

# COVID19 detection from Radiographs: Is Deep Learning able to handle the crisis?

Muhammad Saqib<sup>a</sup>, Saeed Anwar<sup>b,c</sup>, Abbas Anwar<sup>d</sup>, Lars Petersson<sup>b</sup>, Nabin Sharma<sup>a</sup>, Michael Blumenstein<sup>a</sup>

<sup>a</sup>University of Technology Sydney

<sup>b</sup>Data61-CSIRO, Australia

<sup>c</sup>Research School of Engineering, Australian National University, Australia

<sup>d</sup>University of Engineering and Technology, Peshawar, Pakistan.

## ABSTRACT

The COVID-19 is a highly contagious viral infection which played havoc on everyone's life in many different ways. According to the world health organization and scientists, more testing potentially helps governments and disease control organizations in containing the spread of the virus. The use of chest radiographs is one of the early screening tests to determine the onset of disease, as the infection affects the lungs severely. This study will investigate and automate the process of testing by using state-of-the-art CNN classifiers to detect the COVID19 infection. However, the viral could of many different types; therefore, we only regard for COVID19 while the other viral infection types are treated as non-COVID19 in the radiographs of various viral infections. The classification task is challenging due to the limited number of scans available for COVID19 and the minute variations in the viral infections. We aim to employ current state-of-the-art CNN architectures, compare their results, and determine whether deep learning algorithms can handle the crisis appropriately. All trained models are available at <https://github.com/saeed-anwar/COVID19-Baselines>

## 1. Introduction

The COVID-19 infection is caused by a virus known as SARS-C0v-2 or novel Coronavirus belongs to the Corona family. The virus is highly contagious, as is evident from the exponential growth of positive cases throughout the world in a short period with limited testing. The infection causes severe damage to the lungs causing pneumonia with accompanying symptoms of sore throat, dry coughing, sneezing, and high temperature. Moreover, some of the patients don't show symptoms acting as a carrier is a worrying concern for the health organization.

The World Health Organization (WHO) recommended con-

ducting more tests for screening out COVID-19 patients to contain the virus's spread. The testing will help in identifying and isolating potential COVID-19 patients from patients of other kinds of infections.

Generally, x-ray imaging is used for the majority of chest infections, such as pneumonia, bronchitis, and bronchiolitis. Although the use of X-Ray is considered non-specific in radiological findings, however, it will help in further management of a disease. The Reverse Transcription Polymerase Chain Reaction (RT-PCR) test kits are primarily used for the testing of COVID-19 patients. The test kits are expensive and also limited in supply. Moreover, the turnaround time for the test is, on average, 24 hours, which considerably slows the testing process. Since the test kits are limited in supply, the use of X-Ray can be viable options, especially in far-flung and rural areas. This

Corresponding authors:

*e-mail:* muhammad.saqib@uts.edu.au (Muhammad Saqib),

saeed.anwar@csiro.au (Saeed Anwar)

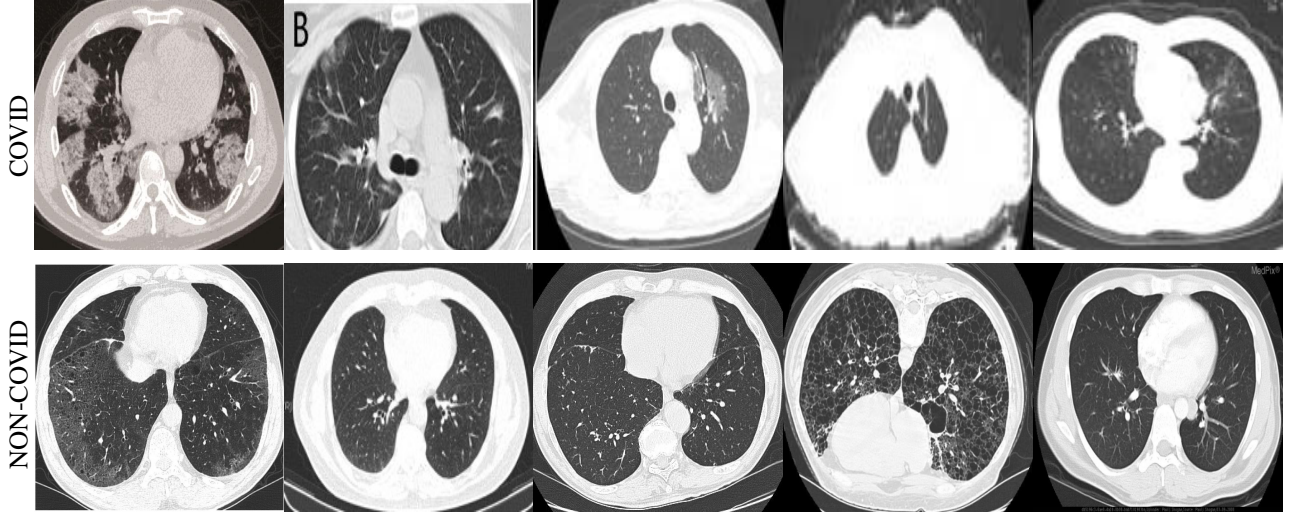


Fig. 1: Representative images of the COVIDCT dataset employed for training and evaluation of algorithms.

pandemic is causing health systems overwhelmed with the high number of patients. The use of chest X-rays could be used in conjunction with related tests to quickly rule out non-COVID patients, ultimately easing the burden on health systems. Like any other pneumonia infection, the COVID-19 has shown distinct markers on chest X-Rays.

This research investigates how state-of-the-art deep learning can capture fine-grained details from the images and classify them from other pneumonia and bacterial infections.

The remainder of the paper is organized as follows. In the following section 2, related work to Covid-19 using computer vision is discussed. The section 3 presents the methodology, and experimental results are discussed in section 4. The paper is finally concluded in section 5.

## 2. Related Work

Chen et al. (2020) retrospectively collected 46,096 high-quality CT images from 106 admitted patients anonymously. There were 51 laboratory-confirmed COVID-19 patients, while the rest of 55 were patients of other diseases. Three expert radiologists with more than five years of experience annotated the COVID-19 dataset with combined consensus. The problem is framed as a segmentation task, and UNet++ (Zhou et al., 2018) was trained to segment valid areas in the CT images. The

trained model was deployed at the Renmin Hospital of Wuhan University and as a web API to assist the diagnosis of COVID-19 cases around the world.

The COVNet (Li et al., 2020) uses 3D deep learning framework to extract 2D local and 3D features for the detection of COVID-19. The ResNet (He et al., 2016) is employed as the backbone to extract features from the input CT slices. The extracted features are passed through the max-pooling operation. The final feature map is fed through the fully-connected layer and eventually through the softmax function to get probabilities for each class. Previous studies have shown the successful application of deep learning methodologies to chest X-Rays for the diagnosis of bacterial and viral infections (Rajaraman et al., 2018; Kermany et al., 2018).

A deep learning-based CT diagnosis system, termed as DeepPneumonia, is proposed to detect and localize the lesions causing COVID-19 (Song et al., 2020). Firstly, the lung region is extracted in each CT image and then fed to the Details Relation Extraction neural Network (DRE-Net) to produce top K details in a CT scan using pre-trained ResNet with Feature Pyramid Network (FPN) and attention module. The attention module is used to learn the importance of each detail. The predictions are aggregated by the aggregation module to predict the patient-level diagnosis.

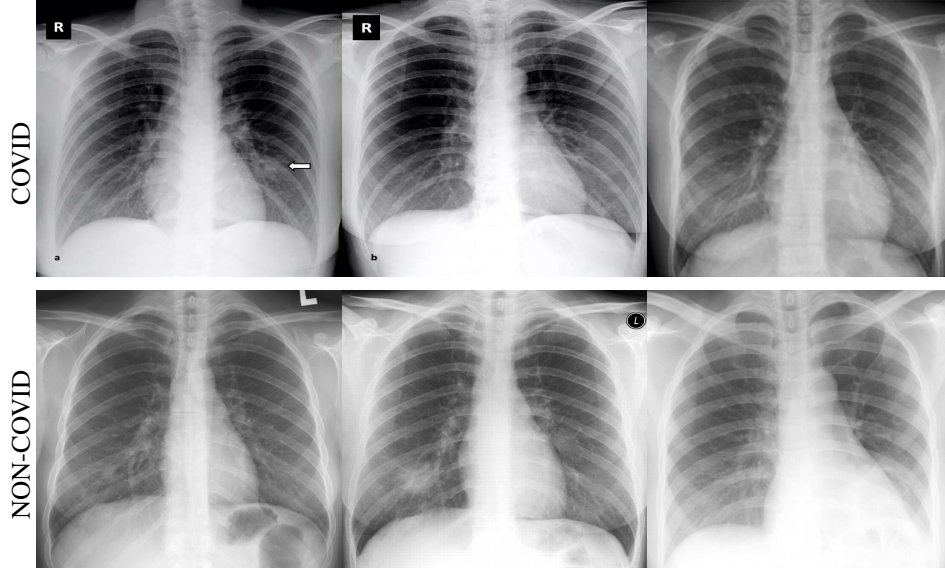


Fig. 2: Samples from the COVIDx dataset. The upper row shows the COVID19 infected examples, while the lower row presents the infection-free images.

Deep learning requires a significant amount of annotated data for training the model. As the radiologists are busy dealing with the pandemic, the annotation task is difficult and costly; therefore, a weakly-supervised technique is presented by [Zheng et al. \(2020\)](#) that utilizes weak patient-level labels for the rapid diagnosis of COVID-19 subjects. A 3D deep convolutional neural network called DeCoVNet ([Zheng et al., 2020](#)) is used to take input of CT volumes and 3D lung masks to output the probabilities of COVID-19 and non-COVID-19. A pre-trained model is used to generate a 3D lung mask. The first stage of the architecture consists of a vanilla 3D convolutional base, followed by batch normalization and max-pooling to create a 3D feature map. In the second stage, a 3D feature map is passed through two 3D residual blocks with the batch norm. In the last step, a Progressive Classifier (ProClf) progressively abstracts the information in 3D volumes and classifies using softmax function to output the probability of being COVID-19 and non-COVID-19.

The Chest x-ray radiography (CXR) is widely used for the diagnosis of various infections due to its lower cost and broader availability. The COVID-19 patients show lung consolidation over the period and, therefore, could be used as a diagnos-

tic tool in conjunction with a CT scan for better radiological analysis ([Ng et al., 2020](#)). A two-step human-machine collaborative strategy is proposed to design network architecture for the detection of COVID-19 cases from CXR images ([Wang and Wong, 2020](#)). In the first step, the initial network design prototype is constructed using human-driven principles. In the second step, initial prototype and human-specific designs are used in machine-driven exploration to find optimal macro architecture and microarchitecture for the final deep neural network architecture. The final architecture COVID-Net is applied for three-class classification into a) no infection b) non-COVID-19 and c) COVID-19 infection. The experimentation is carried out on a COVIDx dataset, curated from five multiple sources. [Farooq and Hafeez \(2020\)](#) proposed three-stage ResNet architecture to classify classes from the COVIDx dataset.

It is essential to estimate uncertainty in deep learning models and to avoid COVID-19 misdiagnoses, a Dropweights based Bayesian Convolutional Neural Networks (BCNN) ([Ghoshal and Tucker, 2020](#)) is proposed to deal with the uncertainty in deep learning. The experiments were carried out on the COVID-19 CXR dataset. The author found a strong relationship between uncertainty and accuracy, which helped identify

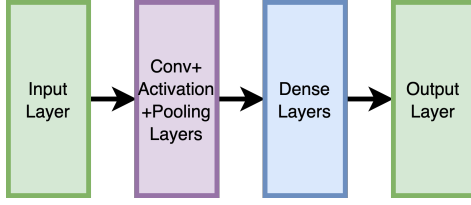


Fig. 3: Sample architecture of plain networks.

false positives/unknown cases.

We aim to employ the available deep learning state-of-the-art algorithms to identify the COVID19 and non-COVID19 features. The purpose is two-fold: 1) this research will provide baselines, and 2) it will also establish the performance of current state-of-the-art deep learning algorithms.

### 3. Methodology

For the sake of completeness, we will discuss the basic building blocks of the current state-of-the-art deep learning architectures for image classification tasks. The Convolutional Neural Networks (CNNs) are commonly used for image classification and can extract powerful, generic features from the image by applying convolution filters. In CNNs, the filter parameters are learned using backpropagation, low-level features such as edges are determined in the lower layers of the architecture, and high-level features such as shapes are discovered in the deeper layers of the network. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Deng et al., 2009) played a huge role in the design choices of the modern CNN architectures. The novel architectures should be end-to-end trainable and be capable of learning multiscale features with fewer parameters and smaller model size. The other design choices include dropouts, batch normalizations, optimizations, and loss functions, *etc.*

The modern architectures are broadly grouped into categories.

#### 3.1. Plain Networks

AlexNet Krizhevsky et al. (2012) is the first architecture that sparked the research interest in deep learning when it won the ImageNet challenge by a substantial margin. The architecture

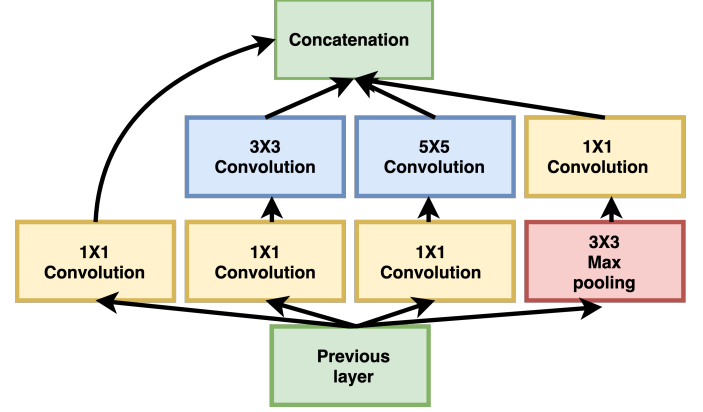


Fig. 4: Inception: Basic building block of GoogleNet.

consists of eight layers, including five convolutional layers, activation function, and three fully-connected layers. AlexNet, for the first time, used multi-GPUs to train bigger models and reduce training time.

Subsequently, Simonyan and Zisserman (2015) proposed Visual Geometry Group (VGG) that comes in different variations such as VGG16 and VGG19 are the most common architectures with 16 and 19 layers, respectively. The typical pattern among these architectures is the use of only  $3 \times 3$  filters. The initial layers utilize a few filters but increase their number as the depth of the network increase, the kind of pattern that can also be seen in other architectures. In earlier layers of VGG or other plain architectures, learn more spatial information for filters, while the later layers used more filters to balance out the availability of less spatial information. Initially, the VGG architecture was difficult to train from random initialization of weights. However, the training became easier with the introduction of intelligent initialization techniques such as Xavier (Glorot and Bengio, 2010) and (He et al., 2015). VGG19, the high accurate model, has a size of 574MB.

Contrary to the plain networks, the succeeding architectures share a common property *i.e.* using shortcut paths from earlier layers to later layers, which addresses the vanishing gradient problem (Hochreiter, 1998) in training deep neural networks.

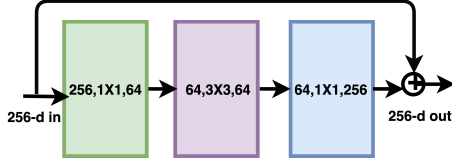


Fig. 5: Basic building block of Residual Network.

### 3.2. GoogleNet Inception Networks

GoogleNet introduced by Szegedy et al. (2015), was the winning architecture in the ImageNet challenge. The performance of GoogleNet was slightly better than VGG; however, the GoogleNet model was considerably smaller in size, with only 28.12MB as compared to 574MB of the VGG model.

The basic building block of the GoogleNet is termed as the Inception module, which comes in different variations making it more accurate than the original implementation of GoogleNet Inception. The idea of inception is to use filters of varying dimensions simultaneously, and then it is left to the network to decide during optimization which weights are essential. In this way, the network learns multiscale features efficiently.

The  $1 \times 1$  convolution was used to reduce the dimension of feature map volume-wise before applying any other filter, thus decreasing the model size significantly. The inception module, as shown in the figure 4, applies  $1 \times 1$  convolution in the first layer and max-pooling followed by any other filter. The output of all the filters is concatenated volume-wise before passing into the next layer of the network.

### 3.3. Residual Networks

The traditional network suffers from vanishing gradient problem (Hochreiter, 1998) during backpropagation. The gradient becomes very smaller and cannot update the parameters in the initial layers causing the network learning to be prolonged. The Residual Network (ResNet) made it possible to train deeper networks (He et al., 2016).

The basic module of ResNet is called Residual block, as shown in the figure 5, which starts at the input of the module with two branches. One of the branches takes the input through the series of convolutions, activations, and batch normalization

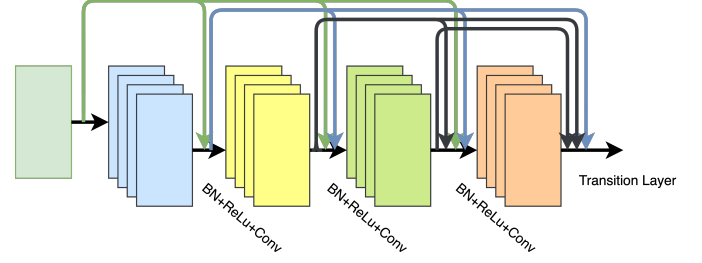


Fig. 6: The main module of DenseNet showing the concatenation of features from previous layers.

while the other branch is a shortcut that skips all the operations and is added to the output of the other branch, also known as identity-mapping. The residual layer starts learning at the identity function and learns more sophisticated and robust features going towards the depth of the architecture. In the recent version of ResNet, the order of operations in the first branch has been changed from convolution, activation, batch normalization (Conv-ReLU-BN) to batch normalization, activation, convolution (BN-ReLU-Conv). The method is called a pre-activation.

### 3.4. Dense Networks

In DenseNet (Huang et al., 2017), each layer concatenates the feature maps from all the previous layers, using the collective knowledge in the current feature map's computation. The current layer passes on its feature map to all the subsequent layers, which ensure maximum information flow and gradients between layers of the network.

On the contrary, ResNet adds the features from the module input to the output layer. Figure 6 shows the module of DenseNet; a composition layer uses pre-activation on all the previous layers before concatenating with the current layer. The DenseNet has fewer parameters and can learn more complex features.

### 3.5. Efficient Networks

Generally, deeper ConvNets tend to obtain better top 1% accuracy on challenging tasks such as ImageNet detection and classification. However, the trained models are over parameterized, difficult to train, and deploy on available hardware resources *e.g.* Gpip (Huang et al., 2019) requires a considerable



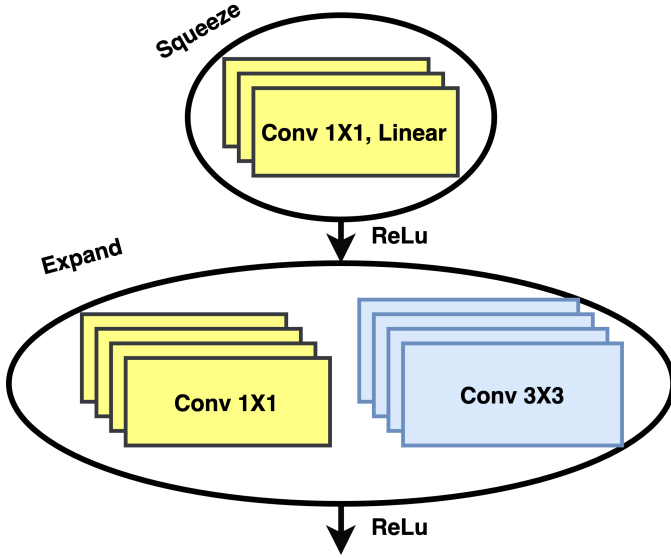


Fig. 7: Fire module: the backbone module of Squeeze Network.

number of parameters (*i.e.* 577M) to train and achieve top 1% accuracy of 84.3%.

There always has been a tradeoff between the accuracy and efficiency of model selection for a specific application. Traditionally, models achieved better accuracy by increasing depth of the architecture (using more layers), a width of architecture (via more channels), or increasing the resolution of an input image. Tan and Le (2019) proposed compound scaling to size-up all the three critical parameters (width, depth, and input image resolution) to improve the model performance. The proposed network is called EfficientNets (Tan and Le, 2019), which is the family of highly scalable and efficient neural network architectures to use compound scaling for the selection of models such as EfficientNet-B0 to B7, keeping in view the resource requirements.

The building block of EfficientNet uses mobile inverted bottleneck MBConv (Sandler et al., 2018) as shown in fig. 9 with squeeze-and-excitation optimization (Hu et al., 2018).

### 3.6. Squeeze Networks

Most of the CNN architectures are resource-hungry to achieve good accuracy on a particular dataset. The smaller architectures with equivalent accuracy require less bandwidth, and easily deployable on the hardware of limited capacity as

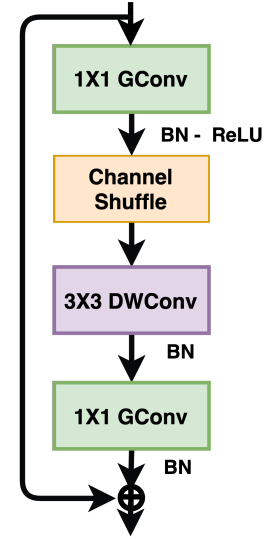


Fig. 8: Shufflenet Basic building block. “GConv” stands for group convolution and “DWConv” denote depth-wise convolution.

well as offer benefits in distributed training. The model weights for most of the CNN architectures range from 100MB (*e.g.* ResNet) to 553MB (*e.g.* VGG). The weights for AlexNet sits in the middle with 249MB. Recently, in SqueezeNet (Iandola et al., 2016), the author proposed the architecture with comparable accuracy to AlexNet with fewer parameters with an incredibly lower weight of 4.9MB. The weights are further reduced to 0.5MB by applying compression techniques such as weight pruning and sparsifying a model.

The basic module called Fire Module is used in SqueezeNet architecture with a clever combination of  $1\times 1$  and  $3\times 3$  filters. The module consists of a two-phase operation of “squeeze” and “expands”. The squeeze phase applies a smaller number of  $1\times 1$  filter compared to input volume, thus reducing the dimension of the output feature map. Before feeding to expanding phase, a ReLU activation is applied to the output of the squeeze phase. During the expanding phase, a combination of  $1\times 1$  and  $3\times 3$  filters are used to capture the spatial relationship and extract more complex features, as shown in fig 7.

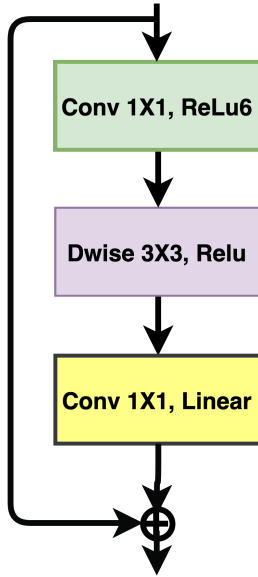


Fig. 9: Mobile Network - MBConvolution Block

### 3.7. Shuffle Networks

ShuffleNet (Zhang et al., 2018) is another lightweight architecture design belong to the family of architectures, such as MobileNet (Howard et al., 2017), CondenseNet (Huang et al., 2018), MobileNetV2 (Sandler et al., 2018), and Xception (Chollet, 2017). These architectures use group and depth-wise convolution and are suitable for low-end devices. The authors of ShuffleNet proposed guidelines for effective network architecture design.

Traditionally a widely accepted indirect metric called FLOPS is used as an only measure of computational complexity (estimation of actual run-time). However, a direct metric such as speed and latency are more relevant when considering group and depth-wise convolutions on low-end devices. The memory access cost of these operations should be considered in the neural architecture design for low-end devices. The second important factor to consider is the degree of parallelism. For example, under the same FLOPs, a model with a high degree of parallelism could perform better than the low-degree counterpart. The author used these guidelines to design a network called ShuffleNet as shown in Figure 8.

### 3.8. Mobile Networks

MobileNetV2 (Sandler et al., 2018) is tailored for use in computer vision applications designed for resource-constrained devices. The model uses less number of operations and memory access to give comparable accuracy to AlexNet. Unlike standard convolution, the MobileNetV1 proposed the use of depth-wise separable convolutions, which means that a depth-wise convolution is followed by  $1 \times 1$  convolution. In depth-wise convolution, a single filter is applied per input channel, and point-wise operation is used to combine the output of the depthwise convolutions. The models are lightweight due to the decreased number of multiplications that reduce computational complexity.

In MobileNetV2, the same authors introduced an inverted residual with a linear bottleneck layer. In a residual block with the stride of one, the first layer adopts  $1 \times 1$  convolution, followed by depthwise convolution in the second layer. The third layer uses a  $1 \times 1$  convolution without activation function, as shown in fig 9.

### 3.9. Neural Architecture Search NASNet

Neural architecture search belongs to the family of deep learning methods called meta-learning. These algorithms use auxiliary search methods such as random search, evolutionary search, recurrent neural network, and deep reinforcement learning to design various characteristics of network architectures. These characteristics include learning rate, number of filters, and filter maps *etc.*. In neural architecture search, these characteristics are learned by another neural network for searching in discrete search space. Neural architecture search, use two kinds of convolutional layers called the normal cell and reduction cell. The normal cell returns the feature map of the same dimension as that of input, while the reduction cell reduces the dimension by a factor of two.

The fundamental idea is to design a single cell opposed to a whole network. The search algorithm will search for optimal parameters from a set of parameters and then create complete architecture by stacking normal and reduction cells. Typically, searching architecture is carried out on smaller datasets, and

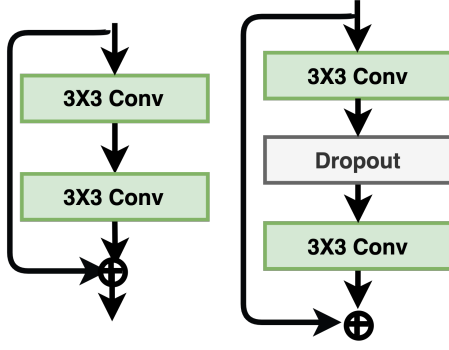


Fig. 10: The basic schematic of Wide-ResNet block with and without dropout layer.

learned layers are transferred to search architecture for large datasets. However, these approaches do not permit layer diversity required for high accuracy and lower latency in mobile applications. In MNASNets (Tan et al., 2019), a multi-objective search using deep reinforcement is proposed to find CNN models for high accuracy and low inference latency suitable for mobile devices.

### 3.10. Wide ResNets

When ResNet has scaled up to a thousand layers, a fractional improvement in accuracy requires doubling the layers and, hence, exponentially increasing the training time. In Wide-ResNet (Zagoruyko and Komodakis, 2016), the authors suggested the architecture with decreased depth and increased width compared to the ResNets to obtain good accuracy over its thin and deep counterpart.

In a typical ResNet, there are two blocks, a basic (residual) and a bottleneck block. The bottleneck block was used to make the network thinner, computationally less expensive, and suitable for the design of deeper networks. Nevertheless, the authors used only the residual block with an increasing number of convolutional layers, feature maps, and filter sizes for better performance. The authors considered two factors: i) deepening factor  $l$  and ii) widening factor  $k$  where  $l$  represents the number of the convolutions in a block, and  $k$  is the number of feature maps in convolutional layers. The number of parameters increases linearly with  $l$ , and quadratically with  $k$ . Keeping in

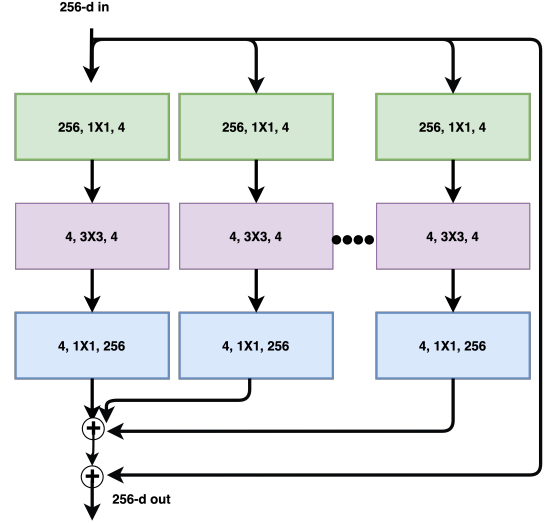


Fig. 11: ResNeXt backbone module showing different paths.

the mind that GPUs are more efficient in parallel computations, consequently widening the architecture is more effective. The performance can be further improved if dropout is used in the residual block.

Figure 10 the building block of Wide-ResNet (Zagoruyko and Komodakis, 2016) with and without the dropout.

### 3.11. ResNeXt

The ResNeXt (Xie et al., 2017) takes its inspiration from ResNet's skip connections, VGG's stacking of layers, and Inception's (Xie et al., 2017) split-transform-merge strategy.

The ResNeXt module is very similar to the Inception's the split-transform-strategy, except that the output paths are merged by addition instead of depthwise concatenation, as shown in Fig 11. Another key difference from Inception is that all split paths share the same topology. ResNeXt proposed a hyper-parameter called cardinality, which refers to independent paths used to adjust the model's capacity. The model achieved better results with the increase in cardinality rather than going deeper and wider. The model is also easier to adapt to the new datasets, as there is only one parameter for adjustment.



## 4. Results

### 4.1. Experimental Setup

We used the default settings for each of the networks. The input size of the image is the same, as specified by the authors of the networks. The input batch size is set to 32 with an initial learning rate of 0.0001. The models are finetuned for 500 epochs from the weights of ImageNet (Deng et al., 2009) due to limited number of images available in the datasets. The last layer for classification in all networks are changed to binary, to differentiate between COVID19 and Non-COVID19 radiographs or CT scans. PyTorch is used as a framework for training and testing the algorithms.

### 4.2. Datasets

Since there is no single sizeable dataset available for CXR images of COVID-19 patients, the dataset is curated from multiple sources to have sufficient data for training, testing, and validation.

**COVIDx:** The first dataset COVIDx (Wang and Wong, 2020) was made public to the research community, a collection of four classes of CXR, namely 1. Normal 2. Bacterial 3. Pneumonia (non-COVID), and 4. COVID-19. A total of 5941 PosteroAnterior(PA) CXR are collected from 2839 patients. Currently, the dataset contains only 68 X-Ray for COVID-19 patients. There are 1203, 931, and 660 samples available for negative pneumonia, bacterial pneumonia, and non-COVID viral pneumonia, respectively.

**COVIDCT:** The University of San Diego collected a CT dataset of 349 CT images called COVID-CT (He et al., 2020). The COVID-CT dataset contains clinical information of 216 COVID-19 patients. The CT images are extracted using PyMuPDF4 from 760 preprints such as medRxiv2, bioRxiv3, NEJM, JAMA between January 19th and March 25th, 2020.

### 4.3. Metrics

We take into account the following most common five metrics used in detection to evaluate each algorithm discussed earlier in the manuscript.

- **Precision:** The ratio of correctly predicted positive COVID19 patients to the total positive predictions (*i.e.* True positives and False positives). This metric gives the ability of an algorithm to

determine the rate of false positives. The higher the precision is, the lower the false positives are.

- **Recall:** is also known as the sensitivity of the algorithm. It is the ratio of correctly predicted positive outcomes (*i.e.* True positives) to the actual class observations (*i.e.* True positives and False negatives).
- **F1 Score:** takes false positives and false negatives by taking the weighted average of the earlier mentioned metrics. F1 score is useful in cases where class distribution is uneven.
- **Accuracy:** It is the most used and intuitive measure in classification. Accuracy is defined as the ratio of the correct predictions to the total number of samples. Although high accuracy may be a good measure; however, it may not be the best in certain situations where the class distribution is not symmetric. Hence, we use other metrics to evaluate the performance of algorithms.
- **AUC** stands for *area under the curve* and is the second most used metric for classification. It represents the degree of separability. The aim here is to model the capability of the network in distinguishing between classes. The higher value of the AUC means the model is better in predicting correct values or, in other words, positive as positives and negatives as negatives.

### 4.4. Evaluations

The quantitative results are reported in Table 1 and Table 2 for COVIDCT and COVIDx datasets, respectively.

The accuracy of the COVIDCT dataset varies from 70% to 81%. Moreover, GoogleNet achieves the highest average recall of 94.29%, and DenseNet169 has the highest precision of 100%. The highest performance for area under the curve is 88.80%, achieved via ResNet101.

The accuracy of the deep learning models on the COVIDx dataset is higher as compared to the COVIDCT dataset ranging from 78.23% to 87.1%. On average, the accuracy is more than 82%. On the other hand, the deep models yield lower results for recall while for similar results for precision on the COVIDx dataset. The highest recall achieved is 47.62% by MNASNet1.0. Similarly, the best precision is by GoogleNet and EfficientNet-b3, which is 83.33%. Similarly, the models are struggling to produce comparable results for the AUC metric, where DenseNet201 gives the top performance, achieving 78.59%.

### 4.5. Attention

Each model focuses on specific aspects of the image to detect an object or an artifact. Here, we present the models on infected and non-

Table 1: Five quantitative measures for state-of-the-art deep learning networks on COVIDCT. The variants of the same algorithm are differentiated via the number at the end of the method’s name.

Methods	average recall	average precision	average F1	average accuracy	average AUC
AlexNet	0.7810	0.7321	0.7558	0.7389	0.8007
VGG11	0.8952	0.7344	0.8069	0.7783	0.8785
VGG13	0.7524	0.8144	0.7822	0.7833	0.8610
VGG16	0.8381	0.7333	0.7822	0.7586	0.8395
VGG19	0.8476	0.7876	0.8165	0.8030	0.8796
ResNet18	0.7524	0.7670	0.7596	0.7537	0.8397
ResNet34	0.8667	0.7982	0.8311	0.8177	0.8851
ResNet50	0.7905	0.8300	0.8098	0.8079	0.8769
ResNet101	0.8571	0.7826	0.8182	0.8030	0.8880
ResNet152	0.7333	0.8191	0.7739	0.7783	0.8670
DenseNet121	0.1905	0.8000	0.3077	0.8548	0.7009
DenseNet161	0.1429	0.7500	0.2400	0.8468	0.6805
DenseNet169	0.2381	1.0000	0.3846	0.8710	0.7379
DenseNet201	0.1905	0.4444	0.2667	0.8226	0.7859
GoogleNet	0.9429	0.6306	0.7557	0.6847	0.7815
Efficient-b0	0.7333	0.7700	0.7512	0.7488	0.8518
Efficient-b1	0.8095	0.7589	0.7834	0.7685	0.8588
Efficient-b2	0.7143	0.7895	0.7500	0.7537	0.8325
Efficient-b3	0.8190	0.8037	0.8113	0.8030	0.8862
Efficient-b4	0.8190	0.8037	0.8113	0.8030	0.8862
Efficient-b5	0.6952	0.7300	0.7122	0.7094	0.7903
Efficient-b6	0.7524	0.7900	0.7707	0.7685	0.8552
Efficient-b7	0.7905	0.7905	0.7905	0.7833	0.8566
SqueezeNet1.0	0.9048	0.7090	0.7950	0.7586	0.8722
SqueezeNet1.1	0.9333	0.7259	0.8167	0.7833	0.8725
MNASNet0.5	0.5238	0.7857	0.6286	0.6798	0.7938
MNASNet1.0	0.8952	0.7833	0.8356	0.8177	0.8845
ResNeXt50-32x4d	0.8000	0.8235	0.8116	0.8079	0.8726
ResNeXt101-32x8d	0.8000	0.7368	0.7671	0.7488	0.8624
Wide-ResNet50.2	0.7905	0.7757	0.7830	0.7734	0.8571
Wide-ResNet101-2	0.8667	0.7712	0.8161	0.7980	0.8863
ShuffleNet-v2-x0.5	0.8190	0.7611	0.7890	0.7734	0.8654
ShuffleNet-v2-x1.0	0.7714	0.7431	0.7570	0.7438	0.8289
MobileNet-v2	0.7810	0.7736	0.7773	0.7685	0.8549

Table 2: Quantitative results for state-of-the-art deep learning algorithms on COVIDx. The numbers at the end of the method name indicate variants of the same algorithms.

Methods	average recall	average precision	average F1	average accuracy	average AUC
AlexNet	0.0952	0.5000	0.1600	0.8306	0.6445
VGG11	0.1429	0.5000	0.2222	0.8306	0.6815
VGG13	0.1905	0.5000	0.2759	0.8306	0.6875
VGG16	0.1905	1.0000	0.3200	0.8629	0.7240
VGG19	0.1905	0.8000	0.3077	0.8548	0.6366
ResNet18	0.2857	0.8571	0.4286	0.8710	0.7406
ResNet34	0.1905	0.6667	0.2963	0.8468	0.7235
ResNet50	0.1429	0.7500	0.2400	0.8468	0.6588
ResNet101	0.0952	0.4000	0.1538	0.8226	0.5825
ResNet152	0.1429	0.4286	0.2143	0.8226	0.6084
DenseNet121	0.1905	0.8000	0.3077	0.8548	0.7009
DenseNet161	0.1429	0.7500	0.2400	0.8468	0.6805
DenseNet169	0.2381	1.0000	0.3846	0.8710	0.7379
DenseNet201	0.1905	0.4444	0.2667	0.8226	0.7859
GoogleNet	0.2381	0.8333	0.3704	0.8629	0.6990
EfficientNet-b0	0.1429	1.0000	0.2500	0.8548	0.6893
EfficientNet-b1	0.2381	0.4545	0.3125	0.8226	0.6644
EfficientNet-b2	0.3333	0.7000	0.4516	0.8629	0.7180
EfficientNet-b3	0.2381	0.8333	0.3704	0.8629	0.7263
EfficientNet-b4	0.1429	0.7500	0.2400	0.8468	0.7499
EfficientNet-b5	0.0952	0.5000	0.1600	0.8306	0.6251
EfficientNet-b6	0.1429	0.3750	0.2069	0.8145	0.6671
EfficientNet-b7	0.1429	0.7500	0.2400	0.8468	0.7069
SqueezeNet1.0	0.2381	0.7143	0.3571	0.8548	0.6399
SqueezeNet1.1	0.2381	0.4545	0.3125	0.8226	0.6403
MNASNet0.5	0.3333	0.7000	0.4516	0.8629	0.7855
MNASNet1.0	0.4762	0.4545	0.4651	0.8145	0.7305
ResNeXt50-32x4d	0.2381	0.5556	0.3333	0.8387	0.7110
ResNeXt101-32x8d	0.1429	0.7500	0.2400	0.8468	0.6990
Wide-ResNet50.2	0.1905	1.0000	0.3200	0.8629	0.6630
Wide-ResNet101-2	0.2381	0.3125	0.2703	0.7823	0.6144
ShuffleNet-v2-x0.5	0.2381	0.5556	0.3333	0.8387	0.6620
ShuffleNet-v2-x1.0	0.1905	0.4000	0.2581	0.8145	0.6482
MobileNet-v2	0.3333	0.7000	0.4516	0.8629	0.7180

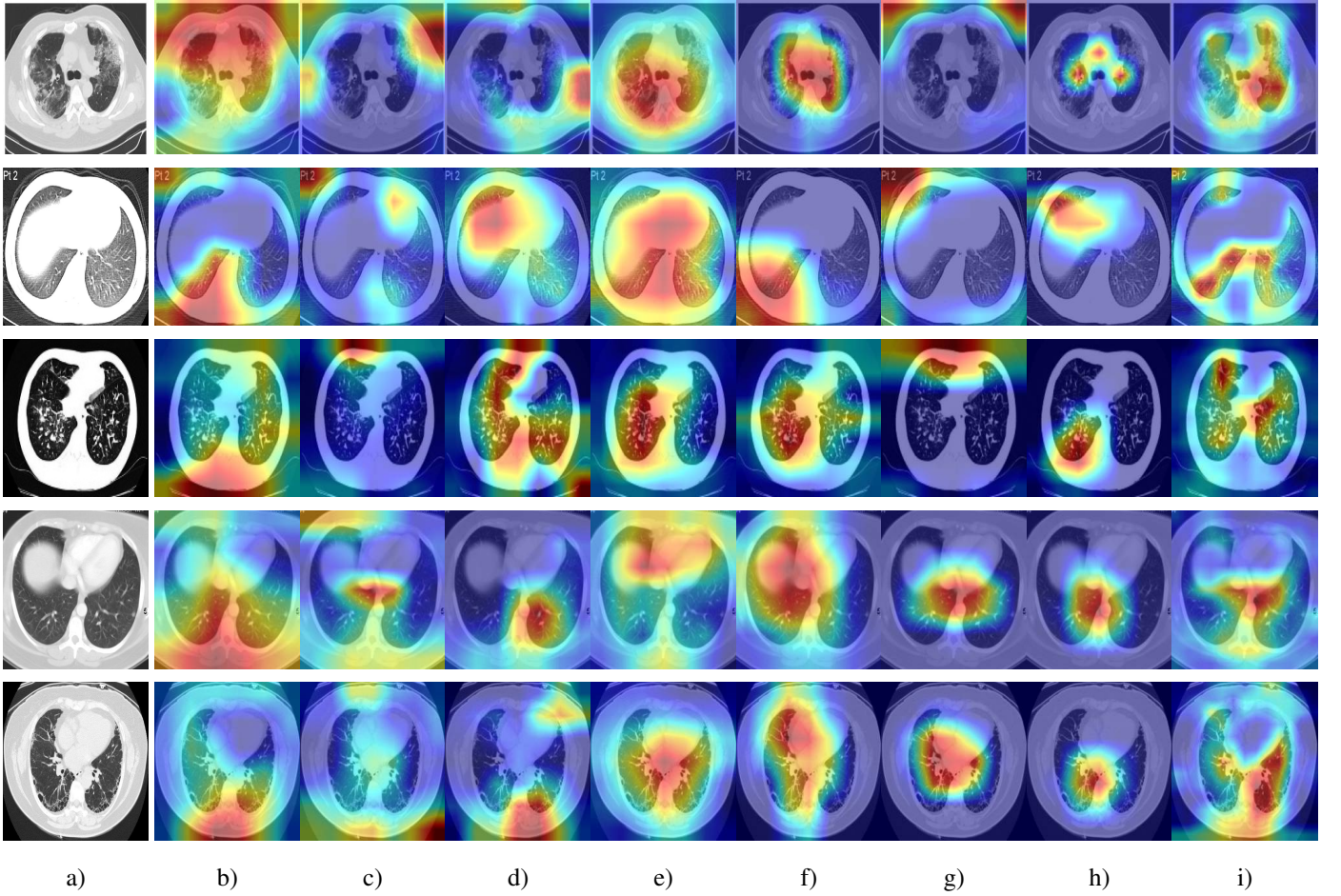


Fig. 12: Representative images from COVID-CT to show the algorithm focus areas. The red color in the images shows the focus of the network. The first three rows of images are of infections with COVID19, while the last two rows of images are COVID19 infection-free. The sequence of the images is a) Input Images, b) AlexNet, c) VGG, d) ResNet, e) DenseNet, f) GoogleNet, g) EfficientNet-b0 h) EfficientNet-b7 and i) SqueezeNet.

infected radiographs. In Figure 12, we present the CT images with feature attention where the red color indicates the region where the models have focused. The first three rows contain COVID19 infections, while the remaining two rows in Figure 12 are infection-free.

Figure 13 shows four different COVID19 infection radiographs from four different orientations. ResNet, DenseNet, and GoogleNet presented in the second, third, and fourth columns focused on most of the chest radiographs while the remaining models concentrated on particular regions of the chest. It is challenging for the models to pinpoint exactly the artifacts caused by COVID19, as is obvious from the feature attention mechanism.

## 5. Conclusion

In this work, we have tested the capacity of the current state-of-the-art deep learning algorithms and provide baselines from future research comparisons on two publicly available COVIDCT and COVIDx datasets. We aimed to differentiate between COVID19 infected and non-infected scans and radiographs. We have shown the quantitative results and attention of the models on sample images. We have employed several metrics to give a more comprehensive understanding of network performance. Although the results are promising, the need for a more significant number of images will be helpful for further training and testing.



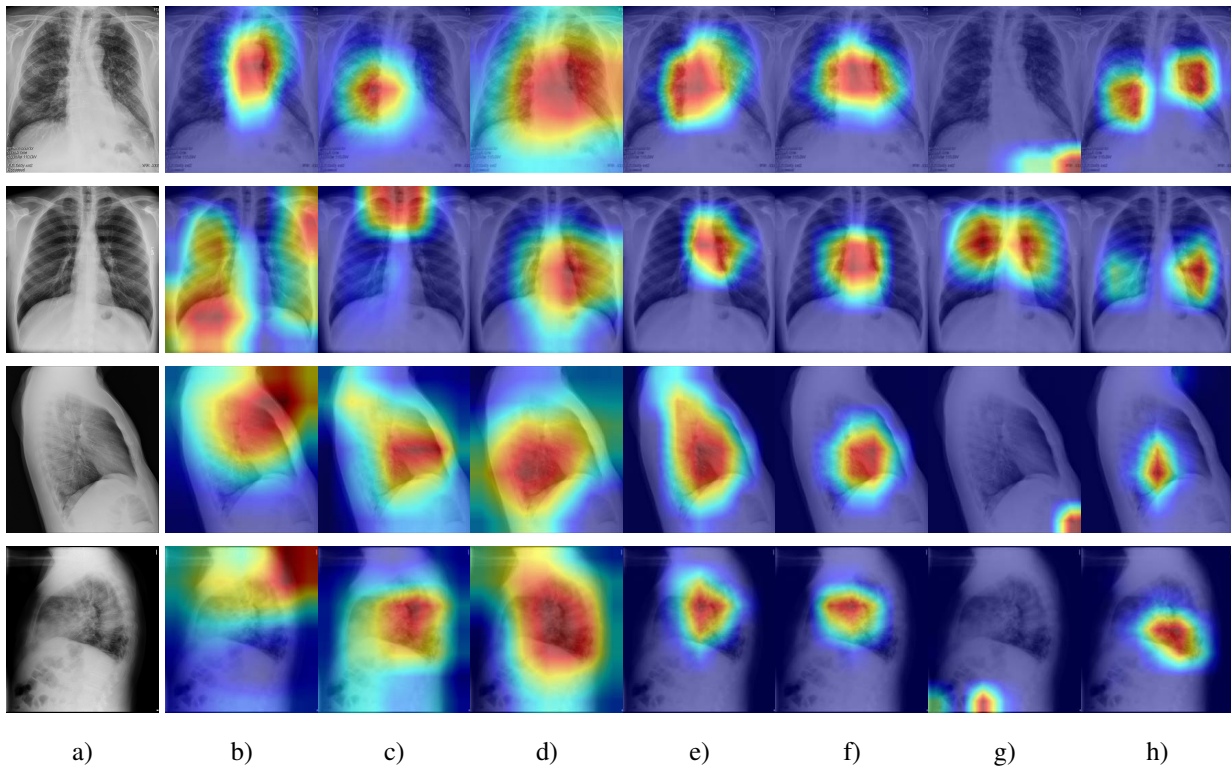


Fig. 13: Samples from the COVIDx dataset to show the feature attention. All input images contain COVID19 infections. The sequence shown in the figure is a) Input Images, b) ResNet, c) DenseNet, d) GoogleNet, e) ResNeXt, f) MNASNet, g) EfficientNet, and h) ShuffleNet.

## References

- Chen, J., Wu, L., Zhang, J., Zhang, L., Gong, D., Zhao, Y., Hu, S., Wang, Y., Hu, X., Zheng, B., et al., 2020. Deep learning-based model for detecting 2019 novel coronavirus pneumonia on high-resolution computed tomography: a prospective study. medRxiv .
- Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251–1258.
- Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. ImageNet: A Large-Scale Hierarchical Image Database, in: CVPR09.
- Farooq, M., Hafeez, A., 2020. Covid-resnet: A deep learning framework for screening of covid19 from radiographs. arXiv preprint arXiv:2003.14395 .
- Ghoshal, B., Tucker, A., 2020. Estimating uncertainty and interpretability in deep learning for coronavirus (covid-19) detection. arXiv preprint arXiv:2003.10769 .
- Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feed-forward neural networks, in: Teh, Y.W., Titterton, D.M. (Eds.), Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010, JMLR. pp. 249–256.
- He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1026–1034.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.
- He, X., Yang, X., Zhang, S., Zhao, J., Zhang, Y., Xing, E., Xie, P., 2020. Sample-efficient deep learning for covid-19 diagnosis based on ct scans. medRxiv .
- Hochreiter, S., 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 6, 107–116.
- Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 .
- Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132–7141.
- Huang, G., Liu, S., Van der Maaten, L., Weinberger, K.Q., 2018. Condensenet: An efficient densenet using learned group convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2752–2761.
- Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely



- connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700–4708.
- Huang, Y., Cheng, Y., Bapna, A., Firat, O., Chen, D., Chen, M., Lee, H., Ngiam, J., Le, Q.V., Wu, Y., et al., 2019. Gpipe: Efficient training of giant neural networks using pipeline parallelism, in: Advances in Neural Information Processing Systems, pp. 103–112.
- Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K., 2016. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 0.5 mb model size. arXiv preprint arXiv:1602.07360 .
- Kermany, D.S., Goldbaum, M., Cai, W., Valentim, C.C., Liang, H., Baxter, S.L., McKeown, A., Yang, G., Wu, X., Yan, F., et al., 2018. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell* 172, 1122–1131.
- Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, pp. 1097–1105.
- Li, L., Qin, L., Xu, Z., Yin, Y., Wang, X., Kong, B., Bai, J., Lu, Y., Fang, Z., Song, Q., et al., 2020. Artificial intelligence distinguishes covid-19 from community acquired pneumonia on chest ct. *Radiology* , 200905.
- Ng, M.Y., Lee, E.Y., Yang, J., Yang, F., Li, X., Wang, H., Lui, M.M.s., Lo, C.S.Y., Leung, B., Khong, P.L., et al., 2020. Imaging profile of the covid-19 infection: radiologic findings and literature review. *Radiology: Cardiothoracic Imaging* 2, e200034.
- Rajaraman, S., Candemir, S., Kim, I., Thoma, G., Antani, S., 2018. Visualization and interpretation of convolutional neural network predictions in detecting pneumonia in pediatric chest radiographs. *Applied Sciences* 8, 1715.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510–4520.
- Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations.
- Song, Y., Zheng, S., Li, L., Zhang, X., Zhang, X., Huang, Z., Chen, J., Zhao, H., Jie, Y., Wang, R., et al., 2020. Deep learning enables accurate diagnosis of novel coronavirus (covid-19) with ct images. *medRxiv* .
- Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9.
- Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q.V., 2019. Mnasnet: Platform-aware neural architecture search for mobile, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2820–2828.
- Tan, M., Le, Q.V., 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946 .
- Wang, L., Wong, A., 2020. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. arXiv preprint arXiv:2003.09871 .
- Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K., 2017. Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1492–1500.
- Zagoruyko, S., Komodakis, N., 2016. Wide residual networks. arXiv preprint arXiv:1605.07146 .
- Zhang, X., Zhou, X., Lin, M., Sun, J., 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6848–6856.
- Zheng, C., Deng, X., Fu, Q., Zhou, Q., Feng, J., Ma, H., Liu, W., Wang, X., 2020. Deep learning-based detection for covid-19 from chest ct using weak label. *medRxiv* .
- Zhou, Z., Siddiquee, M.M.R., Tajbakhsh, N., Liang, J., 2018. Unet++: A nested u-net architecture for medical image segmentation, in: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer, pp. 3–11.