcomes the problem of transformations of X-ray imagery of chest to preserve spatial information for effective identification of COVID-19 cases and achieved an accuracy of 95.7%, the specificity of 95.8% and sensitivity of 90% respectively.

Evaluation of state-of-the-art CNN architectures was carried out in [176] using transfer learning [177] and achieved remarkable results on small datasets as well with an accuracy of 96.78%, the sensitivity of 98.66% and specificity of 96.46% respectively. Likewise [10] proposed two architectures (i) ***mAlexNet*** and (ii) hybrid of mAlexNet and Bidirectional Long Short Term Memories (BiLSTM). They contribute to performing ANN-based automatic lung segmentation [178] from X-ray images to obtain robust features and then for early detection of COVID-19 infection they developed a CNN-based transfer learning-BiLSTM network which achieved 98.7% accuracy. Adopting transfer learning technique and image augmentation for the training and validation of various pre-trained deep CNNs were utilized by [179] to introduce a robust technique for the automatic prognosis of COVID-19 infection. For 2-class classification (COVID and Normal) and 3-class classification (Normal, COVID-19 Pneumonia, and Viral) they achieve classification accuracy, precision, sensitivity and specificity as 99.7%, 99.7%, 99.7%, 99.55% and 97.9%, 97.95%, 97.9%, 98.8% respectively.

A deep learning framework called ***COVID-ResNet*** was proposed by [180] which utilizes state-of-the-art training techniques like progressive resizing, discriminative learning rates, and cyclical learning rate finding to train accurately, quickly residual neural networks, and fine-tuned the pre-trained ResNet-50 model for better performance metrics to reduce training time. Newly deep learning framework called ***COVIDx-Net*** was proposed by [181] which includes seven different architectures of deep CNN models (including DenseNet, MobileNet, VGG19). Amongst all VGG19 and DenseNet perform better for the classification of COVID-19 with 89% and 91% accuracy on their dataset. A promising attempt of

Table 1
Deep CNN approaches with their methods/models, data type used and performance metrics for COVID-19.

Method	Data Type	3 class (**)	2 class (*)	Ac%	Pr%	Sn%	Sp%	Architectural properties	Dataset	Ref.
COVID-Net	Chest X-Ray	✓	✓	93.30	92.4	91.0	NA	Tailored deep CNN have been used. Heterogeneous mix of convolution layers with diversity of kernel sizes (7x7 to 1x1) and grouping configurations are used. 1x1 convolutions are used for either input features to lower dimension or for expanding features to higher dimension. 3x3 are adopted for reducing computational complexity.	COVIDx dataset has been used which contain 13,975 CXR images across 13,870 patients.	[174]
Faster- RCNN	Chest X-Ray	×	✓	97.36	99.29	97.65	95.48	VGG-16 network based Faster Regional CNN have been used. 16 weight layers are used with stacked 3x3 convolutional layers. Max. pooling is used. 2 FC layers with 4096 units are used with Softmax classifier.	183 COVID-19 +ve X-ray images and 13,617 Non-COVID X-ray images of chest were used.	[16]
VGG-19	Chest X-Ray	✓	✓	98.0* 93.0**	NA	92.0	98.0	Adopted transfer learning with CNN. Stacked 3x3 convolutional layers are used. Max. pooling is used for dimensional reduction. Two fully connected layers each with 4096 nodes are used with Softmax classifier. 19 weight layers are used.	Total 1427 X-ray images including 224 COVID-19 +ve case images were used.	[176]
Deep CNN ResNet-50	Chest X-Ray	✓***	✓***	98.0	NA	96.0	100	Adopted 3 variants of CNN models (InceptionV3, ResNet50, InceptionResnetV2). ResNet-50 has 48 convolution layers with 1 Max. pool and 1 Avg. pool layer. It over 23 million trainable parameters. Moreover, 50 weight layers are used.	Data used include 50 open access COVID-19 X-ray images from Joseph Cohen and 50 typical images from kaggle repository.	[170]
COVIDX-Net	Chest X-Ray	×	✓	90.0	100	100	NA	COVIDx-Net adopts 7 CNN models. VGG9, DenseNet201, ResNetV2, InceptionV3, InceptionResNetV2, Xception, and MobileNetV2	Data used contain 50 chest X-ray images (25 covid +ve, 25 healthy)	[181]
DRE-Net	Chest CT scan	×	✓	86.0	80.0	96.0	NA	Adopted ResNet50. ResNet-50 has 48 convolution layers with 1 Max. pool and 1 Avg. pool layer. It over 23 million trainable parameters. Moreover, 50 weight layers are used.	CT images of chest (777 covid +ve, 708 healthy).	[162]
M-Inception	Chest CT scan	×	✓	82.90	NA	81.0	84.0	Modified Inception is used. Use of 7x7 convolution is adopted. 48 layers of deep CNN has been utilized.	Total 453 images (including 195 COVID-19 +ve and 258 Non-COVID-19 images.) were used.	[163]
ResNet	Chest CT scan	×	✓	86.7	81.03	86.7	NA	3D CNN model with location-attention mechanism has been adopted. 2 convolutional layers interspersed with 2 max pooling layers followed by 2 fully connected layers are used.	Total 618 CT samlpes containing 219 images with COVID-19, 175 normal CT images, and 224 images containing viral pneumonia were used.	[164]
Dark CovidNet	Chest X-Ray	✓	✓	98.08* 87.02**	98.03* 89.96**	95.13* 858.35**	95.3* 92.18**	Based on DarkNet model. DarkNet is open source neural network framework using C and CUDA. YOLO is adopted for better detection. YOLO is 106 layer fully convolutional architecture with consecutive 3x3 and 1x1 convolutions.	Totally 1125 images containing 500 pneumonia, 125 COVID-19, and 500 with no findings were used.	[13]
COVNet	Chest CT scan	✓	×	96.0	NA	90.0	96.0	ResNet50 is used as backbone. ResNet-50 has 48 convolution layers with 1 Max. pool and 1 Avg. pool layer. It over 23 million trainable parameters. Moreover, 50 weight layers are used.	4352 chest CT scans from 3322 patients are collected containing (1292 covid +ve, 1735 CAP, and 1325 Non-pneumonia abnormalities)	[168]

(*continued on next page*)

Table 1 (*continued*)

Method	Data Type	3 class (**)	2 class (*)	Ac%	Pr%	Sn%	Sp%	Architectural properties	Dataset	Ref.
Insta CovNet-19	Chest X-Ray	✓	✓	99.53* 99.08**	100* 99.0**	99.0* 99.0**	NA	Integrated stacked deep CNN utilizing pre-trained models like Xception, ResNet101, MobileNet, InceptionV3, and NasNet. Xception is 71 layers deep network and uses 36 (3x3) convolutional layers for feature extraction with 16 input and 32 output channels. It has 4608 parameters. MobileNet has 28 layers and 4.2M parameters.	Combined dataset [179, 166] is used which contain 361 covid images, 1341 pneumonia and 1345 normal images.	[171]
COVI Diagnosis - Net	Chest X-Ray	✓	×	98.26	98.26	98.26	99.13	Based on deep Bayes-SqueezeNet. A squeeze convolutional layer has only 1 1 filters. Max. pooling, RELU activation and Dropout techniques are adopted in SqueezeNet. FC layer is absent.	COVIDx dataset [174] containing 5949 chest radiography images from 2839 patients (1583 normal, 4290 pneumonia, 76 covid infected.	[172]
CVDNet	Chest X-Ray	✓	×	96.69	96.72	96.84	NA	Adopts residual Neural Network. ResNets can have variable sizes, depending on how big each of the layers of the model are, and how many layers it has e.g. 34, 50, 101, etc. 3 × 3 convolutions are normally used with kernel size of 7, and a feature map size of 64. Max. pooling is adopted.	Dataset [179,166] containing 219 COVID-19, 1345 with viral pneumonia and 1341 healthy images.	[163]
CoroNet	Chest X-Ray	✓	✓	99.0* 95.0** 89.6***	98.3* 95.0** 90.0***	99.3* 96.9** 89.92***	98.6* 97.5** 96.4***	Adopted Xception architecture. Xception is 71 layers deep network and uses 36 (3x3) convolutional layers for feature extraction with 16 input and 32 output channels. It has 4608 parameters.	1251 chest X-ray images containing 310 normal, 327 pneumonia viral, 330 pneumonia becterial, and 284 covid-19)	[173]
COVID-CAPS	Chest X-Ray	×	✓	95.7	na	90.0	95.8	Based on Capsule Networks. Encoders and decoders constitute 6 layers in capsule network.	COVIDx dataset [174] has been used.	[175]
mALexNet + BiLSTM	Chest X-Ray	×	✓	98.70	98.77	98.76	99.33	Adopts residual Neural Network. ResNets can have variable sizes, depending on how big each of the layers of the model are, and how many layers it has e.g. 34, 50, 101, etc. 3 × 3 convolutions are normally used with kernel size of 7, and a feature map size of 64. Max. pooling is adopted.	Dataset [179,166] containing 219 COVID-19, 1345 with viral pneumonia and 1341 healthy images.	[10]
COVID - ResNet	Chest X-Ray	✓***	✓***	96.23	100	100	100	Based on ResNet50 model. ResNet-50 has 48 convolution layers with 1 Max. pool and 1 Avg. pool layer. It over 23 million trainable parameters. Moreover, 50 weight layers are used.	COVIDx dataset [174] has been used.	[180]

[16] by proposing OxfordNet network-based faster regions with CNN (***COVID Faster R-CNN*** framework to diagnose novel coronavirus disease from X-ray chest imagery. They achieved an accuracy of 97.36%, a sensitivity of 97.65%, and a precision of 99.28%. A deep CNN model called ***CVDNet*** adopting residual neural network was proposed by [182] to classify COVID-19 from healthy and other pneumonia infected cases using X-ray images of the chest. Although the dataset used was small but still, achieves an accuracy of 97.20% for the prognosis of COVID-19, and for 3-class classification (Covid vs. Viral Pneumonia vs. Normal) it achieves 96.69% accuracy. An ensemble model by fusing StackNet metamodeling and deep CNN called ***CovStackNet*** was proposed by [183] for fast prognosis of COVID-19 using chest X-ray images. CovStackNet reached an accuracy score of 98%.

Although a lot more has been suggested, proposed, and implemented regarding COVID-19 as shown in Table 1,[1] however, there still exists a huge space and gap for researchers to work on so that state-of-the-art deep learning models could be officially recognized by the research community of medical expertise for COVID-19 positive case detection from radiology images. Although Ramdesiver, Lopinavir, Ritonavir, oseltanivir, and Pfizer significantly blocked the COVID-19 infection but researchers are continuously and tirelessly working to develop efficient therapeutic strategies and trials to cope with and complete deportation of human coronavirus.

[1] Ac, Pr, Sn, and Sp represents Accuracy, Precision, Sensitivity, and Specificity in Table 1 respectively. *, **, *** represents 2-class, 3-class and 4-class classification in Table 1 respectively.

Table 2
Comparison of open source and most commonly used Datasets by different research works for COVID-19.

Study	Data type	Application	Source
[186]	X-ray and CT scan	COVID-19 diagnosis	https://github.com/ieee8023/covid-chestxray-dataset
[187]	CT scans	COVID-19 diagnosis	https://github.com/UCSD-AI4H/COVID-CT
[163]	CT scans	COVID-19 diagnosis	https://ai.nscc-tj.cn/thai/deploy/public/pneumoniact
[188]	Segmented CT scans	COVID-19 infected area segmentation	https://zenodo.org/record/3757476
[174]	X-ray imagery	COVID-19 diagnosis	https://github.com/lindawangg/COVID-Net, https://github.com/agchung/Actualmed-COVID-chestxray-dataset
[181]	X-ray imagery	COVID-19 diagnosis	https://github.com/ieee8023/covid-chestxray-dataset
[170]	X-ray imagery	COVID-19 diagnosis	https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia + [186]
[189]	X-ray and CT scan	COVID-19 data set augmentation	https://data.mendeley.com/datasets/8h65ywd2jr/3
[190]	Ultrasound	COVID-19 diagnosis	https://tinyurl.com/yckfqrcg
[191]	Segmented CT scans	COVID-19 infected area segmentation	NA
[192]	Segmented CT scans	COVID-19 infected area segmentation	http://medicalsegmentation.com/COVID-19/
[179]	X-ray imagery	COVID-19 diagnosis	https://www.kaggle.com/tawsifurrahman/COVID-19-radiography-database

6. Challenges, limitations and outlook

Apart from the recent developments and promising performance deep CNNs has achieved in prognosis COVID-19, the research literature indicates several vital challenges and problems to be explored, some of which are described below.

6.1. Architecture, modelling and algorithms

- Theoretical understanding is lagging to define which architecture should perform better than others. The various architectures are defined by the context from one study to another and so it is a challenge to define general architecture from one domain to another in terms of parameters of each layer.
- It is difficult to inform the choice of architecture with the statistics of the data.
- Computational expense to train the models.
- Training a noise-tolerant model remains a challenge.
- Determination of hyperparameters and fine-tuning the model is a big challenge. Retraining models to track changes in data distribution is challenging.
- Scaling computations, reducing the overhead of optimizing parameters, avoiding expensive inference and sampling.
- In the absence of labeled data, selecting features in an unsupervised way is a very challenging, although essential task.
- Constructing optimum codebooks and low-complexity high-accuracy coding and decoding algorithms is demanding.
- Rescaling of input images to certain dimensions is needed when pre-trained networks are employed as feature extractors which may discard valuable information.

6.2. Dataset

- Obtaining and maintaining large-scale datasets to improve the efficiency of learning models is a challenge. Fewer data leads to overfitting and more data leads to time complexity to train the model infeasible amount of time.
- Datasets not only are essential for a fair comparison of different algorithms but also bring more challenges and complexity through their expansion and improvement.
- Handling high dimensional datasets is a challenging task.
- Big data and data modalities bring technical challenges.
- Limited availability of COVID-19 imagery datasets restricts the performance metrics of most of the AI/ML-based models. Some of the most common and open source datasets compiled by research work [184–186] included in this paper are listed in Table 2.
- Although data augmentation helps to develop more training data but developing generalizable augmentation policies for data augmentation is still a challenge.
- Mostly chest X-ray radiography is used for COVID-19 research purposes due to its low cost, low radiation, and wider accessibility, however, it is less sensitive than CT scan imagery. CT scan, magnetic resonance imaging (MRI), and ultrasound imaging need to be explored to make a well-annotated dataset.

6.3. Classification and detection

- Selecting a worthy classifier, resampling algorithm, feature representation, fusion strategies, labels, etc. is a big challenge for a particular task like classification or detection.
- Large-scale classification is hard and obtaining class-labels are expensive.
- Achieving efficiency in both feature extraction and classifier training without compromising performance is a challenge.
- Training of traditional models over large-scale datasets like large ImageNet classification is difficult and leads to poor and slow convergence.
- Fine-grained image classification is a big challenge.
- Image classification of non-rigid instances is also a challenge.
- Intra-class variation in appearance, computational complexity with a huge number of object categories, lighting backgrounds, deformation, non-rigid instances, large range of scales, and aspect ratios are some of the main challenges in object detection.
- Weakly supervised, multi-domain and 3D object detection are still challenging problems in the detection domain.
- Adopting specific pre-processing strategies with or without GPU acceleration can be used to increase performance metrics.
- Apart from X-ray and CT scan imagery, magnetic resonance imaging (MRI) and ultrasound imaging can be explored to make potential usage of novel approaches of deep CNNs in the prediction and diagnosis of COVID-19 infection.
- If more laboratory parameters are collected, ML algorithms may extract more valuable candidate makers to identify COVID-19.
- Age, gender, genetics, medical history of patients (chronic liver, kidney, diabetes, etc.), geolocation data can be utilized to boost performance metrics of diagnosis, classification, detection, prediction, spread, and forecasting of COVID-19.
- To detect and trace transmissibility from asymptomatic individuals of COVID-19 is still a challenge.
- Although most of the models are up to the mark to tackle the pandemic, but still there is no recognition from the research association of medical proficiency for COVID-19 positive case detection from radiology imagery using a deep learning framework.

7. Conclusion

This paper presents the deeper insight of deep convolutional neural networks with their latest developments and adopts a categorization scheme to analyze the existing literature. Some widely used techniques and variants of CNN are investigated. More specifically, state-of-the-art approaches of convolutional neural network are analyzed, expounded, and illustrated in detail because it is the most extensively utilized technique for radiographic visual imagery applications. Selected applications of deep CNN to COVID-19 are also highlighted. Its applications related to COVID-19 like image classification based on different classes of Pneumonia, recognition, and detection of different classes of coronavirus', instance segmentation of the region of interest (infected lung) are also explored. The pipeline of the general architecture for COVID-19 prognosis has also been proposed and illustrated. Finally, some important challenges, limitations, and outlooks are also discussed for better design, modeling, and training of learning modules along with several directions that may be further explored in the future.

Conflict of interest

The authors declare no conflict of interest.

References

[1] A. Borji, Negative results in computer vision: a perspective, Image Vis. Comput. 69 (2018) 1–8.

[2] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, M.S. Lew, Deep learning for visual understanding: a review, Neurocomputing 187 (2016) 27–48.

[3] World Health Organization Updates: Coronavirus Disease (Covid-19) – Events as They Happen, 2019. http://www.who.int/emergencies/diseases/novel-coronavirus-2019/events-as-they-happen.

[4] F. Rustam, A.A. Reshi, A. Mehmood, S. Ullah, B. On, W. Aslam, G.S. Choi, Covid-19 future forecasting using supervised machine learning models, IEEE Access (2020).

[5] A.M. Ismael, A. Şengür, Deep learning approaches for covid-19 detection based on chest *x*-ray images, Expert Syst. Appl. 164 (2020) 114054.

[6] L. Sun, G. Liu, F. Song, N. Shi, F. Liu, S. Li, P. Li, W. Zhang, X. Jiang, Y. Zhang, et al., Combination of four clinical indicators predicts the severe/critical symptom of patients infected covid-19, J. Clin. Virol. (2020) 104431.

[7] S. rekha Hanumanthu, Role of intelligent computing in covid-19 prognosis: a state-of-the-art review, Chaos Solitons Fractals (2020) 109947.

[8] J. Wu, P. Zhang, L. Zhang, W. Meng, J. Li, C. Tong, Y. Li, J. Cai, Z. Yang, J. Zhu, et al., Rapid and accurate identification of covid-19 infection through machine learning based on clinical available blood test results, medRxiv (2020).

[9] A.M.U.D. Khanday, S.T. Rabani, Q.R. Khan, N. Rouf, M.M.U. Din, Machine learning based approaches for detecting covid-19 using clinical text data, Int. J. Inf. Technol. 12 (3) (2020) 731–739.

[10] M.F. Aslan, M.F. Unlersen, K. Sabanci, A. Durdu, Cnn-based transfer learning-bilstm network: a novel approach for covid-19 infection detection, Appl. Soft Comput. (2020) 106912.

[11] L. Yan, H.-T. Zhang, J. Goncalves, Y. Xiao, M. Wang, Y. Guo, C. Sun, X. Tang, L. Jing, M. Zhang, et al., An interpretable mortality prediction model for covid-19 patients, Nat. Mach. Intell. (2020) 1–6.

[12] M. Polsinelli, L. Cinque, G. Placidi, A Light CNN for Detecting Covid-19 From CT Scans of the Chest, 2020 (arXiv preprint), arXiv:2004.12837.

[13] T. Ozturk, M. Talo, E.A. Yildirim, U.B. Baloglu, O. Yildirim, U.R. Acharya, Automated detection of covid-19 cases using deep neural networks with X-ray images, Comput. Biol. Med. (2020) 103792.

[14] A.A. Ardakani, A.R. Kanafi, U.R. Acharya, N. Khadem, A. Mohammadi, Application of deep learning technique to manage covid-19 in routine clinical practice using ct images: results of 10 convolutional neural networks, Comput. Biol. Med. (2020) 103795.

[15] C. Ouchicha, O. Ammor, M. Meknassi, Cvdnet: a novel deep learning architecture for detection of coronavirus (covid-19) from chest *x*-ray images, Chaos Solitons Fractals 140 (2020) 110245.

[16] K.H. Shibly, S.K. Dey, M.T.U. Islam, M.M. Rahman, Covid faster r-cnn: a novel framework to diagnose novel coronavirus disease (covid-19) in *x*-ray images, medRxiv (2020).

[17] D.H. Hubel, T.N. Wiesel, Receptive fields of single neurones in the cat's striate cortex, J. Physiol. 148 (3) (1959) 574–591.

[18] Y. Le Cun, O. Matan, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L. Jacket, H.S. Baird, Handwritten zip code recognition with multilayer networks, in: 1990] Proceedings. 10th International Conference on Pattern Recognition, vol. 2, IEEE, 1990, pp. 35–40.

[19] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L. D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (1989) 541–551.

[20] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al., Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[21] K. Fukushima, Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biol. Cybern. 36 (4) (1980) 193–202.

[22] X.-X. Niu, C.Y. Suen, A novel hybrid cnn-svm classifier for recognizing handwritten digits, Pattern Recognit. 45 (4) (2012) 1318–1325.

[23] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014 (arXiv preprint), arXiv:1409.1556.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015) 1–9.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.

[26] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[27] M.Z. Alom, T.M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M.S. Nasrin, B.C. Van Esesn, A.A.S. Awwal, V.K. Asari, The History Began from alexnet: A Comprehensive Survey on Deep Learning Approaches, 2018 (arXiv preprint), arXiv:1803.01164.

[28] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2014) 580–587.

[29] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (2014) 1725–1732.

[30] C. Dong, C.C. Loy, K. He, X. Tang, Learning a deep convolutional network for image super-resolution, European Conference on Computer Vision (2014) 184–199.

[31] N. Wang, D.-Y. Yeung, Learning a deep compact image representation for visual tracking. Advances in Neural Information Processing Systems, 2013, pp. 809–817.

[32] A. Toshev, C. Szegedy, Deeppose: human pose estimation via deep neural networks, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2014) 1653–1660.

[33] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015) 3431–3440.

[34] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems, 2015, pp. 91–99.

[35] M. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: Processing of European Conference on Computer Vision, Zurich, Switzerland, 5–12 September, 2014, 2014.

[36] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016) 770–778.

[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016) 2818–2826.

[38] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018) 7132–7141.

[39] D. Eigen, J. Rolfe, R. Fergus, Y. LeCun, Understanding Deep Architectures Using a Recursive Convolutional Network, 2013 (arXiv preprint), arXiv:1312.1847.

[40] K. Jarrett, K. Kavukcuoglu, M. Ranzato, Y. LeCun, What is the best multi-stage architecture for object recognition?, in: 2009 IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 2146–2153.

[41] J. Masci, U. Meier, D. Cireşan, J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction, International Conference on Artificial Neural Networks (2011) 52–59.

[42] G. Desjardins, Y. Bengio, Empirical Evaluation of Convolutional RBMS for Vision, DIRO, Université de Montréal, 2008, pp. 1–13.

[43] A. Krizhevsky, G. Hinton, Convolutional Deep Belief Networks on Cifar-10, vol. 40, 7, 2010, pp. 1–9 (Unpublished manuscript).

[44] M. Mathieu, M. Henaff, Y. LeCun, Fast Training of Convolutional Networks Through FFTS, 2013 (arXiv preprint), arXiv:1312.5851.

[45] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, E. Shelhamer, CUDNN: Efficient Primitives for Deep Learning, 2014 (arXiv preprint), arXiv:1410.0759.

[46] N. Vasilache, J. Johnson, M. Mathieu, S. Chintala, S. Piantino, Y. LeCun, Fast Convolutional Nets With FBFFT: A GPU Performance Evaluation, 2014 (arXiv preprint), arXiv:1412.7580.

[47] Z. Peng, L. Wu, J. Ren, R. Zhang, P. Luo, Cuimage: a neverending learning platform on a convolutional knowledge graph of billion web images, in: 2018 IEEE International Conference on Big Data (Big Data), IEEE, 2018, pp. 1787–1796.

[48] Y. LeCun, Learning invariant feature hierarchies, European Conference on Computer Vision (2012) 496–505.

[49] R. Goroshin, Y. LeCun, Saturating Auto-Encoders, 2013 (arXiv preprint), arXiv: 1301.3577.

[50] M.D. Zeiler, Hierarchical Convolutional Deep Learning in Computer Vision, New York University, 2013. PhD Thesis.

[51] J. Ngiam, Z. Chen, D. Chia, P.W. Koh, Q.V. Le, A.Y. Ng, Tiled convolutional neural networks. Advances in Neural Information Processing Systems, 2010, pp. 1279–1287.

[52] F. Visin, K. Kastner, A. Courville, Y. Bengio, M. Matteucci, K. Cho, Reseg: A Recurrent Neural Network for Object Segmentation, 2015 (arXiv preprint), arXiv: 1511.07053.

[53] M.D. Zeiler, D. Krishnan, G.W. Taylor, R. Fergus, Deconvolutional networks, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 2528–2535.

[54] M.D. Zeiler, G.W. Taylor, R. Fergus, et al., Adaptive deconvolutional networks for mid and high level feature learning, ICCV 1 (2011) 6.

[55] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015) 3431–3440.

[56] C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu, et al., Look and think twice: capturing top-down visual attention with feedback convolutional neural networks, Proceedings of the IEEE International Conference on Computer Vision (2015) 2956–2964.

[57] Y. Zhang, K. Lee, H. Lee, Augmenting supervised neural networks with unsupervised objectives for large-scale image classification, International Conference on Machine Learning (2016) 612–621.

[58] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2) (2015) 295–307.

[59] H. Noh, S. Hong, B. Han, Learning deconvolution network for semantic segmentation, Proceedings of the IEEE International Conference on Computer Vision (2015) 1520–1528.

[60] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016) 2921–2929.

[61] F. Yu, V. Koltun, Multi-Scale Context Aggregation by Dilated Convolutions, 2015 (arXiv preprint), arXiv:1511.07122.

[62] A.v.d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, Wavenet: A Generative Model for Raw Audio, 2016 (arXiv preprint), arXiv:1609.03499.

[63] T. Sercu, V. Goel, Dense Prediction on Sequences With Time-Dilated Convolutions for Speech Recognition, 2016 (arXiv preprint), arXiv:1611.09288.

[64] N. Kalchbrenner, L. Espeholt, K. Simonyan, A.v.d. Oord, A. Graves, K. Kavukcuoglu, Neural Machine Translation in Linear Time, 2016 (arXiv preprint), arXiv:1610.10099.

[65] M. Lin, Q. Chen, S. Yan, Network in Network, 2013 (arXiv preprint), arXiv:1312.4400.

[66] D. Erhan, C. Szegedy, A. Toshev, D. Anguelov, Scalable object detection using deep neural networks, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2014) 2147–2154.

[67] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al., Recent advances in convolutional neural networks, Pattern Recognit. 77 (2018) 354–377.

[68] Y.-L. Boureau, J. Ponce, Y. LeCun, A theoretical analysis of feature pooling in visual recognition, Proceedings of the 27th International Conference on Machine Learning (ICML-10) (2010) 111–118.

[69] D. Scherer, A. Müller, S. Behnke, Evaluation of pooling operations in convolutional architectures for object recognition, International Conference on Artificial Neural Networks (2010) 92–101.

[70] D.C. Cireşan, U. Meier, J. Masci, L.M. Gambardella, J. Schmidhuber, High-Performance Neural Networks for Visual Object Classification, 2011 (arXiv preprint), arXiv:1102.0183.

[71] A. Hyvärinen, U. Köster, Complex cell pooling and the statistics of natural images, Netw. Comput. Neural Syst. 18 (2) (2007) 81–100.

[72] J. Bruna, A. Szlam, Y. LeCun, Signal Recovery From Pooling Representations, 2013 (arXiv preprint), arXiv:1311.4025.

[73] C. Gulcehre, K. Cho, R. Pascanu, Y. Bengio, Learned-norm pooling for deep feedforward and recurrent neural networks, Joint European Conference on Machine Learning and Knowledge Discovery in Databases (2014) 530–546.

[74] M.D. Zeiler, R. Fergus, Stochastic Pooling for Regularization of Deep Convolutional Neural Networks, 2013 (arXiv preprint), arXiv:1301.3557.

[75] O. Rippel, J. Snoek, R.P. Adams, Spectral representations for convolutional neural networks. Advances in Neural Information Processing Systems, 2015, pp. 2449–2457.

[76] H. Zhang, J. Ma, Hartley Spectral Pooling for Deep Learning, 2018 (arXiv preprint), arXiv:1810.04028.

[77] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1904–1916.

[78] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.

[79] K. Chatfield, V.S. Lempitsky, A. Vedaldi, A. Zisserman, The devil is in the details: an evaluation of recent feature encoding methods, BMVC 2 (2011) 8.

[80] W. Ouyang, P. Luo, X. Zeng, S. Qiu, Y. Tian, H. Li, S. Yang, Z. Wang, Y. Xiong, C. Qian, et al., Deepid-net: Multi-Stage and Deformable Deep Convolutional Neural Networks for Object Detection, 2014 (arXiv preprint), arXiv:1409.3505.

[81] A. Ruderman, N.C. Rabinowitz, A.S. Morcos, D. Zoran, Learned Deformation Stability in Convolutional Neural Networks, 2018. CoRR, abs/1804.04438.

[82] Q. Xu, M. Zhang, Z. Gu, G. Pan, Overfitting remedy by sparsifying regularization on fully-connected layers of cnns, Neurocomputing 328 (2019) 69–74.

[83] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2014) 1717–1724.

[84] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, Proceedings of the 27th International Conference on Machine Learning (ICML-10) (2010) 807–814.

[85] P. Ramachandran, B. Zoph, Q.V. Le, Searching for Activation Functions, 2017 (arXiv preprint), arXiv:1710.05941.

[86] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, Proceedings of the 27th International Conference on Machine Learning (ICML-10) (2010) 807–814.

[87] A.F. Agarap, Deep Learning Using Rectified Linear Units (Relu), 2018 (arXiv preprint), arXiv:1803.08375.

[88] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., Mastering the game of go with deep neural networks and tree search, Nature 529 (7587) (2016) 484.

[89] A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, Proc. ICML 30 (2013) 3.

[90] M.D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q.V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, et al., On rectified linear units for speech processing, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2013, pp. 3517–3521.

[91] L. Lu, Y. Shin, Y. Su, G.E. Karniadakis, Dying Relu and Initialization: Theory and Numerical Examples, 2019 (arXiv preprint), arXiv:1903.06733.

[92] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on imagenet classification. Proceedings of the IEEE International Conference on Computer Vision (2015) 1026–1034.

[93] B. Xu, N. Wang, T. Chen, M. Li, Empirical Evaluation of Rectified Activations in Convolutional Network, 2015 (arXiv preprint), arXiv:1505.00853.

[94] K. Jayakumar, K.K. Sankaran, On a generalisation of uniform distribution and its properties, Statistica 76 (1) (2016) 83–91.

[95] J. Rosenblatt, M.K. Roychowdhury, Uniform Distributions on Curves and Optimal Quantization, 2018 (arXiv preprint), arXiv:1809.08364.

[96] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[97] D.-A. Clevert, T. Unterthiner, S. Hochreiter, Fast and Accurate Deep Network Learning by Exponential Linear Units (elus), 2015 (arXiv preprint), arXiv: 1511.07289.

[98] L. Trottier, P. Gigu, B. Chaib-draa, et al., Parametric exponential linear unit for deep convolutional neural networks, in: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2017, pp. 207–214.

[99] I.J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, Y. Bengio, Maxout Networks, 2013 (arXiv preprint), arXiv:1302.4389.

[100] J.T. Springenberg, M. Riedmiller, Improving Deep Neural Networks With Probabilistic Maxout Units, 2013 (arXiv preprint), arXiv:1312.6116.

[101] A. Nicolson, K.K. Paliwal, Deep learning for minimum mean-square error approaches to speech enhancement, Speech Commun. (2019).

[102] V. Vapnik, The Nature of Statistical Learning Theory, Springer Science & Business Media, 2013.

[103] V.N. Vapnik, An overview of statistical learning theory, IEEE Trans. Neural Netw. 10 (5) (1999) 988–999.

[104] T. Zhang, Solving large scale linear prediction problems using stochastic gradient descent algorithms, Proceedings of the Twenty-First International Conference on Machine Learning (2004) 116.

[105] Y. Tang, Deep Learning Using Linear Support Vector Machines, 2013 (arXiv preprint), arXiv:1306.0239.

[106] L. Deng, The mnist database of handwritten digit images for machine learning research [best of the web], IEEE Signal Process. Mag. 29 (6) (2012) 141–142.

[107] H.-J. Xing, M. Ji, Robust one-class support vector machine with rescaled hinge loss function, Pattern Recognit. 84 (2018) 152–164.

[108] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, Proceedings of the 25th International Conference on Machine Learning (2008) 1096–1103.

[109] W.W. Ng, G. Zeng, J. Zhang, D.S. Yeung, W. Pedrycz, Dual autoencoders features for imbalance classification problem, Pattern Recognit. 60 (2016) 875–889.

[110] J. Mehta, A. Majumdar, Rodeo: robust de-aliasing autoencoder for real-time medical image reconstruction, Pattern Recognit. 63 (2017) 499–510.

[111] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets. Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.

[112] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, H. Lee, Generative Adversarial Text to Image Synthesis, 2016 (arXiv preprint), arXiv:1605.05396.

[113] E.L. Denton, S. Chintala, R. Fergus, et al., Deep generative image models using a laplacian pyramid of adversarial networks.. Advances in Neural Information Processing Systems, 2015, pp. 1486–1494.

[114] W. Liu, Y. Wen, Z. Yu, M. Yang, Large-margin softmax loss for convolutional neural networks, ICML 2 (2016) 7.

[115] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, R. Shah, Signature verification using a "siamese" time delay neural network. Advances in Neural Information Processing Systems, 1994, pp. 737–744.

[116] S. Chopra, R. Hadsell, Y. LeCun, et al., Learning a similarity metric discriminatively, with application to face verification, CVPR (1) (2005) 539–546.

[117] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2, IEEE, 2006, pp. 1735–1742.

[118] U. Shaham, R.R. Lederman, Learning by coincidence: Siamese networks and common variable learning, Pattern Recognit. 74 (2018) 52–63.

[119] J. Lin, O. Morère, A. Veillard, L.-Y. Duan, H. Goh, V. Chandrasekhar, Deephash for image instance retrieval: getting regularization, depth and fine-tuning right, Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval (2017) 133–141.

[120] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: a unified embedding for face recognition and clustering, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015) 815–823.

[121] H. Liu, Y. Tian, Y. Yang, L. Pang, T. Huang, Deep relative distance learning: tell the difference between similar vehicles, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016) 2167–2175.

[122] Q. Zhu, P. Zhang, X. Ye, A New Loss Function for cnn Classifier Based on Pre-Defined Evenly-Distributed Class Centroids, 2019 (arXiv preprint), arXiv:1904.06008.

[123] B. Jiang, Y.-F. Liu, Z. Wen, L_p-norm regularization algorithms for optimization over permutation matrices, SIAM J. Optim. 26 (4) (2016) 2284–2313.

[124] A. Cui, J. Peng, H. Li, Modified Lp-norm Regularization Minimization for Sparse Signal Recovery, 2018 (arXiv preprint), arXiv:1801.09172.

[125] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors, 2012 (arXiv preprint), arXiv:1207.0580.

[126] P. Baldi, P.J. Sadowski, Understanding dropout.. Advances in Neural Information Processing Systems, 2013, pp. 2814–2822.

[127] S. Wang, C. Manning, Fast dropout training, International Conference on Machine Learning (2013) 118–126.

[128] J. Ba, B. Frey, Adaptive dropout for training deep neural networks. Advances in Neural Information Processing Systems, 2013, pp. 3084–3092.

[129] D. McAllester, A Pac-Bayesian Tutorial With a Dropout Bound, 2013 (arXiv preprint), arXiv:1307.2118.

[130] S. Wager, S. Wang, P.S. Liang, Dropout training as adaptive regularization. Advances in Neural Information Processing Systems, 2013, pp. 351–359.

[131] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[132] D. Warde-Farley, I.J. Goodfellow, A. Courville, Y. Bengio, An Empirical Analysis of Dropout in Piecewise Linear Networks, 2013 (arXiv preprint), arXiv:1312.6197.

[133] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, C. Bregler, Efficient object localization using convolutional networks, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015) 648–656.

[134] G. Kang, J. Li, D. Tao, Shakeout: a new approach to regularized deep neural network training, IEEE Trans. Pattern Anal. Mach. Intell. 40 (5) (2017) 1245–1258.

[135] B.A. Olshausen, D.J. Field, Sparse coding with an overcomplete basis set: a strategy employed by v1? Vis. Res. 37 (23) (1997) 3311–3325.

[136] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (1) (2019) 60.

[137] H. Yang, I. Patras, Mirror, mirror on the wall, tell me, is the error small? Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015) 4685–4693.

[138] J. Salamon, J.P. Bello, Deep convolutional neural networks and data augmentation for environmental sound classification, IEEE Signal Process. Lett. 24 (3) (2017) 279–283.

[139] S. Xie, Z. Tu, Holistically-nested edge detection, Proceedings of the IEEE International Conference on Computer Vision (2015) 1395–1403.

[140] D. Eigen, R. Fergus, Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture, Proceedings of the IEEE International Conference on Computer Vision (2015) 2650–2658.

[141] M. Paulin, J. Revaud, Z. Harchaoui, F. Perronnin, C. Schmid, Transformation pursuit for image classification, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2014) 3646–3653.

[142] A. Dosovitskiy, J.T. Springenberg, T. Brox, Unsupervised Feature Learning by Augmenting Single Images, 2013 (arXiv preprint), arXiv:1312.5242.

[143] S. Xie, T. Yang, X. Wang, Y. Lin, Hyper-class augmented and regularized deep learning for fine-grained image classification, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015) 2645–2654.

[144] Z. Xu, S. Huang, Y. Zhang, D. Tao, Augmenting strong supervision using web data for fine-grained categorization, Proceedings of the IEEE International Conference on Computer Vision (2015) 2524–2532.

[145] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Netw. 61 (2015) 85–117.

[146] G.E. Hinton, T.J. Sejnowski, et al., Learning and relearning in boltzmann machines, Parallel Distrib. Process. Explor. Microstruct. Cognit. 1 (282–317) (1986) 2.

[147] G.E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. 14 (8) (2002) 1771–1800.

[148] D.H. Ackley, G.E. Hinton, T.J. Sejnowski, A learning algorithm for boltzmann machines, Cognit. Sci. 9 (1) (1985) 147–169.

[149] N. Zhang, S. Ding, J. Zhang, Y. Xue, An overview on restricted Boltzmann machines, Neurocomputing 275 (2018) 1186–1199.

[150] V. Upadhya, P. Sastry, An overview of restricted Boltzmann machines, J. Indian Inst. Sci. (2019) 1–12.

[151] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[152] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks. Advances in Neural Information Processing Systems, 2007, pp. 153–160.

[153] R. Salakhutdinov, G. Hinton, Deep boltzmann machines.. Artificial Intelligence and Statistics, 2009, pp. 448–455.

[154] J. Ngiam, Z. Chen, P.W. Koh, A.Y. Ng, Learning Deep Energy Models, 2011.

[155] C.-Y. Liou, W.-C. Cheng, J.-W. Liou, D.-R. Liou, Autoencoder for words, Neurocomputing 139 (2014) 84–96.

[156] G.E. Hinton, R.S. Zemel, Autoencoders, minimum description length and helmholtz free energy. Advances in Neural Information Processing Systems, 1994, pp. 3–10.

[157] Z.C. Lipton, J. Berkowitz, C. Elkan, A Critical Review of Recurrent Neural Networks for Sequence Learning, 2015 (arXiv preprint), arXiv:1506.00019.

[158] H. Lee, A. Battle, R. Raina, A.Y. Ng, Efficient sparse coding algorithms. Advances in Neural Information Processing Systems, 2007, pp. 801–808.

[159] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, G. Zhang, Transfer learning using computational intelligence: a survey, Knowl.-Based Syst. 80 (2015) 14–23.

[160] S. Lalmuanawma, J. Hussain, L. Chhakchhuak, Applications of machine learning and artificial intelligence for covid-19 (sars-cov-2) pandemic: a review, Chaos Solitons Fractals (2020) 110059.

[161] Radiological Society of North America: RSNA Pneumonia Detection Challenge, 2018. https://www.kaggle.com/c/rsna-pneumonia-detection-challenge.

[162] Y. Song, S. Zheng, L. Li, X. Zhang, X. Zhang, Z. Huang, J. Chen, H. Zhao, Y. Jie, R. Wang, et al., Deep learning enables accurate diagnosis of novel coronavirus (covid-19) with ct images, medRxiv (2020).

[163] S. Wang, B. Kang, J. Ma, X. Zeng, M. Xiao, J. Guo, M. Cai, J. Yang, Y. Li, X. Meng, et al., A deep learning algorithm using ct images to screen for corona virus disease (covid-19), MedRxiv (2020).

[164] X. Xu, X. Jiang, C. Ma, P. Du, X. Li, S. Lv, L. Yu, Q. Ni, Y. Chen, J. Su, et al., A deep learning system to screen novel coronavirus disease 2019 pneumonia, Engineering (2020).

[165] C. Zheng, X. Deng, Q. Fu, Q. Zhou, J. Feng, H. Ma, W. Liu, X. Wang, Deep learning-based detection for covid-19 from chest ct using weak label, MedRxiv (2020).

[166] G. Maguolo, L. Nanni, A Critic Evaluation of Methods for Covid-19 Automatic Detection from X-ray Images, 2020 (arXiv preprint), arXiv:2004.12823.

[167] L. Wynants, B. Van Calster, G.S. Collins, R.D. Riley, G. Heinze, E. Schuit, M. M. Bonten, D.L. Dahly, J.A. Damen, T.P. Debray, et al., Prediction models for diagnosis and prognosis of covid-19: systematic review and critical appraisal, BMJ 369 (2020).

[168] L. Li, L. Qin, Z. Xu, Y. Yin, X. Wang, B. Kong, J. Bai, Y. Lu, Z. Fang, Q. Song, et al., Using artificial intelligence to detect covid-19 and community-acquired pneumonia based on pulmonary ct: evaluation of the diagnostic accuracy, Radiology 296 (2) (2020).

[169] S.H. Kassani, P.H. Kassasni, M.J. Wesolowski, K.A. Schneider, R. Deters, Automatic Detection of Coronavirus Disease (COVID-19) in X-ray and CT Images: A Machine Learning-Based Approach, 2020 (arXiv preprint), arXiv:2004.10641.

[170] A. Narin, C. Kaya, Z. Pamuk, Automatic Detection of Coronavirus Disease (COVID-19) Using X-ray Images and Deep Convolutional Neural Networks, 2020 (arXiv preprint), arXiv:2003.10849.

[171] A. Gupta, S. Gupta, R. Katarya, et al., Instacovnet-19: a deep learning classification model for the detection of covid-19 patients using chest *x*-ray, Appl. Soft Comput. (2020) 106859.

[172] F. Ucar, D. Korkmaz, Covidiagnosis-net: deep bayes-squeezenet based diagnostic of the coronavirus disease 2019 (covid-19) from *x*-ray images, Med. Hypotheses (2020) 109761.

[173] A.I. Khan, J.L. Shah, M.M. Bhat, Coronet: a deep neural network for detection and diagnosis of covid-19 from chest X-ray images, Comput. Methods Programs Biomed. (2020) 105581.

[174] L. Wang, Z.Q. Lin, A. Wong, Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest X-ray images, Sci. Rep. 10 (1) (2020) 1–12.

[175] P. Afshar, S. Heidarian, F. Naderkhani, A. Oikonomou, K.N. Plataniotis, A. Mohammadi, Covid-Caps: A Capsule Network-Based Framework for Identification of COVID-19 Cases From X-ray Images, 2020 (arXiv preprint), arXiv:2004.02696.

[176] I.D. Apostolopoulos, T.A. Mpesiana, Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks, Phys. Eng. Sci. Med. (2020) 1.

[177] M. Huh, P. Agrawal, A.A. Efros, What Makes Imagenet Good for Transfer Learning?, 2016 (arXiv preprint), arXiv:1608.08614.

[178] M.A. Shah, N.Y. Khanday, M. Purohit, M. Gulzar, Enhancement and Segmentation of Lung CT Images for Efficient Identification of Cancerous Cells, 2016.

[179] M.E. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M.A. Kadir, Z.B. Mahbub, K.R. Islam, M.S. Khan, A. Iqbal, N. Al-Emadi, et al., Can AI Help in Screening Viral and COVID-19 Pneumonia?, 2020 (arXiv preprint), arXiv:2003.13145.

[180] M. Farooq, A. Hafeez, Covid-resnet: A Deep Learning Framework for Screening of COVID19 From Radiographs, 2020 (arXiv preprint), arXiv:2003.14395.

[181] E.E.-D. Hemdan, M.A. Shouman, M.E. Karar, Covidx-net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-ray Images, 2020 (arXiv preprint), arXiv:2003.11055.

[182] C. Ouchicha, O. Ammor, M. Meknassi, Cvdnet: a novel deep learning architecture for detection of coronavirus (covid-19) from chest X-ray images, Chaos Solitons Fractals 140 (2020) 110245.

[183] A. Shalbaf, M. Vafaeezadeh, et al., Automated detection of covid-19 using ensemble of transfer learning with deep convolutional neural network based on ct scans, Int. J. Comput. Assist. Radiol. Surg. (2020) 1–9.

[184] J. Shuja, E. Alanazi, W. Alasmary, A. Alashaikh, Covid-19 open source data sets: a comprehensive survey, Appl. Intell. (2020) 1–30.

[185] J. Shuja, E. Alanazi, W. Alasmary, A. Alashaikh, Covid-19 datasets: a survey and future challenges, MedRxiv (2020).

[186] J.P. Cohen, P. Morrison, L. Dao, K. Roth, T.Q. Duong, M. Ghassemi, COVID-19 Image Data Collection: Prospective Predictions are the Future, 2020 (arXiv preprint), arXiv:2006.11988.

[187] J. Zhao, Y. Zhang, X. He, P. Xie, COVID-CT-Dataset: A CT Scan Dataset About COVID-19, 2020 (arXiv preprint), arXiv:2003.13865.

[188] M. Jun, G. Cheng, W. Yixin, A. Xingle, G. Jiantao, Y. Ziqi, Z. Minqing, L. Xin, D. Xueyuan, C. Shucheng, et al., Covid-19 CT lung and infection segmentation dataset, Zenodo 20 (April) (2020).

[189] W. El-shafai, F. Abd el-samie, Extensive and Augmented COVID-19 X-ray and CT Chest Images Dataset, Mendeley Data, v1, 2020, https://doi.org/10.17632/8h65ywd2jr.1. https://data.mendeley.com/datasets/8h65ywd2jr/1.

[190] J. Born, G. Brändle, M. Cossio, M. Disdier, J. Goulet, J. Roulin, N. Wiedemann, Pocovid-net: Automatic Detection of COVID-19 From a New Lung Ultrasound Imaging Dataset (Pocus), 2020 (arXiv preprint), arXiv:2004.12084.

[191] F. Shan, Y. Gao, J. Wang, W. Shi, N. Shi, M. Han, Z. Xue, Y. Shi, Lung Infection Quantification of COVID-19 in CT Images With Deep Learning, 2020 (arXiv preprint), arXiv:2003.04655.

[192] COVID-19 CT Segmentation Dataset, 2020. http://medicalsegmentation.com/covid19/.

[193] N.Y. Khanday, S.A. Sofi, Taxonomy, state-of-the-art, challenges and applications of visual understanding: a review, Comput. Sci. Rev. 40 (2021) 100374. Elsevier.