Application of Artificial Neural Networks in COVID-19 and Pneumonia cases diagnosis via CXR images: A comprehensive analysis of Convolutional Neural Network and Vision Transformers

Viet-Hoang Tran
Faculty of Computer science
University of Information Technology, VNU-HCM

Email: hoangtv21082000@gmail.com

December 28, 2021

Abstract

At the end of 2019, humankind was faced with an epidemic—severe acute respiratory syndrome coronavirus 2 (SARS CoV-2) related Pneumonia, referred to as coronavirus disease 2019 (COVID-19) that people did not expect to encounter in the current era of technology. Nowadays, the COVID-19 situation becomes more and more severe, as well as sophisticated. And our country, Vietnam is being severely affected by this pandemic. That is why we have to take our actions immediately in order to cope with this pandemic. And the very first step in extinguishing this pandemic is that we have to rapidly and precisely identify COVID-19 cases. One significant improvement of the industrial revolution 4.0 is the application of information technology to medical diagnosis as an automatic method. The advances in artificial intelligence (AI) have enabled the implementation of sophisticated applications that can meet clinical accuracy requirements. Thus, various works propose an AI-based solution via X-ray image diagnosis as a quick testing method, which has high productivity in a short time. They majorly employ the Convolutional neural networks as their base feature extractor module. Especially different from their works, our study proposes the Vision-Transformer-based method as the novel approach. In the experiment, our work covers the comprehensive analysis of both feature extraction approaches and the implementation of an AI diagnosis system for COVID-19 and Viral Pneumonia cases by the state-of-the-art Pyramid Vision Transformer. Along with the strong data augmentation based on the clinical consideration, our best PVTv2_B2_li model achieves *92.99%*, *92.38%* sensitivity and *97.55%*, *89.81%* positive predictive value respectively to the COVID-19 and Pneumonia cases on COVIDx8A dataset in the COVID-Net. Our proposed solution can detect COVID-19 in a Chest X-Ray image, that may be a technical proof of the potential of the Transformers-based approach for the vision tasks. The heatmap and confidence score of the detection is also demonstrated, such that the doctors or common users can use them for a final diagnosis in practical usages.

Keywords: COVID-19, Deep Learning, Vision Transformer, Chest X-ray (CXR), diagnosis.

Contents

1 Introduction

In the campaign to combat with the spreading of COVID-19, effective screening and immediate medical response for the infected patients is crucial. Early studies identified abnormalities in chest X-ray images of COVID-19 infected patients that could be beneficial for disease diagnosis. Therefore, diagnostic imaging (Chest X-ray images or CXR images for short) is considered as one of the screening approaches, which can rapidly identify COVID-19 cases. This method is a much faster and inexpensive one in comparison with other traditional methods. However, this method has its own drawbacks. The first cons to be mentioned is that CXR image is never the only factor to diagnose COVID-19 cases absolutely correct, epidemiological characteristics and clinical manifestations are normally required for improved accuracy. The other one is the misdiagnosis between the Viral and COVID-19 Pneumonia's CXR images. These two diseases are difficult for radiologists to distinguish. Moreover, the symptoms of COVID-19 being similar to that of Viral Pneumonia can sometimes lead to the wrong diagnosis.

In order to improve disadvantages from traditional diagnostic imaging, the automatic diagnosis system, which bases on the deep learning method are able to identify both the Viral and COVID-19 Pneumonia (Viral Pneumonia is shortened to Pneumonia in this article, separated with the name COVID-19) from chest X-ray images with high accuracy. The related works [1], [2], [3], etc. majorly employ various forms of the widely used Convolutional neural network (CNN) and reach a great result even on the limited and imbalanced datasets. Specifically, our most related work- COVID-Net [4] gains excellent results: 99% and 97.5% respectively to sensitivity and positive predictive value to COVID-19 cases with the test set including 200 COVID-19 cases on their COVIDx8B dataset. With the dataset COVIDx4, the best model achieves 94.0% and 95.0% sensitivity, 93.1% and 99.0% positive predictive value respectively to COVID-19 cases and Pneumonia cases with the test set including 100 COVID-19 cases.

The above works inspire us to look up the brief history of CNN, its invention and accomplishment through the recent decade. Besides, in recent years, Vision Transformer (ViT) brings a novel approach for the end-to-end image recognition task. Its original one -Transformer has a large dominance in both natural language processing and it is the adapted version mainly for addressing the computer vision task. Our comparison analysis for CNN and ViT consists of the experiments from the authors and recent works to measure their potential: in terms of performance, various forms of robustness and even the compatibility with human vision. Result shows that CNN captures and generalizes the presence of features by the stack of numerous tiny kernels, which makes CNN can learn efficiently even in the limited data with fewer required parameters. On the other hand, ViT tends to capture the relationship of the local areas to obtain the global context of the image. This approach makes it require a large amount of data and sophisticated training. But in return, its gets more potential of learning capacity and more consistent with human vision.

Along with the improvements of CNN, ViT also gains a lot of interest from experts. Following the comparison on many aspects of ViT and CNN, we point out the remaining problems of ViT, show the way they overcome these problems by its new progressions. The progressions include the variant architectures inherited from it, some of them can act a role of a backbone to solve complex vision tasks. Through this study, we propose the state-of-the-art Pyramid Vision Transformer version 2 as a novel base of our method. Along with the strong data augmentation policies from a clinical consideration and implement the ViT's specialized optimizers for the diagnosis of the COVID-19 and Pneumonia disease cases, compared to normal (non-respiratory-disease) cases. Through the progress with the fast training objective, our best model achieves 92.99%, 92.38% sensitivity and 97.55%, 89.81% positive predictive value respectively to the COVID-19 and Pneumonia cases on COVIDx8A dataset in the COVID-Net [4]. We hope that our work is technical proof of the potential of the Transformers-based approach for the vision tasks and may contribute as an AI-based automatic method in order to be a reliable reference source for doctors for clinical diagnosis.

2 Theoretical comparison analysis of CNN and ViT

If you would like to scan our analysis, visit 2.5 to capture the discussion of this section.

2.1 Discussion about Convolutional neural networks

Brief history and accomplishment

Convolutional neural network (CNN) is the most famous and commonly employ algorithm, which has marked an invention of deep learning for the computer vision approach. The birth of CNN is 1980's Neocognition [5] inspired by and earlier research on the cat's visual cortex [6]. CNN have begun to carry out their mission to recognize handwritten digits in [7] by applying the back-propagation algorithm. After about two decades with the work [8], CNN is recognized as a powerful approach by demonstrating the state-of-the-art performance in the large scale dataset ImageNet [9]. The main advantage of the CNN compared to their predecessors is that: it is an end-to-end model, automatically extracts the relevant features without any human supervision. It can learn the representation features of images for every specific class and have been shown to have many parallels to processing in the visual cortex [10].

Standard architecture

The basic structure of CNN does not have many differences from a fully connected neural network. Starting from an input, through layers they apply a dot product, followed by non-linearly activation, some sub-sampling layers, such as max-pooling are added for keeping only dominant features [11]. The last layer, in the classification problem, predicts inputs provided by humans. The difference lies in how many neurons are made, and the way it connects to input. In a fully connected neural network, we would have weights for each connection between an input, in this case, a single channel of a pixel and a neuron, obviously, it is incapable of scaling up, since too many neurons are created, leading to huge computation problem. CNN is originally designed to work with images, it borrows the concept of the receptive field of brain cells by having neurons look at a small portion of the image and slide across it to calculate their output-called a local connectivity. These cluster of neurons are called Convolutional filter, an amount of them compose a standard Convolutional layer of CNN.

In essence, CNN performs learning tasks and creates the architectural inductive biases toward local spatial to assume output. In detail, there are at least three kinds of inductive biases in the networks. (1) By the weight-sharing mechanism of the Convolutional filter, CNN encourages the relative translational equivariance and also the translational invariance with the help of pooling filters. Combining with (2) local inductive bias (by using a small convolution kernel) CNN focuses on local information rather than taking global information, it just determines the presence of local features and is not sensitive to the global features, helping CNN learns quickly with fewer parameters and generalizes better in the presence of a reasonably small amount of data. And (3) the relational inductive bias, when stacking Convolutional (and pooling) filters, can progressively extract higher-level feature map with high resolution, from that CNN can be directly adapted to address pixel-level dense predictions (such as object detection, segmentation and tracking) as a role of the backbone.

Nowadays, we have a vast of advanced architectures of CNN with more structurally complex and be able to do deeper in order to understand the higher-level meaning of the data. In the following contents, we show the experiments of ResNet [12] and its related variants, the first CNN architecture can go as deep as 152 layers, which won the first place on the ILSVRC 2015 image recognition and have been used for many computer vision tasks.

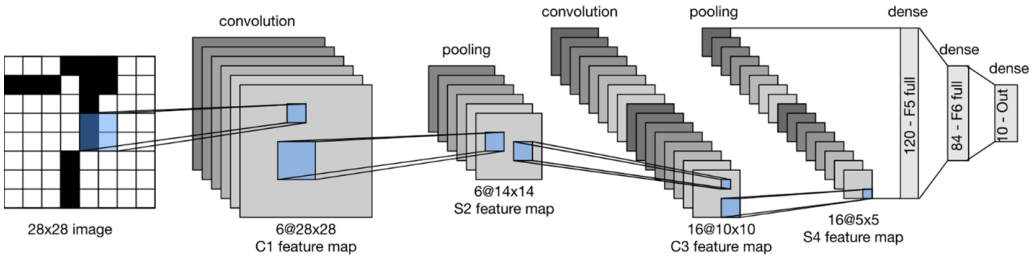


Figure 1: An illustration of the architecture of LeNet [7]. The first CNN can address the 10-digit classification task. The model takes an input image size 28×28, performs 6 convolution filters in the first Convolutional layer to create a feature map C1, with depth 6. Next, the pooling layer includes the number of filters that are similar to the input feature map's depth, having responsible for reducing the size of the feature map, specifically width & height, and preserving depth. It has the role of decreasing computational neurons of the next layers and alleviating the excessive sensitivity of the Convolutional layer to location, giving the compact output S). Then a feature map goes through to two higher layers, getting higher-level features (C3 and S4). After that, a feature map S4 is flattened to a 120-dimension vector, which is called the feature vector of the input image. Now 2 fully-connected layers act as the classifier, which gives the output is a vector of probabilities that represents 10 classes (10 digits). The highest probability belonging to a certain class is considered as predicted output.

2.2 Remaining problems of CNN

Figure 2: [13] With lacking ability of encode relative position of image's features, the figure on the left and right are considered as one and classified as Geoffrey Hinton's.

Different from human vision

According to recent results [14], the local connectivity of convolution causes CNN to make recognition based on superficial textural features rather than on the shape information preferentially used by humans, which is in stark contrast to human behavioral evidence and reveals fundamentally different recognition strategies [15]. Figure 6 shows that CNN strongly texture bias than shape, that makes them inconsistent with human vision. Some earlier approaches addressed this problem, the early work [16] studies feedback mechanism on the human visual cortex and develop a computational version for the neural network, which creates an augmented version of the input image to give a better visualize how the network work and capture the attention with the expected objects, even among of cluttered multiple objects.

Neither translational equivariant nor translational invariant

The drawback of the weight-sharing and the non-reasonable sub-sampling mechanism is making CNN does

not have the ability to encode the position and orientation of the object into its predictions. CNN completely loses its internal data about the position and poses of the object when the input is fed to numerous Convolutional and pooling filters. The common pooling operation such as max-pooling may give some amount of translational invariance, but in the trade of it sacrifices the perfect translational equivariance due to ignoring the Nyquist sampling theorem. In fact, the pooling only expects the dominant information that activates the neural signal and ignores the rest. The proof of it, work [17] observes that the translational equivariance in the Convolutional layers also can be lost by the following subsequent pooling (or sub-sampling) layers. In practical usage, this defect of property ignores the relationship between the part and the whole image. The example in figure 2 shows that CNN encodes the landmark features but except their relative position.

The perfect translation invariance property is not fully guaranteed either, CNN consists of predefined pooling configurations for dealing with the various variations of input's spatial arrangement of data. The pooling filters typically support a small spatial invariance (i.e. only support a small shift in image), therefore not actually invariant to large transformations of the input data [18]. With the above argument, we conclude that equivariance and invariance are mutually exclusive functions. Even modern CNN architectures are designed to sacrifice the complete property of translational invariance to being more translational equivariant [17]. So that an extracted representation by these filters cannot support both equivariance and invariance, perfect invariance supported by a network must come from fully connected layers rather than in the equivariant Convolutional layers [19]. On the other hand, the more-quality equivariance property comes from the stack of Convolutional layers with the reasonable form of pooling operations. The recent work [20] integrates the anti-aliasing with the architectural convolution and max-pooling for further preserving the translational equivariance, hence improving performance of various CNN models on the ImageNet [9] benchmark.

2.3 Transformers - from language to vision

Transformers [22] is the modern method of deep learning, are originally invented to address natural language processing problems and having a large dominance in this field. It relies by a simple yet powerful mechanism called Self-Attention. In general, this mechanism contextually updates weights by the relevance of certain information. It helps sequence data like words to be transmitted and processed parallelly. That creates a great advantage since it is possible to take advantage GPU's power and be able to process data types such as image. Related work [23] proves that with the help of the relative positional encoding [24], Multi-Head Self-Attention (MHSA) layer can express any Convolutional layer. Furthermore, Transformer-based models can simultaneously learn local and global attention based on input content. In general, it learns the positions of its receptive field on the whole image for every block (instead of a fixed grid of CNN). Various architectures based on Transformers also has a lot of success in multiple computer vision tasks, such as object detection [25], image classification [21] and image generation [26], [27], [28].

Transformer-based model in computer vision

To prove the performance of Transformers method to image data, the first pure Transformer-based model is introduced for evaluation and comparison to CNN in computer vision task. Vision Transformer [21] is the first Transformer-based model that approaches state-of-the-art on multiple image recognition benchmarks in the early 2021. In general, ViT treats the image as an input sequence. Initially, (1) image $(W, H, 3)$ is splitted and flatten into a sequence of 2D patches $(N \times P^2.D)$ by the linear projection. Where H, W is the resolution of the original image, D is the number of channels, $P.P$ is the resolution of each image patch, and N is the resulting number of patches. So each patch of image is now considered to be a sequence of words (as mentioned in its paper's title). (2) Analogously as BERT [29], they add an extra embedding (`[class]`) token for classification task (right after flattening the image) which locates at the top of linear

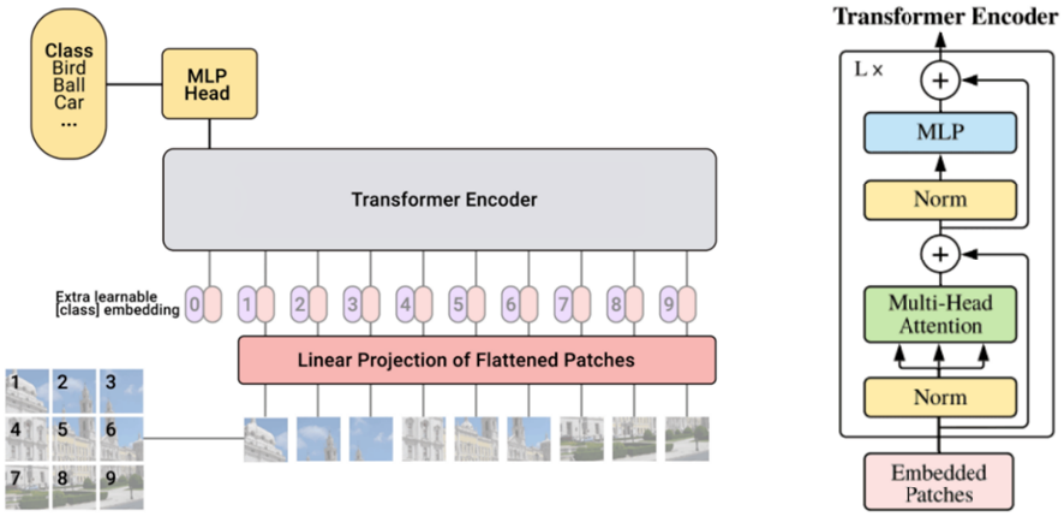


Figure 3: Architecture of ViT [21], the first Transformer-based neural network with convolution free **(left)** for image classification and the Encoder architecture **(right)**. As you can see, the input image is splitted into fixed-size patches, so the image size must be divisible by the patch size.

projection's embedding features and all of them are added with 1D positional embedding. The positional embedding preserves location information from input data. (3) The embedded outputs are subsequently fed into Encoder, where the Attention mechanism is implemented. The Encoder architecture is similar to the original paper [22]. From the output of Encoder, ViT gets a transformed `[class]` token. (4) For classification, it is subsequently fed into the deep neural network, acting as a classification head. Detailed of ViT structure can be shown in figure 3.

2.4 Compare CNN (ResNet) with ViT

Consistency with human vision

Transformers do not have spatial inductive biases learning mechanism like CNN, ViT performs learning tasks entirely by spatial allocation of attention. In the results of comparing the CNN model: ResNet-50 and the Transformer-based model [30]: ViT [21] by experiments on ImageNet-21K [31] and ILSVRC-2012 dataset, they propose that ViT not only outperforms ResNet on accuracy for image classification tasks, but also has higher shape bias and are largely more consistent with human errors. After fine-tuning with texture-bias-decreasing data augmentation methods, they observe that ViT maintains its accuracy, whereas also gaining equivalently in its shape bias when compared to ResNet-50. They subsequently propose the explain that Transformer-based models can focus on the part of the image that is important for the given task and neglect the otherwise noisy background to make predictions. Figure 6 describes how does ViT get more Consistency with human vision than ResNet.

Local/global information preservation

figure 5 is the computation cost comparison and maximum path length of Convolutional and Transformer-

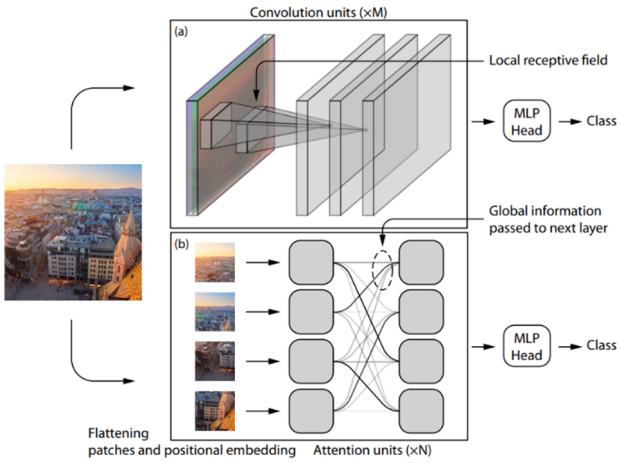


Figure 4: [30] Bird-eye views of the Convolutional **(a)** and the Transformer-based **(b)** network. It is proved that Attention mechanism of Transformers can keep global context of input's information in the early stage.

Layer Type	Complexity per Layer	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(n/r)$

Figure 5: [22] The comparison of 3 layer types by complexity, per layer and maximum path length, with Self-Attention layer represents to analyse ViT; Convolutional layer represents CNN and restricted Self-Attention represents layer ViT's variants. Where n is the sequence length, d is the representation dimension (of the linear projection matrices in Self-Attention layer or convolution kernels in Convolutional layer), k is the Convolutional kernel size and r is the size of neighborhood in restricted Self-Attention.

based in a single layer. A Convolutional layer width $k < n$ does not connect all pairs of input and output positions (in the most Convolutional model, k is very small compared to n). The Convolutional layer requires a stack of $O(n/k)$ Convolutional layers in the case of contiguous kernels, or $O(log_k n)$ in the case of dilated convolutions, that is increase length of the longest paths between any two positions in the network. The longer of the maximum path length, the harder it is to track the long-range dependencies of the input data. That is the explanation why ResNet is strongly locality-sensitive in the early layers and then the observation is progressively expanded to global after the stack of numerous Convolutional layers. On the other hand, the Self-Attention layer is considered to be a convolution with the entire image itself acting as a receptive field and have the equivalent function as convolution. Thus the Transformer-based model concentrates on learning long-distance dependencies even in the early stages. And with the help of small patch size splitting and MSHA mechanism, it can also learn the local dependencies, then tending to larger-region observation in the following stages (see more information at figure 11).

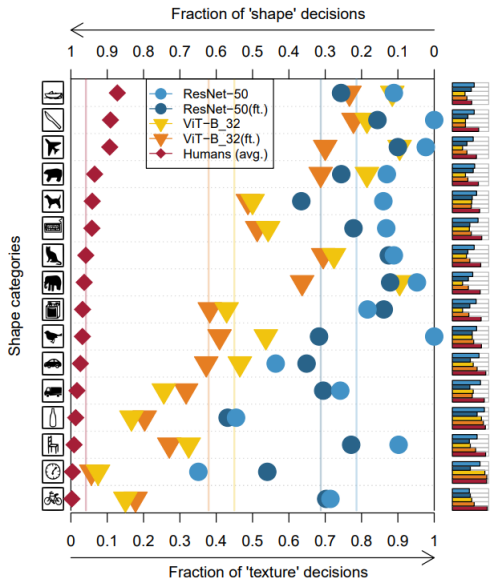


Figure 6: [30] Comparison of 'texture' and 'shape' bias of the standard ResNet-50 and ViT-B/32 before / after fine-tuning (denote as ft.) and human vision. As we can see that ViT models classify images (contains object in categories) strongly biased by texture features than shape. While humans tend to have the opposite preference and ViT biases is much more similar to human error.

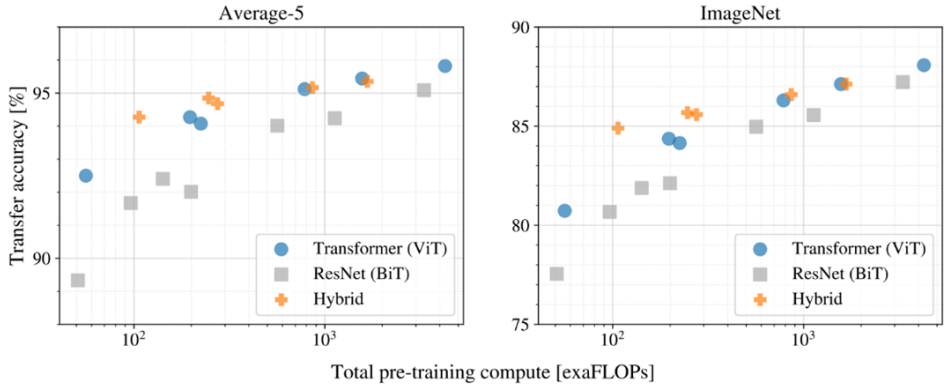


Figure 7: [21] Transfer performance (vertical axis) versus pre-training computation (horizontal axis) of different architectures: ViT, ResNet (BiT) and Hybrid on average over 5 different datasets (**left** figure), and ImageNet dataset (**right** figure). On the left figure, ViT models dominate BiT with the same computational resource, ViT only uses approximately less than four time training computation to achieve the same performance as BiT. Hybrid model (ViT with CNN's extracted features as input) outperforms ViT at small computation, but the difference vanishes when the computation goes larger.

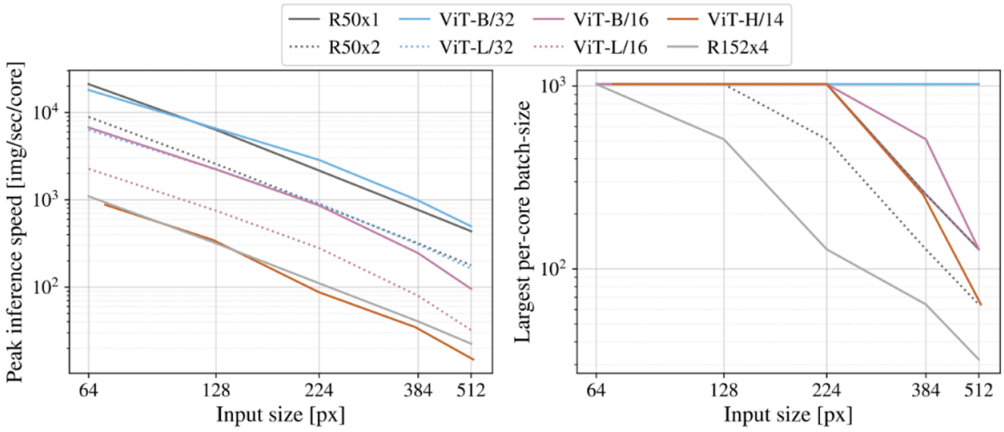


Figure 8: [21] **Left** figure shows how many images one core can handle per second, across input image sizes of various architecture, ResNet: R50x1, R50x2, R152x4 and ViT: B/32, L/32, B/16, L/16, H/14. Where B, L, H represent Base, Large and Huge number of parameters respectively and /32, /16 represent to 32×32, 16×16 input patch size. ViT models have results comparable to ResNets, whose speeds are scaling bi-quadratically with image size.
Right figure shows the largest batch-size of each model can fit into the same core across input sizes with the same computation core, the larger is better for scaling to the large dataset. ViT models are clearly beats ResNet about memory-efficient. As you can see, the compact R50x1 is forced to reduce the largest batch-size rapidly as input size increases early, whereas ViT-B/32 can hold this metric until becomes to the size 512. But once the metrics sign to decrease, they decrease faster than the ResNet model's, this is due to the property of computational complexity of layer type in each model (detail in figure 5).

Computational complexity

Let's look at figure 5. For the Self-Attention layer, the comparison of computational complexity largely depends on the input data size. In computer vision task, the sequence length is considered to be feature map dimension, which means the complexity of the Self-Attention layer is quadratic to the input dimension. Thus exploding the computation time and memory needs when implementing with higher resolution images. Using restricted Self-Attention may reduce the complexity, but sacrifice the learning of long-range dependencies. In fact, ViT performs MHSA with many heads in order to reduce the complexity, while also preserving their connectivity. On the other hand, the complexity of the Convolutional layer is quadratic to the number of filters (assume that the number of filters is equal to the dimension of feature map in the worst case), which is pre-configured in model architecture. We can not conclude which has less complexity. For experiments with low-resolution images, ViT paper [21] proves that the training performance of (huge-dataset) pre-trained ViT can dominate BiT (set of pre-trained ResNet for downstream tasks in Big Transfer [32]) at computational complexity and reach better results (look at figure 7 for more detail). On the other hand, ViT has a drawback that it can not handle images having high-resolution, whereas ResNets can work with variable-size image.

Data demanding

Besides the out-performance of computation and performance, the requirements of input data are also im-

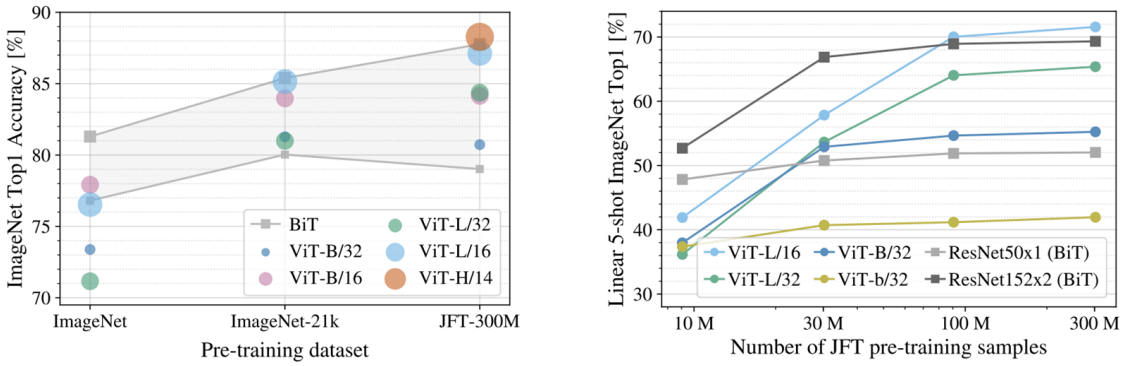


Figure 9: [21] **Left** figure is the transfer results of pre-training different ViT and BiT models on (ImageNet, ImageNet-21K and JFT-300M) dataset. BiTs outperform ViTs in the small dataset. In the bigger dataset, ViTs overtake BiTs and the performance of ViTs keep increasing, meanwhile BiTs sign to decrease.
Right figure is the few-shot evaluation results of these models versus different training sizes in the JFT dataset. ResNets perform better with smaller pre-training datasets but plateau sooner than ViTs, which perform better with larger pre-training. ViT-b is ViT-B with all hidden dimensions halved.

portant to compare these two approaches. We yield the comparison of ViT and BiT by image classification performance/ data size demanding. ViT paper [21] experiments 3 size of ViT models (B, L, H) on 3 increasing-size datasets which are similar to BiT (ImageNet [9], ImageNet-21k [31] and JFT-300M [33]). In the smallest dataset ImageNet, ViT-Large models underperform compared to ViT-Base models; in larger dataset ImageNet-21k, their performances are quite equivalent. In the biggest dataset JFT-300M, larger models can show their potential. Besides, when comparing to the performance of CNN (shaded area of various-scale ResNet models in figure 9 left), it surpasses ViT's performance in the ImageNet dataset. ViT outperforms in the bigger dataset in contrast. Figure 9 right is the results of comparing performance of models across the volume of the training set. ViT gets lower performance than ResNet when the data size is small, since it meets over-fitting in this case whereas computation volume is equivalent to BiT, its performances increase when increasing data size whereas BiT's is saturated, even degraded when the data is too large. It may support the intuition that learning spatial inductive bias of Convolutional network is appropriate to the small amount of data. When the data goes more gigantic, learning by this assumption may harm the model [34]. On the other hand, learning directly by relevant information promotes the potential of these Transformer-based model.

Experiments on various-scale datasets

figure 11 describes work's [35] experiments ViT and ResNet-50 in the various-scale datasets and give the result that: If ViT receives a sufficiently large amount of data (relative to the model's scale) to train from scratch. Then the very first MHSA layers of ViT will have a representation similar to the corresponding low layers of ResNet-50, both of them focus on local information, ViT gets more focus on global information as the depth increase. In contrast, ViT does not pay attention to the local information if it gets insufficient data, and the performance changes worse as discussed. Thus we can make a conclusion that if we want to develop a model which learns properly with smaller data, then the local learning mechanism of early layers is absolutely essential (according to [23]). The work [36] also points out that ViT models converge faster

and only able to surpass state-of-the-art CNN when employing restricted Convolutional layers (also known as *Convolutional stem*) in the early stages. This is maybe a crucial design choice, because of it balances the inductive biases of CNN and the representation learning ability of Transformer blocks.

Data augmentation, Optimization and Layer extending

ViT is proposed as a novel approach for image feature recognition task, thus the training methods from CNN may not compatible to ViT. The author confirms that the lack of convolution-like inductive biases is the challenge of training ViT models. Recent work [37] shows that ViT models are sensitive to the choice of optimizer (compatible with AdamW [38] and SGD in the experiment of many works); requires the sophisticated tuning of hyper-parameters; strong data augmentations and large-scale pre-trained dataset to avoid over-fitting.

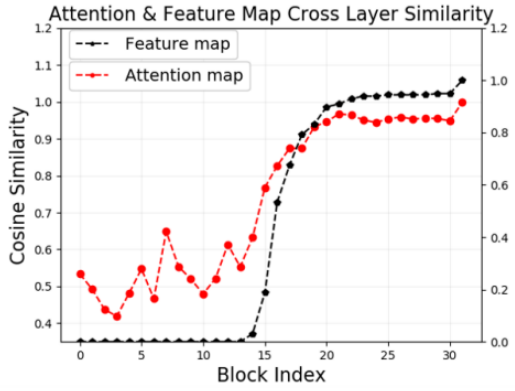


Figure 10: [39] Cross layer similarity of attention map and features for ImageNet-1k pre-trained ViTs. The black dotted line shows the cosine similarity between feature maps of the last block and each of the previous blocks. The red dotted line shows the ratio of similar attention maps of adjacent blocks.

On the other hand, these inductive biases encourage ResNet to escape from bad local minima when trained on visual data. The work [40] compares the training ability of ViT and CNN by the condition number κ of Neural Tangent Kernel [41], where κ represents the divergent does the network suffer to converge in the sharp region on its loss landscape. For comprehensible, the sharper region, the easier to stuck in the local minima and lead to over-fitting. The various-scale ResNet models have stable κ indicate that they enjoy superior training ability regardless of the depth. Besides that κ diverges with the scaling of ViT models thus it suffers poor training ability.

About the layer extending study, the works [39], [42] experiment on extended layers of standard ViT can not boost performance or even get worse (this phenomenon is called as *attention collapse*). Since the layers of ViT have the uniform similarity structure throughout the network [35]. Therefore its extracted attention maps and feature maps across the layers are getting more similar when goes to deeper, leading to more reduction of learning capacity and inference generalizing of model as depicted in figure 10.

2.5 Discussion

Convolutional neural network uses spatial inductive biases to assume outputs. It in nature uses Convolutional filters and downsampling to focus and generalize local information, this process stacks up several times to capture global information. Whereas Vision Transformer learns to focus on both local and global context information from the beginning. Therefore ViT easily neglects noisy background and encourage the shape bias, resulting in a large consistency with human vision. But the trade-off also appears with the poor ability to encode features of the small objects. ViT dominates CNN when comparing computational complexity and performance in various datasets. But its complexity largely depends on the image size, while CNN's depends on its fixed-configuration in contrast. About the learning capacity, ViT is able to learn more knowledge but requires a huge amount of data, while CNN has a lower understanding of the addressed task but require less

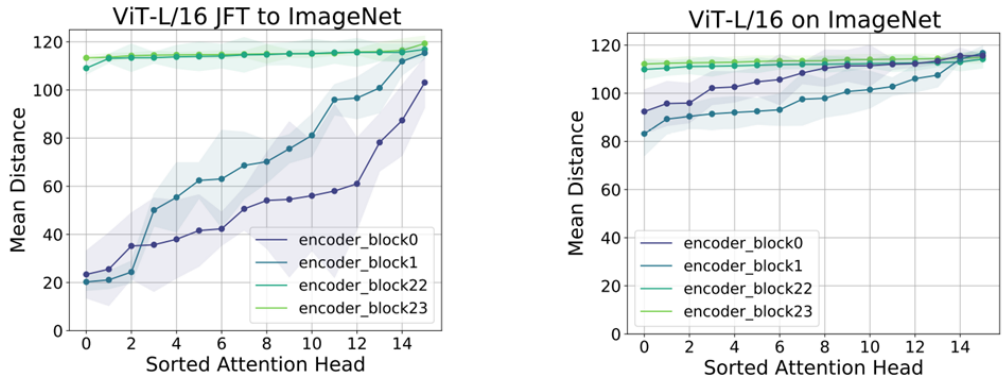


Figure 11: [35] Mean of the Self-Attention distances of pixels by head through layers of ViT which is fine-tuned with ImageNet from *large-scale pre-trained JFT-300M* **(left)** and *training-only* on ImageNet **(right)**. The attention distances represent that how far does the pixels take queries to others. As we can see in the low layers, the model which has large-scale training data pays attention to both locally and globally. As the depth increases, the attention distances increase. And all the heads attend widely across the image in the intermediate layers onwards. On the other hand, less data-training model does not attend locally.

data. In the training progress, ViT is also sensitive to the optimizer, requires sophisticated tuning hyper-parameters and strong data augmentation. Extending ViT with deeper layers can not boost the performance or even get worse, while CNN with residual connection can enjoy the superior training ability regardless of its depth. The developments of Transformer-based networks which follow the achievements of ViT tend to incorporate local learning mechanism of CNN to image feature recognition framework to achieve better performance and reduce computational complexity.

One thing to note is that the above comparisons are from the image feature recognition task. ViT is applicable to perform image classification, but unlike CNN, it is challenging to directly adapt ViT to address pixel-level dense predictions. Because (1) the output feature map of ViT is single scale and low resolution. (2) Additionally, it takes high computation and memory cost even for common input image sizes (as pointed in [43]). The following section will cover the most recent progressions of image feature recognition, which are tending to inherit both CNN and ViT merits and reaching a great achievement on image recognition (even on pixel-level dense predictions) benchmarks.

3 ViT Improvements

ViT (10/2020)[1] [21] is gaining a lot of interest from researchers since it offers a novel approach that is away from CNN. Therefore, there are a lot of improved variants of Transformer networks that start from the success of ViT. Their approaches mainly solve four problems: *(1)* how to extend or reconstruct ViT to either data-efficient training or better generalizing at inference with more computational complexity saving; *(2)* how to reinforce the both of local and global representation learning ability in an efficient way; *(3)* how to find the compatible optimization, regularization and data augmentation strategy of ViT; *(4)* and how to make ViT enable to produce multi-scale feature maps with high resolution (like CNN) to address pixel-level dense prediction tasks. Based on our knowledge, we further yield typical improvements which inherit the baseline of ViT through time and address 4 problems above in the following content.

3.1 Typical improvements

In order to address the *(1)* and *(3)* problems, DeiT (12/2020) [37] introduces the knowledge distillation training strategy for Vision Transformers. Combining with the selected regularization and strong image augmentations can extend ViT to a data-efficient network. It can be trained on ImageNet from scratch and get competitive results. According to the experiment, ViT achieves 88.55% top-1 ImageNet accuracy by JFT-300M pre-trained model, whereas the best DeiT achieves 84.4% accuracy with the same training regime and be capable of affording up to 3x more throughput.

To help ViT go deeper effectively *(1)*, DeepViT (3/2021) [39] proposes the Re-Attention mechanism to encourage the diversity of the attention representation. Thus helping DeepViT increase the exploitation of more context information at the intermediate and high layers with negligible computation and memory cost. The author announces that the best DeepViT not only improves 1.6% top-1 accuracy when training only on ImageNet but also obtains more parameter-efficient than DeiT.

Inherit the baseline of DeiT to obtain more locality prior *(2)*, TNT (3/2021) [44] proposes to model both patch-level representation and pixel-level representation by serial pairing the inner and outer Transformer blocks. This method provides a pixel-level Self-Attention for extracting explicitly local information and simultaneously preserves complexity by weight sharing mechanism. The best TNT achieves 82.8% top-1 accuracy on ImageNet benchmark (without pre-training), while requiring less parameters and FLOPs compared to DeiT.

Also address *(2)*, CvT (3/2021) [45] and CMT (7/2021) [46] leverage a Convolutional layers as a projection into vanilla Transformer blocks. For CvT, their ImageNet-22k pre-trained model achieves 87.7% top-1 accuracy on ImageNet benchmark. Besides, CMT maximizes the advantage of utilizing both CNNs and Transformers by studying the different components including Convolutional stem as the low-level extractor, the Transformer block consists of depth-wise Convolutional layers, spatial reduction MHSA along with the inverted residual block presented in MobileNetv2 [47]. Empirical results demonstrate the performance of this hybrid architecture. The best CMT achieves 83.5% top-1 accuracy on ImageNet without pre-training, while being 14x smaller on FLOPs than DeiT. Experiment of detection tasks (to solving *(4)*), CMT achieves 44.3% mAP on COCO val2017 detection task, serving as a backbone of the RetinaNet [48].

Further effort to address *(3)*, the work (6/2021) [40] proposes the Sharpness-Aware Minimizer (SAM) optimizer [49] to ViT, which outperforms ResNet-50 for both of the performance and many forms of robustness even in a larger size without the dependency for large-scale pre-training or strong data augmentations. In detail, SAM focuses on seeking the set of parameters that lie in neighborhoods within a certain radius which have a uniformly minimum loss value, rather than seeking only singleton parameters having low loss (like

[1] submit date on Arxiv

Method	Params.(M)	FLOPs(G)	Top-1 Acc (%)
DeiT_tiny/16 [37]	**3.4**	**0.6**	70.5
PVTv2_B0 [51]	5.7	1.3	**72.2**
ResNet18 [12]	11.7	1.8	69.8
CvT-13-NAS [45]	**18.0**	4.1	82.2
ViT_small/16-SAM [40]	22.0	9.9	78.1
DeiT_small/16 [37]	22.1	4.6	79.9
PVTv2_B2_li [51]	22.6	**3.9**	82.1
TNT_small [44]	23.8	5.2	81.3
DeepVit_small*[39]	27	$12.4^{?}$	82.3
CMT_small [46]	25.1	4.0	**83.5**
ResNet-50 [12]	25.6	4.1	76.2
Swin-T [50]	29.0	4.5	81.3

Table 1: Performance and robustness of ResNet and Transformers based models on ImageNet benchmark in the segment of tiny and small-scale of parameters. The "FLOPs" computations are under the image size of 224×224. * denotes the model trained with knowledge distillation regime. ? for the assumption that 1 FLOP = 2 × MAdd.

other first-order optimizers). SAM explicitly smooths loss landscapes during model training. Though the trade-off that comes with it is 2x training time and more computation required. ViT-B with SAM optimizer and strong augmentation achieves 81.5% accuracy top-1 accuracy on ImageNet without pre-training.

PVT (2/2021) [43] and SwinT [50] (3/2021) are the first progression of producing multi-scale feature maps *(4)* (also known as the hierarchical feature representation). In general, it can serve as a general-purpose backbone for pixel-level dense prediction tasks. They all point out that using a small-insufficient patch size in the early layers can not extract high-resolution feature map. For SwinT, it splits the input image into fine-grained patches and employs the relative positional biases into the restricted MHSA by shifted windows, which is integrated into the Encoder to achieve a linear computational complexity with respect to the image size. Besides, PVT employs the convolution layers as a spatial reduction in MHSA to greatly reduce the computation cost. Both of these models are able to control the output size at the end of their stages, creating a set of compact feature maps in a pyramid form. According to the author, SwinT-base reaches a greater performance with 84.5% top-1 accuracy on ImageNet without pre-training and demonstrates a superior backbone when comparing with others.

3.2 Pyramid Vision Transformer version 2

After taking a survey, we observe that the hierarchical model such as PVT and SwinT, whose output is the set of feature representations in a pyramid form. These feature maps bring meaningful information from low-level to high-level simultaneously, thus compatible with the role of feature extractor for pixel-level dense prediction tasks. The further work of PVT, PVTv2 [51] improves its baseline by (1) reinforcing the local and global representation learning ability by the combination of Convolutional stem and Self-Attention head, (2) further reducing the computational complexity of MHSA to linear with respect to image size, (3) flexible solving images of arbitrary size. For the empirical results, the table 1 demonstrates performance and robustness in ImageNet recognition task and PVTv2 attains relatively good results. Thus, in this work, we consider the PVTv2 as a comprehensive method, along with compatible data augmentation, optimization and regularization strategies to address the four raised problems above through the clinical disease diagnosis

problem by deep extracting the CXR images.

Overall architecture

As described in figure 13 the PVTv2 (denoted by **(1)**) has four Stages and all Stages share a similar structure. Which consists of an Overlapping embedding (denoted by **(2)**) and a stack of Transformer Encoder layers. For addressing the classification task, they add fully-connected layers for yielding the predictions.

Workflow of four Stages

Given the input image of resolution: $H \times W \times 3$ comes to the first module: the Overlapping embedding of Stage 1. First, the module enlarges the image by zero-padding in order to preserve the resolution. Next, it employs the Convolutional stem as the padding layer of stride S_1, with the square kernel of size $2S_1 - 1$ to the enlarged image by zero-padding size of $S_1 - 1$. Observe that the adjacent patches overlap half of the area, that makes the model obtains more local and continuous feature representation. The outputs are the embedded patches having size of $\frac{H}{S_1} \times \frac{W}{S_1} \times C_1$, where C_1 is the number of convolution kernels. Then these embedded patches along with positional encoding passed through the stack of L_1 Transformer Encoder layers. From here, The Encoder reinforces the local feature representation to global by performing MHSA with a unique head. Finally, the output is reshaped into feature map F_1 with size $\frac{H}{S_1} \times \frac{W}{S_1} \times C_1$.

The following three Stages: 2, 3, 4 have the same way of extracting feature maps. They use the output of the previous Stage as input. Each Stage stacks up L_i times, specifically the set of Encoder layers in all Stages is the set $\boldsymbol{L}$. Through all four Stages, we obtain the set of feature maps $\boldsymbol{F} = \{F_1, F_2, F_3, F_4\}$ in a pyramid form. This means they progressively apply the Convolutional strides: $\boldsymbol{S}$ to reduce the HW sizes throughout four Stages, while extending the channel sizes of output feature maps (configured by the set $\boldsymbol{C}$).

Transformer Encoder

As discussed below, the Transformer Encoder in the Stage i has L_i Encoder layers, each of them composes of a Linear Spatial-Reduction Attention (LSRA in figure 13 **(1)**) and a Convolutional feed-forward layer **(3)**, belong with two residual connections go parallel sequentially.

Linear Spatial-Reduction Attention

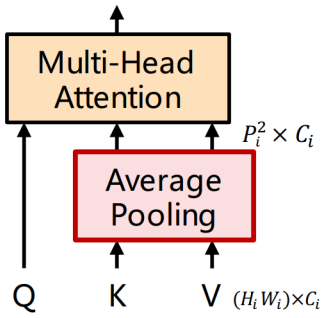


Figure 12: [51] Visualization of the module Linear SRA.

Analogously to the MHSA of Transformers [22], LSRA receives Query Q, Key K, Value V as input and the output is MHSA feature map. The difference lies in the spatial reduction of K and V before performing attention operation. Which can enjoy the computational memory cost respects to the input size like a Convolutional layer. The detail of Linear SRA operation in the Stage i can be formulated as follow:

$$\text{LSRA}(Q, K, V) = \text{Concat}(\text{head}_0, ..., \text{head}_{Ni})W^O, \tag{1}$$

$$\text{head}_j = \text{Attention}(QW_j^Q, \text{LSR}(K)W_j^K, \text{LSR}(V)W_j^V), \tag{2}$$

In (1), the LSRA$(\cdot)$ acts a role of a MHSA operation. And $W_j^Q \in \mathbb{R}^{C_i \times d_{head}}, W_j^K \in \mathbb{R}^{C_i \times d_{head}}, W_j^V \in \mathbb{R}^{C_i \times d_{head}}$, and $W^O \in \mathbb{R}^{C_i \times C_i}$ are linear projection parameters of Q, K, V and Concat$(\cdot)$ operation respectively. N_i is the head number of Stage i (the set of head number for all Stages is $\boldsymbol{N}$).

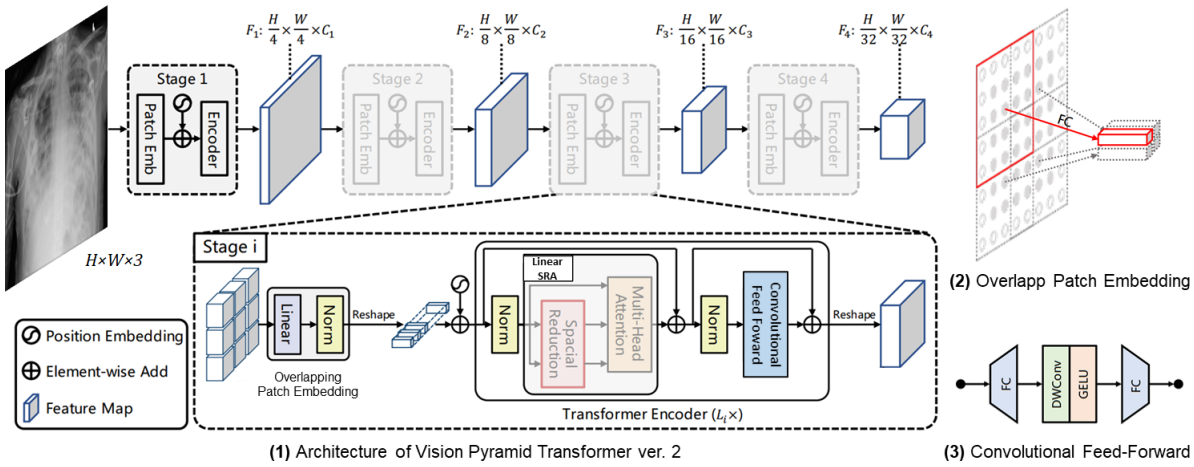


Figure 13: **Overall architecture of PVTv2**, used figures are taken from [43], [51]. The whole model **(1)** is divided into four Stages. Each Stage has an Overlapping patch embedding **(2)** concatenates with the modified Transformer Encoder, it extracts the feature map from the previous Stage and passes through to the following. The resolution of these feature maps progressive shrinks by the set of stride: $\boldsymbol{S}$ (the set $\boldsymbol{S} = \{4, 2, 2, 2\}$ is an example of this figure) while their channels $\boldsymbol{C}$ increases, constructing a pyramid form.

The input feature map has dimension C_i in this Stage is divided in to N_i heads whose dimension (i.e., d_{head}) of $\frac{C_i}{N_i}$.

In (2), the attention operation Attention$(\cdot)$ performs as original paper [22]. The LSR$(\cdot)$ is employed to reduce dimension of input sequence K and V. Formulated as below:

$$\text{LSR}(x) = \text{Norm}(\text{Conv}(\text{Pool}(x, P_i)), P_i). \tag{3}$$

Where $x \in \mathbb{R}^{(H_i W_i) \times C_i}$ represents a input sequence, and P_i denotes the adaptive average pooling size of the linear SRA in Stage i (set of configuration for P_i in all Stages is $\boldsymbol{P}$). Pool$(\cdot)$ reshapes the input sequence x to the sequence of size ${P_i}^2 \times C_i$ by adaptive average pooling. Conv$(\cdot)$ refers to the Convolution stem operation which preserves the output shape. Norm$(\cdot)$ refers to layer normalization [22]. Through these formulas above, we observe that the model achieves the linear computational/memory costs by using fixed-size K and V to perform head attention, which is configured to be tiny as a Convolutional kernel in reality (see $\boldsymbol{P}$ in table 4). Thus the model can handle larger input feature maps with limited resources.

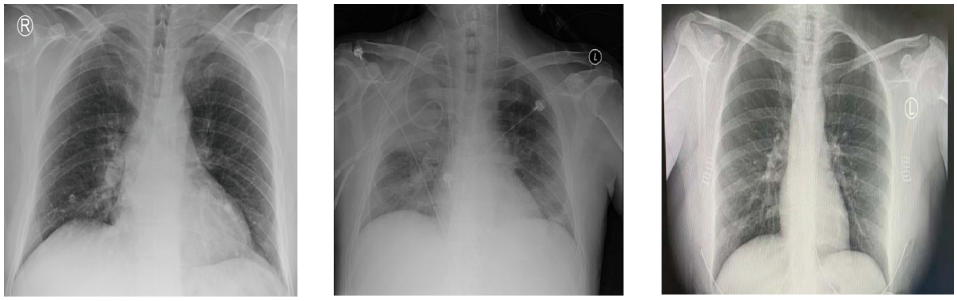
Convolutional feed-forward

After obtaining refined feature map by LSRA module. It is passed through the next module Convolutional feed-forward (**(3)** in figure 13) in order to process variable resolution of feature maps. It consists of a depth-wise convolution with kernel size of 3×3 and padding size of 1 between two feed-forward layers with non-linear activation GeLU [52]. The set $\boldsymbol{E}$ is used to stipulate the hidden feature dimension of depth-wise convolution of all Stages.

4 Experiments

Theoretical basis for the method of Viral and COVID-19 Pneumonia diagnosis via CXR image.

Recent researches [1] have shown that the combination of deep learning method and chest X-rays could be faster and less expensive than the traditional method for COVID-19 diagnosis. Deep learning method provides the ability to learn and non-linearly associate high-dimensional features in X-ray images that feature COVID-19. In order to apply deep learning method for the diagnosis of COVID-19, as well as distinguishing between COVID-19 and Pneumonia, we need to know certain patterns of both diseases compared to the case of no disease. As we can see in figure 14, the chest X-ray images of Pneumonia and COVID-19 patients are different from normal lungs in that: the lungs of infected people have more or less white spots in different places in the lungs. These white spots are medically known as the ground glass pattern. The ground glass pattern is an incompletely solidified lesion with a higher density than the surrounding lung parenchyma, we can still see the border of blood vessels or bronchi inside the lesion. A doctor who specializes in radiologists may say that these blurry glass images are the cause of the white spots in the images [53]. Therefore the specialists can use this feature to distinguish COVID-19 patients from Pneumonia. Therefore, we can use deep learning networks to extract these features, then classify to give the most appropriate diagnostic results for each case.



(A) **(B)** **(C)**

Figure 14: The chest X-ray images of **(A)** COVID-19 cases, **(B)** Pneumonia cases and **(C)** normal cases (no respiratory disease). All are taken in the dataset.

4.1 The COVIDx dataset

We use the COVIDx8A dataset [4] includes of about 20000 images and labels from different patients in 6 sources: Cohen et al. [54], General [55], General [56], Radiological Society of North America [57], Radiological Society of North America [58] and RICORD dataset [59]. The dataset includes 3 classes: COVID-19 cases, Pneumonia cases and normal cases (no respiratory disease). The authors also provide image processing tools, including a tool to convert medical images from 'mri.' to 'jpg,' and a program to remove redundant components of the aggregated dataset. We collect a total of 20401 image of cases in this material dataset, comprising 10283 normal cases, 7410 cases confirmed as Pneumonia and 2708 cases confirm as COVID-19.

4.2 Pre-processing

Class	Development		Test
	Training	Validation	
Normal	10083	100	100
Pneumonia	7305	100	105
COVID-19	2394	100	214

Table 2: The detail of distribution of image amounts by cases and dataset splitting.

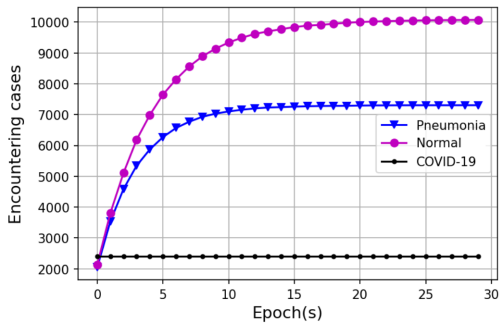


Figure 15: During the training phase, the model will encounter most of cases in our dataset at the 15^{th} epoch.

Dataset division

In our experiment, we still follow the division of development and test set of the original dataset. Different from other related works performing k-cross validation in the training step, we deduct a part of 100 images from each class of the development set for the validation. One of our objectives is implementing the experiments by fast training, thus choosing this alternative approach for data division. table 3 depicts the number of images in split sets. As we can see that the original dataset is imbalanced, the amount of normal and Pneumonia cases are dominant to COVID-19 cases. There are only a few COVID-19 images as training cases, but requires a double test amount compared to the rest classes. It is a challenge for finding the most specific COVID-19 diagnosis method.

In order to leverage of these valuable data, for each training epoch, we stochastically portion 2394 images of normal and Pneumonia classes from the original dataset and load them along with all of COVID-19 cases. The characteristic of these classes is that they are easily making the model confused with each other. The result in predictions is biased to only one or two classes if the training set is class-imbalanced. This method can help us overcome this problem while ensuring good data exploitation. Figure 15 depicts the increase of training cases for each class as the models will encounter through epochs.

Size configuring

The resolution of the input image is another problem we are interested in. The images in the RICORD dataset are reviewed by the MicroDicom viewer and export high-resolution images, they can reach 4240×3480 pixels. Besides, the images come from other datasets are about 3x smaller. In detail, the lowest-resolution image is 321×334 pixels. About the aspect ratio, the width and height of all images are not more than 10% different. Thus changing the images to square shape may not affect much to the image content. Follow [60], we uniformly resize all images to 256×256 pixels.

Clinical consideration for data augmentation

A good data augmentation not only being able to provide diverse morphological features but also preserve original features, therefore increasing the learning capacity of the deep learning model. In contrast, improper data augmentation can generate noise that will be harmful to the training phase. For instance, it is encouraged to apply rotations and flips for detecting a cat in an image, such as in ImageNet challenges. But for the purpose of chest X-ray image disease diagnosis, rotating images arbitrarily is not recommended. Furthermore,

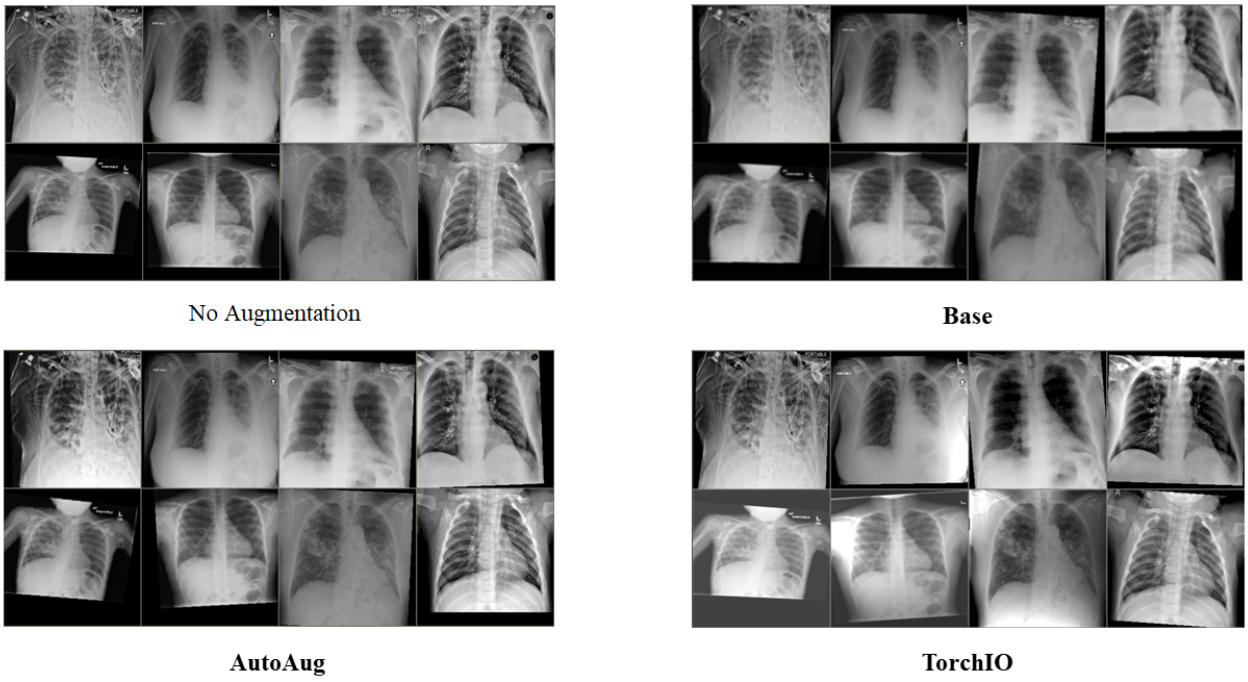


Figure 16: Visualization of 3 augmentation configurations. The **Base** augmentation is considered as a regular method. The **AutoAug** is the option to observe the impact of other transformations, such as sharpness and contrast. The last option, **TorchIO** consists of domain-specific transformations for medical images.

the accuracy of deep learning is heavily impacted by the rotation degree. In order to distillate the accepted geometric transformations for chest X-ray image, we follow the clinical considerations, which are referred in the work [61]:
1. Reflection. Both reflections in x-axis and y-axis are not recommended. Since putting the horizontal images as input is impractical and vertical flipping would leads to non-physiologic images (eg. heart in the right thorax rather than the left thorax).
2. Rotation. Applying slight rotations between $-5°$ and $5°$ are regularly seen in clinical practice, but severe rotations such as $\pm 45°$ or $\pm 90°$ are hardly to be encountered and leading noises to the learning model.
3. Scaling. An equal image scaling in x-axis and y-axis is possible with both large and small scaling. But scaling in only the x-axis or y-axis or severe scaling can be considered not recommended clinically.
4. Shearing Sheared images do not exists in medical image, thus this step is not recommended.
5. Translation. Translation could be a useful augmentation, due to the X-ray images do not always produce lungs in the center of the image. However, there is no clearly recommended range for translation.

We abide by these considerations above to uniformly apply the geometric transformations for all data augmentation configurations. Specifically are $\pm 5°$ rotation, slightly scaling within the range $[0.9, 1.1]$ and translation ± 0.1 in both x-axis and y-axis. Following the work [62], we encourage the blurring and also restrict the augmentations which make noise to the image.

In this work, we majorly study the effect of photometric augmentations to the learning capacity and

Base	AutoAug	TorchIO
GaussianBlur(std=[0.1, 2.0])	*Geometric transforms*	75% *Geometric transforms*
RandScale $[0.9, 1.1]^{\dagger}$	SVHNPolicy()	25% RandElasticDeformation
RandRotation $[-5°, 5°]^{\dagger}$		RandomBiasField [0.25]
RandXTranslation $[-0.1, 0.1]^{\dagger}$		RandomMotion [0.1]
RandYTranslation $[-0.1, 0.1]^{\dagger}$		

Table 3: The detail of 3 augmentation configurations, *Geometric transforms* is the set of transformations denoted with †. The TorchIO augmentation has 75% chance of applying this transformation, and remaining 25% chance of applying RandElasticDeformation.

performance of the learning model. In detail, we propose 3 configurations of data augmentation. **(1)** The base configuration consists of the geometric transformations as mentioned above and Gaussian blurring. **(2)** The modified version from AutoAug [63] policy, which is an automated approach to find the best-fit augmentation for large-scale datasets. Since the alphanumeric data partly has similar properties to chest X-ray images (eg. rotation and reflection for some specific characters do not preserve their meaning). Thus we collect and fine-tune the augmentation policy of SVHN [64] dataset. Specifically removing random coloring, posterizing, existing geometric transformations and pixel inverting. The remaining operations are the sharpness, brightness, contrast adjusting and the equalize transformation. **(3)** The configuration from TorchIO library [65], consisting of the intensity and spatial transforms for 2D medical image augmentation.

4.3 Training

Transfer learning

Transfer learning method was adopted to demonstrate the reusability of CNN pre-trained models. And it has shown good performance in carrying out various image classification tasks in previous research, without exception of medical diagnosis. In the experiments, we study the knowledge reusability of the large-scale pre-trained Transformer-based models on the ImageNet dataset for addressing this novel task. The experiment includes the comparison of the non-frozen and various-level frozen models to study the inheritance efficiency of generic feature extraction knowledge.

We focus on implementing the tiny and the larger but linear-complexity (respect to image size) versions of PVTv2 as a feature extraction head, they respectively are PVTv2_B0 and PVTv2_B2_li (whose specifications are depicted in table 4). Then for the new classification head, we employ 3 connected linear layers in conjunction with GeLU activation, dropout and batch normalization layers (see table 5).

Through experiments, we also determine which are the modules primarily receive the generic feature representation. As described in the table 4, Stage 3 in PVTv2_B2_li is in charge of the largest computation, it is a great advantage for our method if the knowledge in this Stage is useful for the novel prediction. Thus we make consideration for the set of 2 or 3 first Stages as two options for parameter freezing in the study. In general, PVTv2_B2_li consists of a stack of 21 Encoder blocks for 4 Stages and the trainable FC-layers. Its trainable parameters are 20.4M or 12.3M, respectively with the above freezing options compared to 22M overall. PVTv2_B0 has only 8 Encoder blocks and 3.5M parameters for the overall training.

Optimization and scheduler

In the training phase, the cross-entropy loss is adopted as the loss function. For each specific training, we follow 3 optimization approaches, the Adam optimization policy applies the allowed epoch limit as 20 and

	PVTv2_B2_li				PVTv2_B0			
Stage	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
L	3	4	6	3	2	2	2	2
C	64	128	320	512	32	64	160	256
S	4	2	2	2	4	2	2	2
N	1	2	5	8	1	2	5	8
P	7	7	7	7	-	-	-	-
E	8	8	4	4	8	8	4	4

Table 4: Configuration of the feature extraction head PVTv2_B0 and PVTv2-b2 with Linear complexity (denoted in the figure as PVTv2_B2_li) feature extraction head. These specifications: ***L, C, S, N, P, E*** are fully described in the section 3.2. PVTv2_B0 does not have ***P*** since due to not employing the Linear SRA. We can see that the difference of their scales lies in the size of outputs and the number of stacking blocks in each Stage (***L***).

Layer No.	**PVTv2_B2_li**	**PVTv2_B0**
1	Linear(512,64); GeLU(); Batchnorm(64); Dropout(0.5)	Linear(256,64); GeLU(); Batchnorm(64); Dropout(0.5)
2	Linear(64,16); GeLU(); Batchnorm(16); Dropout(0.3)	Linear(64,16); GeLU(); Batchnorm(16); Dropout(0.3)
3	Linear(16,3)	Linear(16,3)

Table 5: Configuration of the classification head respects to each feature extractor in table 4. The fully-connected layers are appied with the same input and output dimension, GeLU [52] activation and batch normalization.

the SGD & SAM_SGD applies this limit as 35, along with the uniform batch size is 12 and the early stopping patience as 4 for all training experiments:
1. AdamW. Training model with the Adaptive Moment Estimation (Adam) optimizer [38] with weight decay = 10^{-6} and apply the learning rate scheduler. The base learning rate = 3×10^{-4}; learning rate change ratio = 0.5; minimum learning rate = 5×10^{-6}.
2. SGD. Adaptive learning is not designed to find the best optimal result, its results is acceptable in many cases. But in the medical task that requires as best results as possible, the good parameter-tuned SGD optimizer can guide the model to reach more optimal results. In this work, we also apply the Stochastic Gradient Decent (SGD) optimizer with momentum: 0.9 and apply learning rate scheduler with another setting. The base learning rate = 1×10^{-2}; learning rate change ratio = 0.5; minimum learning rate = 1×10^{-4}.
3. SAM_SGD. Follow the work [40], we employ the training model with the Sharpness-Aware Minimization (SAM) optimizer [49] on SGD as the base, whose learning rate and momentum setting are: learning rate = 2e-3, learning rate change ratio = 0.55, change patience = 1 and minimum learning rate = 2e-4.

Experiments implementation

We propose 10 experiments to implement 2 scales of model PVTv2 with different fine-tuning statuses from the pre-trained models, 3 types of data augmentation approaches (Base, AutoAug and TorchIO). The AdamW optimizer is employed as the main approach due to its fast training ability, only two last experiments employ the others. The training and validation process is uniformly implemented with the material dataset and

division method introduced in section 4.2. About the experiments implementing model PVT2-b0, we do not apply freezing layers, due to some layers of the ImageNet pre-trained network is incompatible with its original configuration in source code. The trainable parameters versus total and other configurations of these experiments is depicted in table 6.

4.4 Evaluation methods

Similar to other works, the model performance is evaluated by using the common statistical measure confusion matrix, from that we obtained various metrics like accuracy and other epidemiological metrics are positive predictive value (shortened as PPV), sensitivity and F1-score. The description of the evaluation metrics are briefly described below:
(1) Accuracy: the metric evaluates the correctness of the model by measuring a ratio of accurately predicted cases out of total number of cases.
(2) Positive predictive value: the ratio of correctly predicted 'disease' to the total predicted 'disease'. High precision relates to a low false-positive rate, i.e. the model achieves high accuracy in the result it indicates.
(3) Sensitivity: is the probability that a test will correctly indicate 'disease' among those with the truly disease. The higher sensitivity, the higher rate of disease detection in the population. In the objective of investigating as most COVID-19 cases as possible, the best diagnosis model is the model that has the highest sensitivity metric with COVID-19 class with the acceptable PPV metric.
(4) F1-score: is the specific measurement in case of uneven class distribution especially with a large number of true negative observations. It provides a balance between metric (2) and (3). The macro average measures the F1-score of positive predictive value and sensitivity averaged across all labels, no matter of the amount of cases.

These included metrics can be defined as mathematical formulas below:

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{TP} + \text{FN} + \text{FP}} \quad (1)$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2)$$

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3)$$

$$\text{F1-score} = \frac{2 \times \text{PPV} \times \text{Sensitivity}}{\text{PPV} + \text{Sensitivity}} \quad (4)$$

where TP: true positive, represents the subjects correctly classified in predefined (positive) class. FN: false negative, represents the subjects mis-classified in the other (negative) class. FP: false positive, represents the subjects mis-classified in predefined (positive) class. TN: true negative, represents the subjects correctly classified in the other (negative) class.

4.5 Empirical results

Results on PVTv2_B0 experiments

With the start epoch defined as 0. Two first experiments: Exp.1 and Exp.2 implement the tiny version of PVT and their augmentation approaches respectively are AutoAug and Base. Their training sessions end at the 13^{th} epoch and they both gain their best accuracy equivalently in the validation set. Their metrics are slightly changed when the models perform in the test set, since it requires greater performance to COVID-19 cases. Specifically, the sensitivity to COVID-19 cases of Exp.1 is better than Exp. 2, but Exp. 2 can cover normal cases better than Exp. 1. Besides, the indication accuracy to Pneumonia and normal cases of Exp.1

is quite better than Exp.2 but it slightly weaker than Exp.1 about COVID-19 predictions. Detail of overall results is depicted in table 7.

Results on PVTv2_B2_linear experiments

We spent most of the duration for the experiments on PVTv2_B2_linear models. Figure 18 shows the training and validation loss from the Exp.3 to Exp.7.

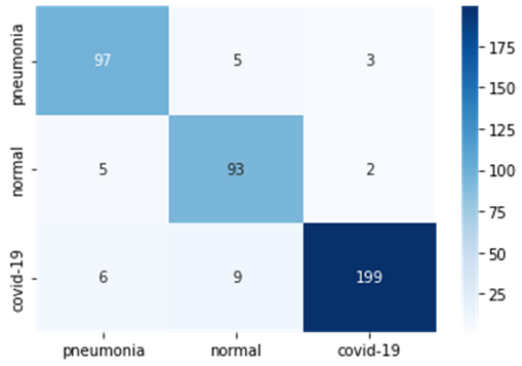


Figure 17: Confusion matrix of the Exp.5 model prediction on test set. The vertical axis is the ground truth and horizontal axis is the model prediction.

Exp.3 is a warm start for the series of fine-tuning model, which uses AutoAug augmentation approach. In Exp.3, this 2-Stage-frozen model got the high validation loss in the first epochs, it is progressively stable similarly to other experiment models in the following epochs. The training process achieves the greatest validation result at epoch 6 on the total 10 learning epochs (count from zero).

Exp.4 has the same augmentation and optimization as Exp.3, but further freezing an extra 3rd Stage. The model in Exp.4 gains slightly better accuracy and F1 results. The increase can be seen significantly in the sensitivity to Pneumonia. These results are the evidence for the conclusion that the most experimental knowledge of the PVT models is reusable for the novel tasks. The training process achieves the greatest validation result at epoch 5 on the total 9 learning epochs (count from zero).

Both of Exp.5 and Exp.6 have a great performance on the test set, they achieve the same overall accuracy. Specifically, the model in Exp. 5 has the highest sensitivity score to both COVID-19 and Pneumonia cases. In the Figure 17, we can see that the model in Exp.5 can recognize 315 respiratory cases on the overall 319, it almost completely separates the patients from normal people. It is noticed that the model is misclassified 9 cases of COVID-19, which is dangerous in the reality. In fact, the decision belongs to the doctor, who can consult these supportive AI models and clinical symptoms to conclude the final conclusion.

Exp.7 is the training experiment with SGD optimizer. The model on this experiment has a reduced loss slowly than others on both training and validation set. The training session takes 35 epochs to reach 82% accuracy as highest on the validation set and ends by early stopping on the next 5 epochs. The model seems to be stuck in the local minima. Therefore, it gains modest results on the test set.

Exp.8 is the further improvement with employing sequentially SAM_SGD, AdamW and SGD optimizer to lead the model avoid the local minima stuck. Specifically, we realize that the SGD optimizer with sharpness-aware minimizing navigates the model well, but the trade-off 2x time and severely more memory consumption. Thus, at epoch 12 we subsequently replace it with the AdamW optimizer, whose configuration is similar to the above. Then at epoch 20, we continue to replace with the momentum SGD optimizer with a large learning rate and strict scheduler to search for a global minima, but the result slightly increases and encounters over-fitting.

General discussion

Similar to Convolutional layers, the approach of MHSA layers also inherits the pre-trained knowledge well, resulting in more stable training progress and achieving better results on test set. Besides that, the Transformer-based model is more difficult to train due to the lack of convolution-like inductive bias. Through 8 experiments

Training settings				
Exp. No.	**Model**	**ImageNet fine-tuning stt.**	**Aug. type**	**Opt. type**
Exp. 1	PVTv2_B0	non-freeze	AutoAug	AdamW
Exp. 2	PVTv2_B0	non-freeze	Base	AdamW
Exp. 3	PVTv2_B2_li	freeze to Stage 2	AutoAug	AdamW
Exp. 4	PVTv2_B2_li	freeze to Stage 3	AutoAug	AdamW
Exp. 5	PVTv2_B2_li	freeze to Stage 3	Base	AdamW
Exp. 6	PVTv2_B2_li	freeze to Stage 3	TorchIO	AdamW
Exp. 7	PVTv2_B2_li	freeze to Stage 3	TorchIO	SGD
Exp. 8	PVTv2_B2_li	freeze to Stage 3	TorchIO	AdamW + SGD + SAM_SGD

Table 6: We implement 2 scales of PVTv2 on 8 experiments. This figure depicts their specific configurations.

Performance results on test set (%)								
Exp. No.	**Acc.**	**Sensitivity**			**PPV**			**Macro Avg. F1-score**
		Pneu.	Norm.	COVID-19	Pneu.	Norm.	COVID-19	
Exp. 1	88.54	86.67	86.0	90.65	82.72	83.50	94.17	87.25
Exp. 2	87.35	86.67	90.0	86.45	79.13	81.82	95.37	86.38
Exp. 3	89.98	83.81	96.0	90.18	87.12	82.76	95.54	89.04
Exp. 4	91.88	**92.38**	94.0	90.65	85.84	**87.85**	97.48	91.25
Exp. 5	**92.84**	**92.38**	93.0	**92.99**	89.81	86.91	**97.55**	92.05
Exp. 6	**92.84**	88.57	**99.0**	92.05	**93.94**	84.61	97.04	**92.30**
Exp. 7	81.62	75.24	93.0	79.44	87.78	65.95	90.42	80.93
Exp. 8	86.4	86.67	89.0	85.04	82.73	75.42	95.28	85.39

Table 7: The experimental results obtained from test set, with Acc. and PPV respectively denote the mean accuracy and positive predictive value for each class. Pneu. and Norm. respectively denote Pneumonia and normal cases.

with the objective of fast training, we suppose the PVT models are more compatible with the Adam optimizer to the SGD, although the validation loss is unstable and brings the high risk of stuck at local minima. One of our effort to overcome this drawback is applying sharpness-aware minimizing to SGD, this method may improve the training progress but the trade-off also occurs with higher computation and memory requirements. In the future work, we intend to use the stratified cross-validation and implement the SGD optimizer with suitable hyper-parameters to gain better results.

On the effect of data augmentation factors to the results on test set. The Exp.5 & Exp.6 model with the Base and TorchIO augmentation respectively give great results. It is possible to give the intuitive explanation that the property of CXR images is not to encourage neither the contrast, sharpness adjusting nor equalizing transformation due to these operations may affect the appearance of ground glass pattern, which represent the solidified lesion along with borders of blood vessels or bronchi inside the lung.

4.6 Heatmap generation

Based on the perspective of a student who knows nothing about clinical diagnostic imaging. We further want to understand how does the AI model can yield its clinical predictions. In order to help our PVT model show its visual explanations, we tune the GradCAM [66] algorithm to generate the heatmaps for predicted cases, which is relied on the knowledge of the last Linear-RSA layers. In general, the GradCAM uses the gradient of model prediction flowing into the Transformer blocks on the last Stage to produce a coarse localization map highlighting the important regions in the image. In this context, the highlight feature must be the ground glass pattern, which usually appears in respiratory-disease cases. Figure 19 shows several results on the test set, visualized with the raw images and the corresponding generated heatmaps, along with the prediction confidence. After observing these images, we conclude some different characteristics of the COVID-19 with Pneumonia and non-respiratory disease cases below:

(1) In the early stages, COVID-19 x-ray images may not appear unusual symptoms. In fact, the patient on the seventh day after the onset of the disease still did not have any abnormality on the x-ray image. That is a drawback of this approach due to incapable of recognizing people newly exposed to COVID-19.

(2) COVID-19 affects many small areas in the lungs at once, while many other cases of Pneumonia affect large areas in the lungs. There are 4 abnormalities that appear on COVID-19: Ground glass patterns appear with a higher density than the surrounding lung parenchyma, we can still see the border of the blood vessels or bronchi within the lesion. Ground glass opacities are less than 3cm in diameter, circular in shape, maybe solitary or scattered in the lung parenchyma. Bronchial wall thickening are lesions that represent thickening of the bronchial wall. Interlobular septal thickening predominantly appears in the lower lobes on both left and right lungs.

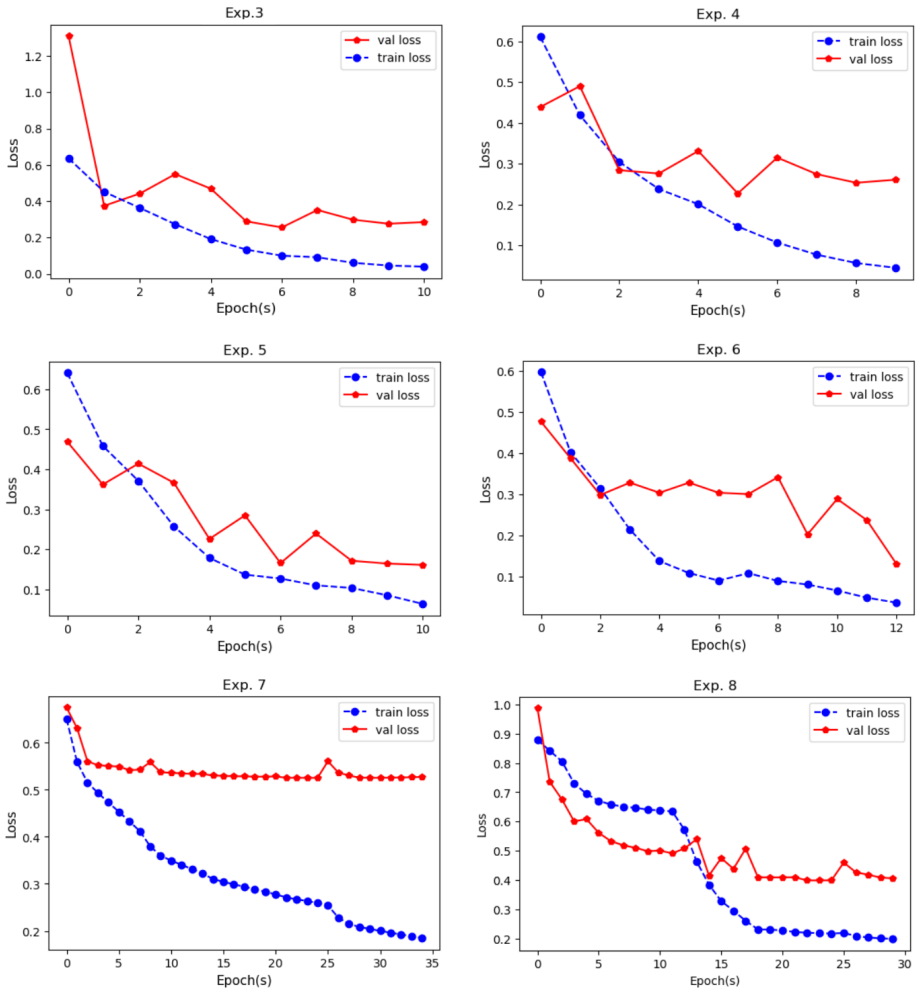


Figure 18: The graphs of training and validation loss from Exp.3 to Exp.8. The loss information of Exp.5 and Exp.6 experiments has been lost after the early stopping counting.

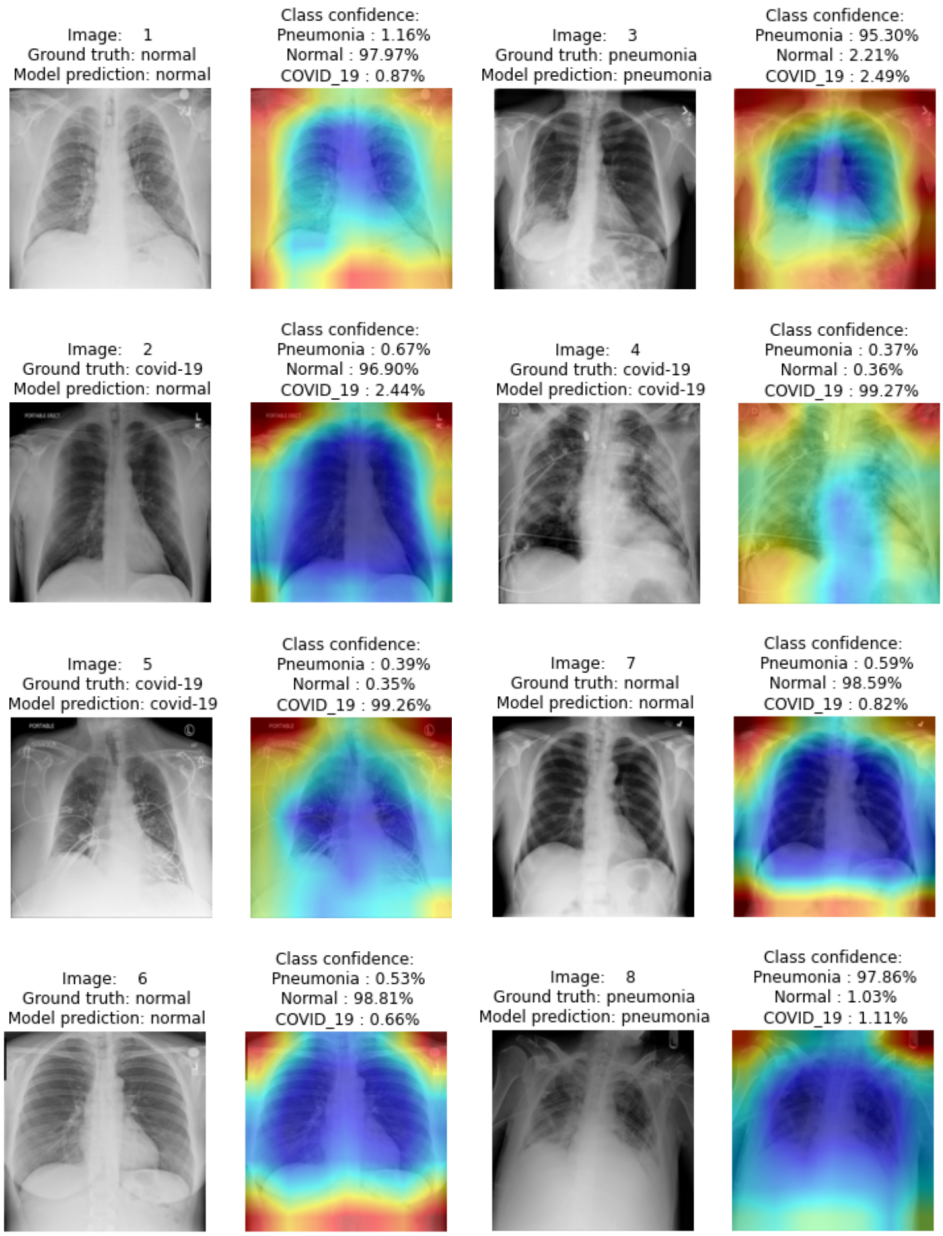


Figure 19: The prediction of Exp.5 model with its confidence score and heatmap generation for several cases in test set. Warmer colors represent more salient features area of the severe lesion.

5 Conclusion and future works

With an effort to seek the quick testing method based on deep learning method to repel the global epidemic and the curious to search for a novel approach of feature extraction, which differs from the rest related works. Through the sections, our study performs the theoretical comparison analysis of the well-known CNN, which its application covers almost all related works. And the ViT, inheriting the novel approach of the Transformers for the computer vision tasks. Results show that their approaches and their vital layers (the Convolutional and Self-Attention layer) have their own advantages and drawbacks. Therefore, the inheritance of both their properties simultaneously is considered as the efficient approach for researchers; and our proposed PVTv2 is one of these works that gains a great performance and robustness. In order to demonstrate this argument, we implement the model in the medical diagnosis task to overcome the four posed problems. Based on the prior knowledge on the pre-trained model as same as the CNN, we achieve the competitive results and considered to be reality-applicable. Along with the generated heatmaps, we hope this is a supportive tool for doctors and common users in order to check the respiratory status. For future works, we believe that model performance can be drastically improved by incorporating a larger COVID-19 chest X-ray image dataset. For our part, we further improve the performance of the models by performing k-fold cross-validation and implementing the optimizers to conclude whether SGD optimizer performs better in these cases. And last but not least, we also address the wonder that does the SAM_SGD optimizer really perform better at the trade-off of twice the training time.

References

[1] B. Sekeroglu and I. Ozsahin, "Detection of covid-19 from chest x-ray images using convolutional neural networks," *SLAS technology*, vol. 25, p. 2472630320958376, 09 2020.

[2] M. E. H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir, Z. B. Mahbub, K. Reajul Islam, M. Salman Khan, A. Iqbal, N. Al-Emadi, M. B. Ibne Reaz, and T. I. Islam, "Can AI help in screening Viral and COVID-19 pneumonia?" *arXiv e-prints*, p. arXiv:2003.13145, Mar. 2020.

[3] G. C. Bacellar, M. Chandrappa, R. Kulkarni, and S. Dey, "Covid-19 chest x-ray image classification using deep learning," *medRxiv*, 2021.

[4] L. Wang, Z. Q. Lin, and A. Wong, "Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images," *Scientific Reports*, vol. 10, no. 1, p. 19549, Nov 2020. [Online]. Available: https://doi.org/10.1038/s41598-020-76550-z

[5] G. A. Carpenter, "Neural network models for pattern recognition and associative memory," *Neural Networks*, vol. 2, no. 4, pp. 243–257, 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/089360808990035X

[6] D. Hubel and T. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, 1962.

[7] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, p. 84–90, May 2017. [Online]. Available: https://doi.org/10.1145/3065386

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[10] D. L. K. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo, "Performance-optimized hierarchical models predict neural responses in higher visual cortex," *Proceedings of the National Academy of Sciences*, vol. 111, no. 23, pp. 8619–8624, 2014. [Online]. Available: https://www.pnas.org/content/111/23/8619

[11] B. Ibrahim and L. Rabelo, "A deep learning approach for peak load forecasting: A case study on panama," *Energies*, vol. 14, no. 11, 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/11/3039

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv e-prints*, p. arXiv:1512.03385, Dec. 2015.

[13] J. Krohn, G. Beyleveld, and A. Bassens, *Deep Learning Illustrated: A Visual, Interactive Guide to Artificial Intelligence*, 1st ed. Addison-Wesley Professional, 2019.

[14] C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu, D. Ramanan, and T. S. Huang, "Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2956–2964.

[15] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," 2019.

[16] C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu, D. Ramanan, and T. S. Huang, "Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2956–2964.

[17] A. Azulay and Y. Weiss, "Why do deep convolutional networks generalize so poorly to small image transformations?" *arXiv e-prints*, p. arXiv:1805.12177, May 2018.

[18] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," 2016.

[19] V. Biscione and J. S. Bowers, "Convolutional Neural Networks Are Not Invariant to Translation, but They Can Learn to Be," *arXiv e-prints*, p. arXiv:2110.05861, Oct. 2021.

[20] R. Zhang, "Making Convolutional Networks Shift-Invariant Again," *arXiv e-prints*, p. arXiv:1904.11486, Apr. 2019.

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv e-prints*, p. arXiv:2010.11929, Oct. 2020.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *arXiv e-prints*, p. arXiv:1706.03762, Jun. 2017.

[23] J.-B. Cordonnier, A. Loukas, and M. Jaggi, "On the Relationship between Self-Attention and Convolutional Layers," *arXiv e-prints*, p. arXiv:1911.03584, Nov. 2019.

[24] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context," *arXiv e-prints*, p. arXiv:1901.02860, Jan. 2019.

[25] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," *arXiv e-prints*, p. arXiv:2005.12872, May 2020.

[26] N. Parmar, A. Vaswani, J. Uszkoreit, Łukasz Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," 2018.

[27] Y. Jiang, S. Chang, and Z. Wang, "TransGAN: Two Pure Transformers Can Make One Strong GAN, and That Can Scale Up," *arXiv e-prints*, p. arXiv:2102.07074, Feb. 2021.

[28] K. Lee, H. Chang, L. Jiang, H. Zhang, Z. Tu, and C. Liu, "ViTGAN: Training GANs with Vision Transformers," *arXiv e-prints*, p. arXiv:2107.04589, Jul. 2021.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *arXiv e-prints*, p. arXiv:1706.03762, Jun. 2017.

[30] S. Tuli, I. Dasgupta, E. Grant, and T. L. Griffiths, "Are convolutional neural networks or transformers more like human vision?" 2021.

[31] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "Imagenet-21k pretraining for the masses," 2021.

[32] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big Transfer (BiT): General Visual Representation Learning," *arXiv e-prints*, p. arXiv:1912.11370, Dec. 2019.

[33] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," 2017.

[34] S. d'Ascoli, H. Touvron, M. Leavitt, A. Morcos, G. Biroli, and L. Sagun, "Convit: Improving vision transformers with soft convolutional inductive biases," 2021.

[35] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" 2021.

[36] T. Xiao, M. Singh, E. Mintun, T. Darrell, P. Dollár, and R. B. Girshick, "Early convolutions help transformers see better," *ArXiv*, vol. abs/2106.14881, 2021.

[37] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers amp; distillation through attention," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 10 347–10 357. [Online]. Available: https://proceedings.mlr.press/v139/touvron21a.html

[38] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019.

[39] D. Zhou, B. Kang, X. Jin, L. Yang, X. Lian, Z. Jiang, Q. Hou, and J. Feng, "DeepViT: Towards Deeper Vision Transformer," *arXiv e-prints*, p. arXiv:2103.11886, Mar. 2021.

[40] X. Chen, C.-J. Hsieh, and B. Gong, "When Vision Transformers Outperform ResNets without Pretraining or Strong Data Augmentations," *arXiv e-prints*, p. arXiv:2106.01548, Jun. 2021.

[41] L. Xiao, J. Pennington, and S. S. Schoenholz, "Disentangling Trainability and Generalization in Deep Neural Networks," *arXiv e-prints*, p. arXiv:1912.13053, Dec. 2019.

[42] C. Gong, D. Wang, M. Li, V. Chandra, and Q. Liu, "Vision Transformers with Patch Diversification," *arXiv e-prints*, p. arXiv:2104.12753, Apr. 2021.

[43] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions," *arXiv e-prints*, p. arXiv:2102.12122, Feb. 2021.

[44] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in Transformer," *arXiv e-prints*, p. arXiv:2103.00112, Feb. 2021.

[45] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "CvT: Introducing Convolutions to Vision Transformers," *arXiv e-prints*, p. arXiv:2103.15808, Mar. 2021.

[46] J. Guo, K. Han, H. Wu, C. Xu, Y. Tang, C. Xu, and Y. Wang, "CMT: Convolutional Neural Networks Meet Vision Transformers," *arXiv e-prints*, p. arXiv:2107.06263, Jul. 2021.

[47] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *arXiv e-prints*, p. arXiv:1801.04381, Jan. 2018.

[48] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," *arXiv e-prints*, p. arXiv:1708.02002, Aug. 2017.

[49] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-Aware Minimization for Efficiently Improving Generalization," *arXiv e-prints*, p. arXiv:2010.01412, Oct. 2020.

[50] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," *arXiv e-prints*, p. arXiv:2103.14030, Mar. 2021.

[51] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "PVTv2: Improved Baselines with Pyramid Vision Transformer," *arXiv e-prints*, p. arXiv:2106.13797, Jun. 2021.

[52] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," *arXiv e-prints*, p. arXiv:1606.08415, Jun. 2016.

[53] J. Hou and T. Gao, "Explainable dcnn based chest x-ray image analysis and classification for covid-19 pneumonia detection," *Scientific Reports*, vol. 11, no. 1, p. 16071, Aug 2021. [Online]. Available: https://doi.org/10.1038/s41598-021-95680-6

[54] J. P. Cohen, P. Morrison, and L. Dao, "COVID-19 Image Data Collection," *arXiv e-prints*, p. arXiv:2003.11597, Mar. 2020.

[55] A. Chung, "Figure 1 covid-19 chest x-ray dataset initiative," https://github.com/agchung/Figure1-COVID-chestxray-dataset, 2019.

[56] ——, "Actualmed covid-19 chest x-ray dataset initiative," https://github.com/agchung/Actualmed-COVID-chestxray-dataset, 2019.

[57] R. Tawsifur, "Covid-19 radiography database," https://www.kaggle.com/tawsifurrahman/covid19-radiography-database, 2019.

[58] Radiological Society of North America, "Rsna pneumonia detection challenge," https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data, 2019.

[59] [Online]. Available: https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=70230281

[60] G. Jia, H.-K. Lam, and Y. Xu, "Classification of covid-19 chest x-ray and ct images using a type of dynamic cnn modification method," *Computers in biology and medicine*, vol. 134, pp. 104 425–104 425, Jul 2021, 33971427[pmid]. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/33971427

[61] M. Elgendi, M. U. Nasir, Q. Tang, D. Smith, J.-P. Grenier, C. Batte, B. Spieler, W. D. Leslie, C. Menon, R. R. Fletcher, N. Howard, R. Ward, W. Parker, and S. Nicolaou, "The effectiveness of image augmentation in deep learning networks for detecting covid-19: A geometric transformation perspective," *Frontiers in Medicine*, vol. 8, p. 153, 2021. [Online]. Available: https://www.frontiersin.org/article/10.3389/fmed.2021.629134

[62] Z. Hussain, F. Gimenez, D. Yi, and D. Rubin, "Differential data augmentation techniques for medical imaging classification tasks," *AMIA ... Annual Symposium proceedings. AMIA Symposium*, vol. 2017, pp. 979–984, 04 2018.

[63] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning Augmentation Policies from Data," *arXiv e-prints*, p. arXiv:1805.09501, May 2018.

[64] "The street view house numbers (svhn) dataset." [Online]. Available: http://ufldl.stanford.edu/housenumbers/

[65] F. Pérez-García, R. Sparks, and S. Ourselin, "Torchio: a python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning," *Computer Methods and Programs in Biomedicine*, p. 106236, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169260721003102

[66] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization," *arXiv e-prints*, p. arXiv:1610.02391, Oct. 2016.