

Truncating fined-tuned vision-based models to lightweight deployable diagnostic tools for SARS-CoV-2 infected chest X-rays and CT-scans

Francis Jesmar Montalbo¹

Received: 8 June 2021 / Revised: 5 October 2021 / Accepted: 25 January 2022 / Published online: 3 March 2022 © The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

In such a brief period, the recent coronavirus (COVID-19) already infected large populations worldwide. Diagnosing an infected individual requires a Real-Time Polymerase Chain Reaction (RT-PCR) test, which can become expensive and limited in most developing countries, making them rely on alternatives like Chest X-Rays (CXR) or Computerized Tomography (CT) scans. However, results from these imaging approaches radiated confusion for medical experts due to their similarities with other diseases like pneumonia. Other solutions based on Deep Convolutional Neural Network (DCNN) recently improved and automated the diagnosis of COVID-19 from CXRs and CT scans. However, upon examination, most proposed studies focused primarily on accuracy rather than deployment and reproduction, which may cause them to become difficult to reproduce and implement in locations with inadequate computing resources. Therefore, instead of focusing only on accuracy, this work investigated the effects of parameter reduction through a proposed truncation method and analyzed its effects. Various DCNNs had their architectures truncated, which retained only their initial core block, reducing their parameter sizes to <1 M. Once trained and validated, findings have shown that a DCNN with robust layer aggregations like the InceptionResNetV2 had less vulnerability to the adverse effects of the proposed truncation. The results also showed that from its full-length size of 55 M with 98.67% accuracy, the proposed truncation reduced its parameters to only 441 K and still attained an accuracy of 97.41%, outperforming other studies based on its size to performance ratio.

Keywords Coronavirus pneumonia \cdot Covid-19 \cdot Deep learning \cdot Deep convolutional neural networks \cdot Model truncation \cdot Medical image diagnosis

Francis Jesmar Montalbo francismontalbo@ieee.org

¹ College of Informatics and Computing Sciences, Batangas State University, Rizal Avenue Extension, Batangas, Batangas City, Philippines

Abbreviation	IS
AUPR	Area Under the Precision-Recall
AUROC	Area Under the Receiver Operating Characteristic
BFS	Base Features Size
BLL	Base Layer Length
BN	Batch Normalization
BPS	Base Parameter Size
BS	Batch Size
CCE _{loss}	Categorical Cross-entropy loss
CXR	Chest X-Rays
CAP	Community-Acquired Pneumonia
CT	Computerized Tomography
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Networks
GAP	Global Average Pooling
Grad-CAM	Gradient-weighted Class Activation Maps
LR	Learning Rate
MSLE	Mean Square Log Error
RT-PCR	Real-Time Polymerase Chain Reaction
ReLU	Rectified Linear Unit
RLRoP	Reduce LR on Plateau
SGD	Stochastic Gradient Descent
TFS	Truncated Feature Size
TLL	Truncated Layer Length
TPS	Truncated Parameter Size

1 Introduction

With the unceasing infection rates induced by the recent coronavirus SARS-CoV-2, the globe had brought itself to an unanticipated transformation. The virus, also known as COVID-19, targets the human respiratory system causing mild to severe impairment to its linings and air sacs, emanating various symptoms that confuse non-experts and experts with other related respiratory ailments [55]. With this dilemma, various testing procedures must commence to rule out the cause, whether the person has COVID-19, a Community-Acquired Pneumonia (CAP), or other respiratory diseases. One of the commonly used methods to justify the presence of the COVID-19 pathogens from a person involves using a Real-Time Polymerase Chain Reaction (RT-PCR) test, which is prone to false diagnoses. This approach involves collecting specimens from various body parts performed by swabbing a 6-in. Q-tip through a potentially infected individual [9, 61]. However, the process tends to cause fear for most people and medical collectors due to its unpleasant procedure and potential hazards. Not only that, the cost of testing for the COVID-19 comes a bit more expensive due to the lack of experts that can perform the task correctly and the inadequate availability of testing kits, mostly in developing countries [17, 48].

In specific instances, Chest X-Rays (CXR) and Computerized Tomography (CT) scans have become an alternative to aid the said limitation and uncomfortable process. Both methods can visualize the respiratory section where the virus usually thrives [3]. However, the lack of knowledge in interpreting COVID-19 from the said images radiates puzzlement even for medical experts due to its resemblance with other diseases like CAP, leading to false diagnoses [4].

Recently, the role of Deep Learning (DL) in medical diagnosis became one of the revolutionary steps for improving healthcare. DL took medical image diagnosis to a whole new level to help medical experts attain a faster and accurate diagnosis for COVID-19 infected individuals. The combined approach of computer vision and DL produced solutions that automated the assessment of COVID-19 infected CXRs and CT scans without the need for long hours and exhausting diagnosis, showing the possibility of yielding immediate and more accurate results [5, 41].

As presented in the study of Wang et al. [62], they began their initiative to diagnose CXRs by collecting data from various reliable sources. Their study had designed a specialized Convolutional Neural Network (CNN) [36] that recognizes the CXR images by using various interconnected layers that performed the extraction of learnable features from their dataset passed to a set of Fully Connected (FC) neural networks. Their proposed model consisted of 1 \times 1 convolutions (Conv) that strode over the image and produced smaller portions of the original image. The network then re-expanded the resized image with another 1 \times 1 Conv, this time, based on its depth. In addition, they also included a 3 \times 3 Depth-wise (DW) layer [8]. The projection then had another 1 \times 1 Conv, immediately followed by an extension with a similar-sized Conv layer. Their machine-driven approach led to successful results that diagnosed the three cases of CXRs as their proposed model, named COVID-Net, attained a parameter size of only 11.75 million and overall accuracy of 93.3%. Compared with other state-of-the-art Deep CNNs (DCNN) like VGG-19 [53] and ResNet50 [19], they only had 83% and 90.6% accuracy but required 20.37 M and 24.97 M parameters, respectively.

The pursuit for an ideal solution that can perform remarkably with less need for computing resources arose. Another study by Das et al. [11] proposed to further reduce the needed cost for the task without a significant reduction in performance to diagnose the three cases of CXRs sorted as COVID-19, CAP, and tuberculosis (TB) from China and the United States of America (USA). In their study, a well-known state-of-the-art DCNN model, the InceptionNet, had undergone the process of layer truncation that mostly eliminated its deeper layers and maintained its effectiveness towards feature extraction. The truncated architecture of their proposed InceptionNet model kept three of its core blocks intact and had its grid sizes reduced. The inception blocks are composed of Conv filters with $1 \times 1, 3 \times 3$, and 5×5 assembled in a DW fashion that produces a rich spectrum of features. However, using a pre-trained InceptionNetV3 [56] in their study required them to replace the final layers, as the original structure's design focused on classifying one thousand classes that had no relevance to their task. Therefore, they fine-tuned their model's ending layers to focus only on the three identified CXR cases. Once trained, they evaluated the results of the truncated model, which attained an accuracy of 99.92%. In addition, their work had about 2.1 M parameters, making it smaller than the previous COVID-Net model with better performance.

With the minimal adverse effects of truncation towards the performance of a DCNN model. Montalbo, F.J.P [42] employed another approach that dealt with layer truncation as a continuing study. However, this time, the study induced partial layer freezing and feature fusion. The selected DenseNet121 model [24] had its layers truncated, leaving nine dense blocks and a single transition block, making it 93% smaller than its original structure. A dense block contains a Batch Normalization (BN) [27], a Rectified Linear Unit (ReLU) [64], and 1×1 Conv, where it then concatenates to another up to a downsizing transition block. Such a concept paved the success of DenseNets to achieve state-of-the-art with lesser parameters than ResNets and InceptionNets. According to the study, the truncated DenseNet model had a mirrored version fused and trained through transfer learning, partial layer freezing, and fine-tuning to diagnose respiratory diseases, including COVID-19 [43]. Once trained and validated, the Fused-Densenet-Tiny produced an overall accuracy of 97.99% and a significantly low parameter requirement of only 1.2 M.

The studies mentioned had focused only on CXRs, which did not cover the diagnosis of lung CT scans infected by COVID-19. In another study by Gunraj et al. [18], they produced a variant of the COVID-Net called the COVIDNet-CT, designed primarily for diagnosing CT scans rather than CXRs infected by COVID-19. Based on their article, their CT-based model had an architecture that resembled the original COVID-Net that had a Pointwise (PW) Conv for dimensionality reduction, replication for an efficient dimensionality expansion, spatial feature representations through a DW Conv, and another expansion channel dimensionality with a PW Conv. The reliance of their model in Projection-Replication-Projection-Expansion (PRPE) (without strides) and PRPE-S (with strides) achieved a remarkable performance towards the task. Their COVIDNet-CT achieved an accuracy of 99.1% with only 1.4 M parameters that diagnosed uninfected, pneumonia-infected, and COVID-19 infected CT scans rather than CXRs.

Polsinelli et al. [47] also proposed a lightweight CNN model for CT scans with a similar goal. Their study diagnosed CT scans with and without COVID-19 using a model that resembles the SqueezeNet model [25], which has fifty times lesser parameters than the well-known AlexNet [34]. The SqueezeNet achieved such a feat through its "*Fire Modules*" consisting of 1×1 and 3×3 Conv layers and a ReLU concatenated towards a succeeding block. The original SqueezeNet had 8 Fire Modules with skip connections referred to as a "*bypass*." However, upon modifying the SqueezeNet model with expanded Fire Modules and additional Exponential Linear Units (ELU), they found that their model could perform better and more efficiently even without a BN layer. Also, they added a Transpose Conv layer at the final Fire Module to expand their features four times and have its weights summed through a 1×1 Conv with 128 depths of filters. As a result, their proposed model attained an accuracy of 85.03% with only 1.26 M parameters, making it a lighter yet competitive model against the studies of Wang et al. [63], Xu et al. [65], and Li et al. [38].

Due to the separated task of diagnosing the CXR and CT scans, Ibrahim et al. [26] decided to combine both into a single pipeline. One of their proposed models that attained the best performance had a composition of a VGG19 subsequently connected to a standard CNN model. Their proposed models trained with 500 and 800 epochs to achieve the best possible scores. Upon evaluation, they identified that their combined VGG19 and standard CNN model with 22.3 M parameters produced the best accuracy rate of 98.05%.

For an overview, Table 1 presents the proposed model, dataset specifications, performance results, and parameter sizes of the discussed recent related studies.

The studies presented regarding various models have shown remarkable performances but mostly towards diagnosing only either CXRs [11, 42, 62] or CT scans [18, 38, 47, 63, 65] using separate pipelines and had less focus on cost-efficiency. On the other hand, though the study of Ibrahim et al. [26] had both types of images combined into a single pipeline, their model still consisted of 22.3 M parameters, required long training epochs, and a lengthy architecture, making it less practical for most low and mid-end devices. After reviewing all the given studies, this work observed that most resolved the problem by either proposing new architectures or compounding multiple models and layers that eventually bloated the parameters without inducing much performance improvements. For that reason, specific devices may

Model	Dataset	Performance	Params
COVNet (ResNet50 backbone) [38]	CT Scans (COVID-19, CAP, Non-Pneumonia)	Sensitivity: 90% Specificity: 96%	25.6 M
Modified-Inception [63]	CT Scans (COVID-19, non-COVID-19)	Accuracy: 89.5%	≈23 M
VGG19 with Standard-CNN [26]	CT Scans and X-Ray (Normal, Pneumonia, COVID-19, Lung Cancer)	Accuracy: 98.05%	22.3 M
COVID-Net [62]	CXRs (Normal, CAP, COVID-19	Accuracy: 93.3%	11.75 M
Location-attention oriented Model (based on ResNet18) [65]	CT Scans (COVID-19, Influenza-A, Viral Pneumonia)	Accuracy: 86.7%	11.7 M
Truncated InceptionNet (3 blocks) [11]	CXRs (COVID-19, CAP, TB (China), TB (USA)	Accuracy: 99.92%	2.1 M
COVIDNet-CT [18]	CT Scans (COVID-19, non-COVID-19)	Accuracy: 99.1%	1.4 M
Lightweight CNN (based on SqueezeNet) [47]	CT Scans (COVID-19, non-COVID-19)	Accuracy: 85.03%	1.26 M
Fused-DenseNet-Tiny [42]	CXRs (Normal, COVID-19, CAP)	Accuracy: 97.99%	1.2 M

Table 1 Summary of recent related studies

*Arranged according to parameter size

experience problems in adapting their solutions or may become difficult to reproduce and deploy in developing countries that lack adequate resources [59]. As a solution, Das et al. and Montalbo, F. J. P. proposed layer truncation methods, where they have shown that even if DCNN models only used less of their layers and parameters, results can still become sufficient with the help of transfer learning and fine-tuning [11, 42, 43]. However, both of their work still had multiple blocks of layers activated, and that the fused solution required random weight generation from another mirrored pipeline that consumed additional training costs. The said findings from these existing studies showed that their current solutions still require a bit of a cost, as their models still had parameters >1 M.

Hence, this work proposed to further lessen the parameter size of state-of-the-art DCNNs into <1 M parameters and diagnose CXRs, and CT scans with COVID-19 infections in a single pipeline while achieving competitive performance. Furthermore, the said studies inspired the design of this work to become flexible enough to recognize both medical images without the need for complex procedures.

Through this work, the effects of a more robust truncation approach can produce new insights in training DCNN models regarding COVID-19. Like most studies, this work employed fine-tuning and pre-training of various DCNNs. However, this work had a better focus on regularization to retain a competitive performance even with fewer parameters. In addition, it is also worth mentioning that this work did not consider the use of lengthy training epochs, costly optimization algorithms, or heavy preprocessing to achieve such performance. Therefore, highlighting the value of the proposed method compared to most studies and providing leeway for better reproduction and deployment in the future. Based on recent literature, only a handful of researchers have dwelled upon this topic, which also became one of the inspirations to pursue this work as it can yield new insights and a broader perspective for other researchers working in a similar domain.

In summary, this work had achieved the following contributions:

 This work produced findings on the effects of a robust truncation method in various recent DCNN models that trained and diagnosed three cases of CXRs, and three cases of CT scans involving COVID-19.

- This work proposed a straightforward method that reduced the parameter size of the various DCNNs to <1 M by eliminating massive layers and retaining only their core conceptual block without sacrificing a significant fraction of their performance in diagnosing CXRs and CT scans in a single pipeline.
- Through stringent evaluations of the proposed method, this work attained competitive results. Concluding that pre-trained, truncated, and fine-tuned DCNNs can achieve better cost-efficiency, better reproducibility, a competitive performance, and a higher possibility of deployment than conventionally trained full-length DCNNs in low-end devices.

2 Materials and methods

2.1 Dataset acquisition and preparation

Collecting CXRs and CT scans, specifically those infected by COVID-19, can become cumbersome due to the limited data available in most healthcare facilities. Therefore, rather than collecting the desired samples and building a dataset from scratch, this work used readily curated datasets from existing reliable sources.

In this study, the curated dataset for the CT scans came from seven recently conducted studies involving COVID-19 [1, 10, 30, 44, 46, 49, 66], added with another set of CXR images from Sait et al. [51]. Upon investigation, the curated datasets already received preprocessing beforehand from their respective authors, alleviating the need to perform the task, as adding further changes may only lead to pre-determined outcomes, added bias, and even degrade the quality of the images. In addition, this study also focused on training the models based on realistic images rather than synthetically modified images to yield genuine findings. Figure 1 illustrates randomly selected image samples from the curated dataset.

As specified in Table 2, the CT scan dataset contains 6893 uninfected samples, 7593 COVID-19 samples, and 2618 pneumonia-infected samples, producing a total of 17,104 CT scan images. On the other hand, the CXR dataset has 3270 uninfected, 1281 COVID-19, and 4657 pneumonia image samples, having a total of 9208 CXR images. However, the given samples still required an appropriate distribution for training and validating the models. Therefore, this work had 80% of the total images or 21,052 images with six classes serving as train images, while the rest of the 20% or 5260 images with six classes as validation images.



Fig. 1 Samples of chest x-rays and computed tomography scans from the curated dataset

Having all six classes separated reduces the confusion between CXRs and CT scans when mixed up together in a single pipeline.

It is also worth mentioning that every individual's CT scan and CXR samples only entered either the train or validation dataset. Therefore, no identical images from the same individual existed on both datasets, preventing data leakage.

2.2 Selection and truncation of models

Due to the sizeable number of parameters found in most DCNNs, reproduction and deployment can become an issue for most low-end devices or infrastructures that lack adequate computing resources [7]. However, relying only on simpler models can become inefficient due to their lack of feature generation and regularization [22]. Hence, this work investigated the effects of the proposed truncation method in DCNNs to produce a lightweight model with a highly accurate classification capability.

Unfortunately, truncating complex DCNN models based solely on specific layers can become gruesome and impractical due to the vast possibilities. As observed from recent studies, they did not identify truncation points according to specific layers. Instead, they first focused on identifying truncation points based on the core block of a specific DCNN that composes it. Once identified, they removed multiple iterations of those core blocks to reduce their model's size [11, 42]. Unlike other studies, for further production of new findings, instead of leaving two, three, or only reducing a couple of core blocks from the original structure, this work had all DCNNs truncated into their simplest form, leaving them with a single core block that had parameters <500 K. This approach provides a whole new perspective to see if DCNNs can still perform the given task competitively, even after removing massive amounts of their layers. Based on the proposed truncation approach, this work managed to locate the first core block of each DCNN based on their respective articles. Eventually, the last layer from their first core block became their truncation point, having all successive layers after it eliminated. With that said, the parameter size of each DCNN reduced significantly but with their core design and identity intact. It is worth mentioning that no existing studies performed this approach to such a degree.

Beforehand, this work selected recently released and profound DCNN models based on either or both lengthy end-to-end network architecture and large parameter size. The candidates chosen for the experiments included the recent architectures like Xception [8],

Computed Tomography Scans				
Data	Train (80%)	Validation (20%)	Total	
CT-Normal	5515	1378	6893	
CT-COVID-19	6075	1518	7593	
CT-Pneumonia	2095	523	2618	
Total	13,685	3419	17,104	
Chest X-Rays				
CXR-Normal	2616	654	3270	
CXR-COVID-19	1025	256	1281	
CXR-Pneumonia	3726	931	4657	
Total	7367	1841	9208	
Overall	21,052	5260	26,312	

 Table 2
 Dataset specification and distribution

InceptionV3 [56], DenseNet121 [24], ResNet50V2 [20], InceptionResNetV2 [57], and EfficientNetB0 [58]. These models had trained in their full-length and truncated forms to unravel a wide range of findings.

2.2.1 InceptionV3

One of the candidates for this work, the InceptionV3, belongs to a group of a DCNN family that reconstructed the conventional size of Conv filters, from the usual 5×5 and 7×7 into a 1×7 and 1×5 , InceptionV3 added a 1×1 bottleneck right before a succeeding larger Conv filter. InceptionV3 achieved a cross-channel correlation through this approach, leading to a massive reduction of parameters than its predecessors. However, the InceptionV3, with its profound approach, still had a lengthy network structure that consisted of about 23 M parameters [56].

2.2.2 Xception

On the other hand, in an inverted version of the InceptionV3, the Xception model reduced its parameter size by utilizing DW and PW Conv layers. The given approach had the Inception's block structure reversed. It also became more expansive with a 1×1 PW Conv directed into a stack of concatenated linear DW Conv layers added with residual connections, resulting in fewer and simpler computations that improved its overall performance. Furthermore, the concept provided by Xception managed to reduce the original parameters of the InceptionV3 from 23 M to 22 M [8].

2.2.3 ResNet

The idea of providing more layers became a popular method since the release of AlexNet and VGGNets. However, the simple idea of adding more layers suddenly became a problem as the performance of deeper models began to saturate due to the vanishing gradient problem. As a solution, ResNet had the idea of utilizing skip connections, allowing the model to learn from residuals. Such a concept paved the way for succeeding models to improve their learning capability even after adding more layers. ResNet consists of residual blocks that contain a series of Conv \rightarrow BN \rightarrow ReLU and a skip connection that summates the feature sets, which added robustness. However, in a later study, ResNetV2 surpassed the previous performance of the original ResNet by interchanging its layer sequence into a BN \rightarrow ReLU \rightarrow Conv, the revised version also came with identity mappings that can now propagate signals through the skip connections back and forth. Due to its reliance on depth, like Xception, the ResNet50V2 variant also had a sizeable parameter size of 25 M [20].

2.2.4 InceptionResNetV2

With the concept of Inception and the emergence of ResNet, a model called the InceptionResNetV2 sprung. This model aggregated concepts into multiple Inception blocks with a corresponding ResNet block connected to a concatenating block. The core concept of the InceptionResNetV2 still follows the design of InceptionV3 but with added residual links. Thus, InceptionResNetV2 eventually became broader and deeper than InceptionV3, Xception, and even ResNetV2, which produced a massive parameter size of 55 M [57].

2.2.5 DenseNet

Unlike ResNets that used the summation of features, the concept of DenseNets relied on densely concatenated layers coupled in a block-wise manner. Each dense block consists of a sequential BN \rightarrow ReLU \rightarrow 1 \times 1 Conv and another with a 3 \times 3 Conv added with a downsizing transition layer that has a BN \rightarrow ReLU \rightarrow 1 \times 1 Conv \rightarrow 2 \times 2 Average Pooling (AP) that relieves the model from the massive surge of features, providing better computational efficiency.

The DenseNet121, one of the smallest family members in the DenseNet family, only needed 8 M parameters to operate through its profound design [24].

2.2.6 EfficientNet

Due to the demand and interest for better efficiency and scalability, EfficientNet had the concept of providing a scalable model that adapts a concept of balancing the width, depth, and resolution of images upon expansion while maintaining a low computational overhead. The model primarily consists of mobile-inverted bottlenecks or MBConv with a Squeeze-and-Excitation, BN, and the recent Swish activation function [50]. The selected base EfficientNetB0 consists of 16 MBConv blocks with varying sizes of receptive fields in this work. The model produces about 5.3 M parameters in its entirety, making it significantly smaller than DenseNet121 [58].

2.2.7 The specification of the truncated models

Even with the improved performance, and reinvented structures of various recent state-of-art DCNNs, this work still had the initiative to investigate whether these models require their entire layers to achieve adequate performance towards the identified task.

In Table 3, the following presents how the proposed truncation method affected each DCNN's feature size. As mentioned, this work identified each model's core block to determine their respective truncation points. From there, the selected DCNNs only had their first core blocks left due to the removal of succeeding layers after their identified truncation points.

Based on the given numbers, the models' Base Layer Length (BLL) with their original head intact significantly reduced after the truncation process, leading to their new Truncated Layer Length (TLL) without the head. This approach, however, also affected the Base Features Size (BFS) that they produce. With these changes, the performance of each DCNN can alter from their full-length version, as a new Truncated Feature Size (TFS) can potentially deteriorate their overall accuracy, which this work investigated.

Model	BLL	TLL	BFS	TFS	Last layer name
InceptionV3	313	41	2048	256	"mixed0"
Xception	134	16	2048	128	"add 1"
ResNet50V2	192	17	2048	256	"conv2 block1 out"
InceptionResNetV2	782	41	1536	320	"mixed 5b"
DenseNet121	429	14	1024	96	"conv2 block1 concat"
EfficientNetB0	240	46	1280	24	"block2b_add"

Table 3 Specifications of the base and truncated models

Table 4 shows the difference in parameter size of each model before and after the truncation process. After truncation, the Base Parameter Size (BPS) of each DCNN had a massive decrease, leaving each with a new Truncated Parameter Size (TPS) of <500 M. Such a significant reduction already guarantees that this work produced lighter models compared to the previously discussed studies. Noticeably, the InceptionResNetV2 and InceptionV3 preserved the most extensive parameters of around 400 K even with their shorter TLL of 41 than EfficietNetB0 with 46. The reason lies in their layer composition, as inception-based models rely on robust aggregation of multiple Conv blocks that made their structure broader than most models.

For a better visual understanding of the proposed truncation method, this work provides the schematics of the given models before and after truncation. The full-length version presents the original architecture of each DCNN model before the proposed truncation, while the truncated version presents what happened to each DCNN model after the proposed truncation.

- Full-Length https://github.com/francismontalbo/covid19_diagnosis_and_analysis_of_ truncated dcnns/tree/main/full length architectures
- Truncated https://github.com/francismontalbo/covid19_diagnosis_and_analysis_of_ truncated_dcnns/tree/main/truncated_architectures

2.3 Transfer learning and fine-tuning

Before any of the models became usable for the task, proper training and tuning commenced. Illustrated in Fig. 2, to alleviate the problem of having inadequate data, each model obtained pre-trained features from ImageNet through transfer learning, attaining leverage for better image recognition, and decreased costly generation of features from scratch. However, due to the unnecessary weights inherited from ImageNet, the models required fine-tuning. That way, the set of proposed fine-tuning layers can tailor-fit the DCNNs only to recognize the defined CXRs and CT scans. Once fine-tuned, the models re-trained all their layers with the prepared dataset, producing a new set of weights specifically for the task [31].

As shown in Fig. 3, because of the needed fine-tuning technique, this work replaced the ending layers of all the selected models to recognize the given classes, provide better regularization, and prevent overfitting [21]. The proposed sequential layers are composed of a Global Average Pooling (GAP) [39], dropout layer [54], and a 6-unit dense layer that represented the given classes activated by a softmax function [45] working as a classifier linked directly to each model's final layers, specified previously in Table 3.

Model	BPS	TPS
InceptionV3	23,851,784	429,440
Xception	22,910,480	55,712
ResNet50V2	25,613,800	84,480
InceptionResNetV2	55,873,736	441,920
DenseNet121	8,062,504	55,488
EfficientNetB0	5,330,571	24,345

Table 4 Comparison of parameter size before and after truncation



Fig. 2 The proposed transfer learning and fine-tuning method

The incorporated GAP layer served as a substitute for the conventional flattening layer used in most CNN models. The advantage contributed by GAP lies in its operation of managing the features for each corresponding case in the final dense layer, where it averaged the feature's pixel values globally and flattened it in a single vector for each class, fed directly to the following dense layer with a softmax activation. Hence, it provided each model with a more native image recognition approach, as the added GAP layer enforced the correspondence of feature sets towards the specified classes. Also, unlike the conventional flattened FC neural network, the GAP layer did not require parameters to operate, thus reducing the need for additional computational cost and providing summed-out spatial information that generated robust spatial translations for each input.

Empirically, a dropout layer can provide the random elimination of dense units and prevent a model from co-adapting too much to a given set of data, which drastically reduces overfitting. Therefore, this work also considered adding a dropout layer to operate between



Fig. 3 Proposed fine-tuning replacement layers for the truncated models

the previously connected GAP layer and the last prediction layer, as it can stochastically lessen the propagated feature weights per unit, which stimulates the model to perform better. After accomplishing each model's pre-training and fine-tuning, the compiled DCNN models had a specified set of hyper-parameter values, loss function, and a callback function intact.

2.4 Selected hyper-parameters

In a DL model, hyper-parameters operate as tweakable components that can intervene with the model's overall performance towards a specific task [69]. In this work, the configured hyperparameters in Table 5 did not come from a rigorous selection process or received help from an optimization algorithm. Instead, the presented hyper-parameter values relied on an empirical tuning approach, proper guidance from other studies, and suitability based on the machine's current specification during the time of experiments. Furthermore, such an approach entails that training the proposed truncated models should become straightforward [12]. Therefore, the configured hyper-parameters only included constant values for the Batch Size (BS), Learning Rate (LR), and epoch, set to reduce the training length while achieving a rapid convergence.

For the BS, the value of 16 gave a decent transfer rate of data through the pipelines and did not prompt memory exhaustion during experiments. In addition, with the need for faster convergence, this work relied on the Adam optimizer, as it can achieve convergence faster than the commonly used Stochastic Gradient Descent (SGD) and RMSprop [32]. As specified, Adam works with an adaptive momentum that made it a popular go-to optimization algorithm in various medical imaging studies that produced substantial results [33]. The said optimizer also consumes less memory than most existing optimizers as it selects subsets stochastically rather than the entire dataset simultaneously, making it efficient for DL models with tiny to large datasets. Furthermore, due to the given characteristics of Adam achieving faster convergence, this work necessitated a smaller LR to prevent each model from learning too fast and cause their performance to saturate during training. Further, setting a dropout rate of 0.5 provided a substantial regularization of the passing features through the fully connected layers, decreasing the possibilities of overfitting.

It is also worth mentioning that the configured values may vary from machine to machine and may involve re-tuning when replicating this work. Nonetheless, the configured BS of 16, employed Adam optimizer with a 0.0001 LR, training length of 25 epochs, and a dropout rate of 0.5, yielded the best results within a short period for this work.

2.5 Callback function

Once the training begins, the model and its hyper-parameters can no longer change. With that said, training each DCNN may become tedious and repetitive when producing the best

Hyper-Parameter	Value
Batch Size	16
Optimizer	Adam
Learning Rate	0.0001
Epochs	25
Dropout Rate	0.5

 Table 5
 Hyper-parameter configuration

possible results. However, relying on optimization algorithms for hyper-parameter tuning can become expensive and time-consuming, defeating the purpose of this work. Hence, to provide a better grasp and adaptability for the models during training without applying a heavy strain on computing resources, this work employed a callback function called, Reduce LR on Plateau (RLRoP) [15].

Presented in Table 6, the RLRoP configured in this work provides better control during training, in case that after two successive epochs that the model's validation accuracy had no improvements, the base LR will decrease by a factor of 0.5, which will then replace its base value (LR * 0.5) with a limit of up 0.000001. Through this method, each model will have the ability to automatically adjust their LR even during training each time their validation accuracy stagnates at a particular epoch, increasing the possibilities of achieving further improvements with less expense [14].

2.6 Loss function

In line with the desire to achieve high accuracy rates, DL models also require evaluating their errors to identify how well they fit the given data. Error rates also differ from accuracy based on measurement, as loss or errors require lower values to signify better performance, unlike accuracy that compels higher values to imply that the model performed well [16].

Apart from hyper-parameters, selecting an appropriate loss function plays a vital role in assessing and improving the overall performance of DCNNs. Due to the three cases of CXRs and three cases of CT scans, the Categorical Cross-entropy loss (CCE_{loss}) for the softmax classifier became an ideal choice compared to a binary cross-entropy [40]. In Eq. (1), *C* indicates the six cases, including Normal, COVID-19, and Pneumonia for CXR (3) and CT scan (3). Every *C* diagnosed had its loss computed accordingly from each observation *o* based on their ground truth values *y*. For every diagnosis *d* performed in each instance *m* of *C*, the model calculates the number of mistakes on every *o* using a natural log function.

$$CCE_{loss} = \sum_{m=1}^{C} y_{o,i} \times log(d_{o,m})$$
(1)

2.7 Evaluation tools and metrics

This work selected and employed the standard evaluation metrics used in most DL tasks that initialized comparable results with other studies. In addition, other visualization tools also played a vital role in this work that differentiated the performance between the selected model's truncated and full-length forms.

Value
"Validation accuracy" 0.5
2 0.000001

Table 6 Learning rate reduction on plateau callback configuration

The trained models had performed their validation based on the 5260 samples with six classes. For every sample diagnosed, this work identified each diagnosis as either True Positive (t_p) , True Negative (t_n) , False Positive (f_p) , or False Negative (f_n) . In widespread practice, a diagnosed sample tagged as a t_p indicates that the model had a correct diagnosis of a specific infection, whether Pneumonia or COVID-19, while t_n indicates a correct diagnosis of a non-infected case. On the other hand, indicators like f_p and f_n determined that the model committed incorrect diagnoses, confusing a non-infected case with the other infections and vice-versa [60]. Due to the combined cases of CXRs and CT scans in a single pipeline, the t_n and t_p became indirectly unnecessary in this work as a multi-class approach does not directly justify each as a t_p or t_n . It is worth mentioning that such occurrence also happened in other studies as well [35]. After the models diagnosed all the validation samples, using the calculations below, this work identified each model's overall accuracy (*accuracy*), precision (pr), recall (r), and f_1 -score (f_1) . [23].

$$accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}$$
(2)

$$pr = \frac{t_p}{t_p + t_n} \tag{3}$$

$$r = se = \frac{t_p}{t_p + f_n} \tag{4}$$

$$f_1 = 2 \times \frac{pr \times r}{pr + r} = \frac{2t_p}{2t_p + f_p + f_n} \tag{5}$$

Further, each validated DCNN had its performance analyzed through various data visualization techniques using learning curves, a confusion matrix, Area Under the Receiver Operating Characteristic (AUROC), Area Under the Precision-Recall (AUPR), Mean Square Log Error (MSLE), and the Gradient-weighted Class Activation Maps (Grad-CAM).

3 Experimental results and discussion

3.1 Model convergence

With the help of learning curves, all models had their performance monitored that determined how well they learned from the given datasets. It is worth noting that the accuracy graph with an upward line indicates an improving model. In contrast, the loss graph measured by the CCE_{loss} illustrates a plummeting line, implying that the model also kept improving over time. Furthermore, models with a gradual yet progressive converging train and validation lines indicate an ideal fit and that the model did not experience over or underfitting. Otherwise, the presence of diverging lines can indicate poor generalization from future foreign test data [68].

Figure 4 illustrates the learning curves produced by the full-length version of the selected DCNNs. As observed, the models achieved high accuracies and low error



Fig. 4 Results of learning curves from the full-length models. a DenseNet121. b EfficientNetB0. c InceptionResNetV2. d InceptionV3. e ResNet50V2. f Xception

rates. However, their accuracy rates converged too early, which entailed a stagnant performance. Also, based on their loss graphs, a noticeable divergence of results transpired at the early epochs that radiated until the end, concluding that the fulllength DCNNs experienced fitting issues.

However, in Fig. 5, the truncated versions had lesser performance than their base models but with better progress and convergence. As presented, the accuracy and loss graphs of (c) to (f) attained full convergence. Although (a) and (b) both had an observable divergence in their losses, compared to the results of their full-length counterparts, the truncated versions still achieved fewer errors and better control against overfitting, indicating a significant improvement from the proposed truncation method applied.

3.2 Confusion matrix

The proposed truncation method showed relevant improvements in how the models fitted with the train and validation data. However, the previously presented graphs still inhibited questions about how well the models performed with their diagnoses on each validation sample. Therefore, this section visualized the diagnoses of each model in a more straightforward fashion using a confusion matrix. A confusion matrix is a standard tool used in most classification tasks to present the number of t_p , t_n , f_p , and f_n that can serve as a basis for each model's overall performance [60].

Figure 6 presents the diagnostic results from each model using a confusion matrix. On this point, the labels presented start from 0 to 5 that corresponds to the given classes: 0-Normal



Fig. 5 Accuracy and loss curves after the employment of the proposed truncation method. a DenseNet121-Tr. b EfficientNetB0-Tr. c InceptionResNetV2-Tr. d InceptionV3-Tr. e ResNet50V2-Tr. f Xception-Tr



Fig. 6 Classification results of the truncated models based on a confusion matrix

CXR, 1-COVID-19 CXR, 2-Pneumonia CXR, 3-Normal CT scan, 4-COVID-19 CT scan, and 5-Pneumonia CT scan. As evaluated, based on the number of correctly diagnosed samples within the vertical trajectory, other truncated models managed to achieve adequate performance. Unfortunately, other models did not respond quite well with the truncation method, like the DenseNet121-Tr and Xception-Tr, as seen on their darker panels beyond the identified trajectory. Though the EfficientNetB0-Tr and ResNetV2-Tr models still had a better performance than them, unlike the rest, the InceptionResNetV2-Tr and InceptionV3-Tr kept a significant amount of their diagnostic prowess even after truncation, as both received the least misdiagnosed samples.

3.3 Sensitivity versus specificity

For a more in-depth evaluation, this work graphically identified the trade-offs between the sensitivity and specificity of the models using the AUROC. For example, a DCNN model with a higher AUROC entails better performance, while an AUROC of <0.5 means that a DCNN cannot genuinely discriminate or identify a specific case [2].

Presented in Fig. 7, all truncated DCNNs had remarkable results across all their given classes despite having lesser parameters. Notably, the InceptionResNetV2-Tr and InceptionV3-Tr both attained similar AUROCs of 1.00 across all areas, showing dominant performance among the rest of the truncated models. On the other hand, the DenseNet121-Tr had the lowest macro-average AUROC of 0.96 and had shown to have noisy fluctuations on its graph. As observed, the perpetrator roots from its lowest AUROC of 0.91 from CT scans with COVID-19 infections at the lower thresholds. The adverse effects of the truncation also had perceptible oscillations in the graphs of the EfficientNetB0-Tr, ResNet50V2-Tr, and Xception-Tr.



Fig. 7 Receiver operating characteristic of the truncated models

Though it appears that the InceptionResNetV2-Tr and InceptionV3-Tr had a highly similar performance, further observations and analysis can still induce a better conclusion to identify which between the two suffered the least.

3.4 Precision versus recall

Another commonly used graphical metric, the AUPR curve, entails a more accurate perception of each model's performance due to the unbalanced distribution of data in this work [29]. Compared to the AUROC, the AUPR focuses more on the number of incorrect diagnoses. On the other hand, like AUROC, the AUPR measurement also pertains to the region covered underneath the curve.

From the results illustrated in Fig. 8, both InceptionResNetV2-Tr and InceptionV3-Tr had a similar micro-average AUPR of 0.996. However, upon observation, the InceptionResNetV2-Tr had no values <0.99, whereas InceptionV3-Tr had an AUPR of 0.988 from the Normal CXR or class 0. Therefore, indicating a slight advantage for InceptionResNetV2-Tr. In addition, other truncated models had also shown significantly lesser performance, where DenseNet121-Tr once again had the lowest performance with an 0.808 AUPR from COVID-19 CT scans or class 4, showing its difficulty in performing the diagnoses. Moreover, even though the InceptionResNetV2-Tr had shown a minimal advantage over the InceptionV3-Tr, they still had a similar micro-average AUPR score, requiring further investigations.



Fig. 8 Precision-recall curves of the truncated models

3.5 Error rates

In this section, an additional metric further analyzed the effects of the proposed truncation method. Using the MSLE, this work identified which model had fewer errors during their validation phase. The following also included the comparison of each model's MSLE from their original full-length and truncated versions. Though not commonly used in this type of task, the MSLE still provides additional knowledge to the current experiment as the results can still dynamically change upon reproduction and inclusion of more data in the future.

The calculation of the MSLE lies on (6), where \hat{y}_i serves as the predicted value of the *i* sample to a corresponding ground truth y_i estimated over a set of $n_{samples}$ from the validation set with a $\log_e(x)$ that functions as the natural log of x [13, 28].

$$MSLE\left(y,\widehat{y}\right) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples-1}} \left(\log_e(1+y_i) - \log_e\left(1+\widehat{y}_i\right)\right)^2 \tag{6}$$

The calculated MSLE of each model illustrated in Fig. 9 shows that the full-length DenseNet121 had the lowest MSLE, landing only at 0.76%, followed by InceptionResNetV2 with 0.84%. Surprisingly, after the truncation process, DenseNet121-Tr suddenly skyrocketed with a 5.26% MSLE and became the least performing model, while InceptionResNetV2-Tr still retained a convincing diagnostic capability with only a 1.47% MSLE. Upon analysis, the employment of the proposed truncation method affected the DenseNet121-Tr the most. Though all models experienced similar effects of performance reduction, the InceptionResNetV2-Tr had shown to deteriorate the least from its adverse effects.

3.6 Summary of overall performance

To summarize the overall performance, Table 7 presents the evaluated results of the truncated models based on *accuracy*, pr, r, and f_l .

Upon evaluation, InceptionResNetV2-Tr achieved the highest overall performance across all metrics with 97.41% accuracy, 97.59% *pr*, 97.52% *r*, and 97.55% f_{l} followed by InceptionV3-Tr with 97.36% accuracy. As analyzed, unlike other models, the performance of both InceptionResNetV2-Tr and InceptionV3-Tr did not deteriorate much from the truncation process, granting them a valuable trade-off and significant benefits in solving the given task.



MSLE Full-Length MSLE Truncated

Fig. 9 Results comparison between the full-length and truncated models based on the mean squared log error (lower the better)

Model	accuracy	pr	r	f_{I}
DenseNet121-Tr	77.98%	80.86%	76.72%	78.25%
EfficientNetB0-Tr	86.03%	88.44%	86.45%	87.34%
InceptionResNetV2-Tr	97.41%	97.59%	97.52%	97.55%
InceptionV3-Tr	97.36%	97.46%	97.40%	97.43%
ResNet50V2-Tr	86.12%	87.77%	86.78%	87.25%
Xception-Tr	78.82%	81.11%	80.43%	80.25%

Table 7 Overall performance of the truncated models

3.7 Class activation maps and saliency

For better interpretability, this section investigated the class activations of all DCNNs before and after the proposed truncation method to better understand how they diagnosed a specific sample. Using gradient-based activation map algorithms provided fascinating outcomes of how each model interpreted various arbitrarily selected samples.

As illustrated in Figs. 10 and 11, a comparison of interpretations generated by the Grad-CAM [52] and the Grad-CAM++ [6] algorithms presented the distinct



Fig. 10 Gradient-weighted class activation maps of the full-length baseline models: DenseNet121 (a), EfficientNetB0 (b), InceptionResNetV2 (c), InceptionV3 (d), ResNet50V2 (e), Xception (f)



Fig. 11 Gradient weighted class activation maps of the truncated models: DenseNet121-Tr (a), EfficientNetB0-Tr (b), InceptionResNetV2-Tr (c), InceptionV3-Tr (d), ResNet50V2-Tr (e), Xception-Tr (f)

elucidations of each full-length and truncated DCNN. Upon examination, compared to their original full-length counterparts, truncated DCNNs had more dispersed attention, portraying the aftereffects of truncation on their interpretation capability. However, among the rest, the truncated InceptionResNetV2-Tr and InceptionV3-Tr still yielded better interpretability and finer localizations, instilling that these models still preserved a massive portion of their image recognition capability. Therefore, supporting that the proposed truncation applied on both DCNNs only brought minimal vision impairment in their diagnostic performance towards the task, showing the effectiveness of their robustly concatenated architecture reflected based on their overall performance in Table 7.

It is worth mentioning that Grad-CAMs cannot entirely provide perfect localizations due to their drawbacks compared to an actual segmentation algorithm, as they work for different purposes [37]. Nonetheless, the Grad-CAM algorithms somehow provided adequate visualizations to understand how the proposed truncation process affected the DCNN's interpretation of the given cases.

3.8 Practicality and ease of deployment

To further justify the potential of truncated DCNN models over conventionally trained fulllength DCNN models, this work presents a bubble chart that compares their differences in terms of overall accuracy, parameter size (bubble size), and FLOPS [67], added with a table showing their practicality in terms of storage consumption.

3.8.1 Cost over performance

As illustrated in Fig. 12, the truncated models became tiny dots compared to full-length models due to their significantly smaller parameter sizes. In addition, looking at the best performing model's FLOPS, the truncated InceptionResNetV2-Tr only required 3.35 GFLOPS with its 440 K parameters yet attained a 97.41% accuracy, making it the most accurate and cost-efficient truncated model that competes highly against full-length models. On the other hand, though the full-length EfficientNetB0 had a 1.32% higher accuracy of 98.73% and 2.56 lesser GFLOPS of 0.79, this work identified that it still required 4.06 M parameters, making it 3.63 M more parameters than the InceptionResNetV2-Tr. Thus, giving the notion that InceptionResNetV2-Tr has significant potential in terms of deployment, reproducibility, and viability even across low to mid-end devices compared to most full-length DCNN models.

3.8.2 Storage consumption

In Table 8, though the full-length EfficientNetB0 initially had a low disk consumption of 48,072 KB and 98.73% accuracy, the truncated InceptionResNetV2-Tr only consumed 5338 KB, yet performed with a 97.41% accuracy with only a minimal performance difference of 1.32%. Furthermore, compared with the full-length DenseNet121 that had an accuracy of



Fig. 12 Overall comparison of cost-efficiency to performance ratio

Model	Full-length		Truncated	
	Size in KB	acc	Size in KB	acc
DenseNet121	83,446	98.94%	714	77.98%
EfficientNetB0	48,072	98.73%	405	86.03%
InceptionResNetV2	638,982	98.67%	5338	97.41%
InceptionV3	256,430	98.40%	5188	97.36%
ResNet50V2	276,649	98.48%	1081	86.12%
Xception	244,766	98.57%	739	78.82%

Table 8 Comparison of storage consumption to performance ratio between full-length and truncated models

98.94% and disk consumption of 83,446 KB, the truncated InceptionResNetV2-Tr landed with 1.53% lesser accuracy but with far lower storage requirements. Therefore, emphasizing InceptionResNetV2-Tr's viability and practicality over full-length DCNN models for future deployment with its lighter storage consumption and significantly accurate performance.

3.9 Discussion

As presented, the InceptionResNetV2-Tr achieved the best overall performance among the rest of the truncated DCNN models. Through a comprehensive investigation, InceptionResNetV2-Tr preserved the highest TFS of 320 and second most TLL of 41 next to the full-length EfficientNetB0 with 46. Even with its massive full-length structure, the proposed truncation process reduced its parameter size from 55 M to only 441 K, producing an overall accuracy of 97.41%. In addition, before identifying the proposed hyper-parameter settings to produce the presented results, this work also considered other hyper-parameter configurations to train the truncated models.

To yield broader findings, Table 9 presents the prior settings of hyper-parameters evaluated during the experiments. As shown, the following had different combinations of optimizers, dropout rates, and LRs, which in theory have the most significant effects on the learning process of a DL model. In addition, the following had different dropout rates of 1.0, 0.3, and 0.5. Finally, to further identify the best possibilities without entirely relying on costly optimization algorithms, the empirical search for hyper-parameters also had the truncated models trained without a dropout or, in its least tuned form, and different optimizers like SGD,

Description	Optimizer	LR	Dropout rate
Setting-1 (a)	SGD	0.01	0.1
Setting-1 (b)			0.3
Setting-1 (c)			0.5
Setting-1 (d)			No dropout (Untuned)
Setting-2 (a)	RMSprop	0.001	0.1
Setting-2 (b)	1 1		0.3
Setting-2 (c)			0.5
Setting-2 (d)			No dropout (Untuned)
Setting-3 (a)	Adam	0.0001	0.1
Setting-3 (b)			0.3
Setting-3 (c)			0.5
Setting-3 (d)			No dropout (Untuned)

Table 9 Hyper-parameter settings across various combinations

RMSprop, and Adam with their respective LRs. This study did not cover other LR besides the given, as an RLRoP is already at play, reducing time and cost to produce the desired results.

In Table 10, the InceptionResNetV2-Tr trained with Setting-1 (c) yielded the best performance of 94.14% accuracy. The second-best performing model, the InceptionV3-Tr, had a 93.94% on the same setting, showing that truncated inception-based networks, even with the classic SGD algorithm combined with a dropout rate of 0.5, can still perform better than the other truncated models.

In Table 11, the best results landed with Setting-2 (a), which again became the InceptionResNetV2-Tr with a 97.36% accuracy. Surprisingly, the EfficientNetB0-Tr had a 93.44% accuracy on (b), outperforming the previous InceptionV3-Tr that only reached a top performance of 92.36% accuracy on (d). Nonetheless, unlike the rest, the InceptionResNetV2-Tr still showed a dominant performance as it lands with accuracies >93% across (a) to (d).

In Table 12, the highest overall accuracy of 97.41% came from Setting-3 (c) InceptionResNetV2-Tr, which had an Adam optimizer, 0.0001 LR, and 0.5 dropout rate. Therefore, proving the effectiveness of the proposed hyper-parameter configurations compared to the rest of the given settings in this work.

Aside from the InceptionResNetV2-Tr, the InceptionV3-Tr also had a remarkable response, as it attained an accuracy of 97.36% from the same Setting-3 (c), making it only 0.05% less accurate and about 150 KB less costly than the InceptionResNetV2-Tr. For further comparison, the InceptionV3-Tr had an exact TPS of 429,440 while the InceptionResNetV2-Tr had 441,920, showing a parameter size difference of 12,480 due to InceptionV3-Tr's lesser TFS of 256. Though the other truncated DCNN models had fewer parameters than the InceptionResNetV2-Tr, most had lesser performances. The DenseNet121-Tr, EfficientNetB0-Tr, ResNet50V2-Tr, and Xception-Tr all experienced substantial

Model	Overall accuracy (%)			
	<i>(a)</i>	(b)	(c)	(d)
DenseNet121-Tr	65.8	64.33	55.78	70.78
EfficientNetB0-Tr	89.03	78.44	86.75	87.19
InceptionV3-Tr	92.36	91.52	93.94	89.39
ResNet50V2-Tr	77.83	65.27	51.81	39.81
Xception-Tr	72.22	77.03	59.41	79.87
InceptionResNetV2-Tr	90.78	93.16	94.14	91.05

Table	10	Setting-1	results
-------	----	-----------	---------

Table 11 Setting-2 results

Model	Overall accuracy (%)				
	<i>(a)</i>	<i>(b)</i>	(c)	(d)	
DenseNet121-Tr	77.15	80.99	77.62	82.53	
EfficientNetB0-Tr	85.34	93.44	82.68	89.01	
InceptionV3-Tr	86.84	65.36	91.71	92.36	
ResNet50V2-Tr	70.87	85.89	46.44	74.92	
Xception-Tr	81.37	74.37	77.41	86.14	
InceptionResNetV2-Tr	97.36	96.88	93.4	95.57	

Model	Overall accuracy (%)				
	<i>(a)</i>	<i>(b)</i>	<i>(c)</i>	(d)	
DenseNet121-Tr	79.07	80.06	77.98	76.5	
EfficientNetB0-Tr	88.40	88.31	86.03	90.74	
InceptionV3-Tr	94.54	96.58	97.36	93.99	
ResNet50V2-Tr	89.01	85.10	86.12	80.10	
Xception-Tr	77.30	82.36	78.82	82.38	
InceptionResNetV2-Tr	95.99	96.73	97.41	95.29	

Table 12	Setting-3	results
----------	-----------	---------

deteriorations. Upon investigation, the DenseNet121-Tr had the least TLL of 14, followed by Xception-Tr with 16, ResNet50V2-Tr with 17, and EfficientNetB0-Tr with 46. Thus, even with the advantage of having the most TLL, EfficientNetB0-Tr still had an inferior performance than InceptionResNetV2-Tr, InceptionV3-Tr, and even ResNet50V2-Tr. However, based on TPS, the EfficientNetB0-Tr only had 24 K parameters that generated an accuracy of 86.03%, while ResNet50V2-Tr still had about 84 K parameters with 86.12% accuracy. Therefore, instigating that TLL and parameter size did not entirely affect their overall performance. Based on another perspective, the EfficientNetB0-Tr had the least TFS of only 24, while the DenseNet121-Tr had 96, Xception-Tr had 128, and ResNet50V2-Tr had 256, showing that TFS also had no direct effect on the performance. The most dominant factor observed in how the InceptionResNetV2-Tr and InceptionV3-Tr gained better performances than the rest trace back to their architectural design, as they both had the most aggregated layers and parameters in a consolidated block connected into the proposed added ending layers. Unlike EfficientNetB0-Tr, which had the largest TLL of 46, it still failed to surpass both InceptionV3-Tr and InceptionResNetV2-Tr, as it draws performance based on a lengthier architecture rather than a broader architecture. The said finding also reflected with the other models like DenseNet121-Tr, ResNet50V2-Tr, and Xception-Tr.

From a broader standpoint, Table 13 compared this work with the previously discussed studies. The proposed InceptionResNetV2-Tr yielded a competitive 97.41% accuracy even

Model	Dataset	Performance	Params
InceptionResNetV2-Tr	CXRs and CT scans (Normal, COVID-19, Pneumonia)	Accuracy: 97.41%	441 K
COVNet (ResNet50 backbone) [38]	CT Scans (COVID-19, CAP, Non-Pneumonia)	Sensitivity: 90% Specificity: 96%	25.6 M
Modified-Inception [63]	CT Scans (COVID-19, non-COVID-19)	Accuracy: 89.5%	≈23 M
VGG19 with Standard-CNN [26]	CT Scans and X-Ray (Normal, Pneumonia, COVID-19, Lung Cancer)	Accuracy: 98.05%	22.3 M
COVID-Net [62]	CXRs (Normal, CAP, COVID-19	Accuracy: 93.3%	11.75 M
Location-attention oriented Model (based on ResNet18) [65]	CT Scans (COVID-19, Influenza-A, Viral Pneumonia)	Accuracy: 86.7%	11.7 M
Truncated InceptionNet (3 blocks) [11]	CXRs (COVID-19, CAP, TB (China), TB (USA)	Accuracy: 99.92%	2.1 M
COVIDNet-CT [18]	CT Scans (COVID-19, non-COVID-19)	Accuracy: 99.1%	1.4 M
Lightweight CNN (based on SqueezeNet) [47]	CT Scans (COVID-19, non-COVID-19)	Accuracy: 85.03%	1.26 M
Fused-DenseNet-Tiny [42]	CXRs (Normal, COVID-19, CAP)	Accuracy: 97.99%	1.2 M

Table 13 Comparison of the top-scoring truncated model against other existing studies

with its far smaller parameter size of 441 K. However, it is worth mentioning that there is no direct comparison as InceptionResNetV2-Tr trained with six cases of both CXRs and CT scans in its pipeline, while the others only had two to three that either diagnosed a CXR or CT scan. Therefore, even with a more complex task, the fine-tuned InceptionResNetV2-Tr with the proposed truncation method can become more effective, convenient to train, and use than the other studies presented in most situations.

4 Conclusions

In this work, the results and analysis had shown that with the proposed truncation method of broad DCNNs like the InceptionResNetV2 and InceptionV3, DCNNs could become highly valuable and practical in terms of reproduction and deployment for low-end devices or medical facilities that lack high-end computing resources, giving them significant viability for future implementations. Though complex, massive, and scalable models have provided promising results in the field of medical imaging, this work still emphasized that decreasing the parameters and layer depth of pre-trained and fine-tuned DCNNs through the proposed truncation method can still emanate competitive results, adding that they are more practical and lighter based on a comparative analysis. Despite such a promising result, a significant drawback showed that the proposed method could decrease the overall performance and lessen their interpretability based on the Grad-CAM evaluation. Nonetheless, the easily reproducible and trainable truncated DCNNs can still improve and scale upon the addition of data without consuming copious amounts of computing resources than models with higher parameters and lengthier architectures.

Based on the results and analysis from the conducted investigations, this work concludes that DCNNs with broader architectures and a more robust layer aggregation have less susceptibility to performance reduction when truncated than DCNNs that rely on depth. Furthermore, this work also concludes that DCNN's, when appropriately fine-tuned, even with massively deduced parameters, specifically InceptionResNetV2-Tr and InceptionV3-Tr, can achieve a competitive performance towards the diagnosis of COVID-19 from both CXRs and CT scans simultaneously through a single pipeline.

Based on the given conclusions, future studies can consider adopting the proposed work in segmentation and object detection models, adding more data to produce better and more realistic results, or employing attention mechanisms to improve or maintain their overall interpretability without re-bloating their parameter size.

Acknowledgments The author wishes to acknowledge Batangas State University for the support given to the accomplishment of this work.

Code and data availability The author provided the codes and links of the datasets used in this work due to the importance of research reproducibility.

→ https://github.com/francismontalbo/covid19_diagnosis_and_analysis_of_truncated_dcnns/

Dataset and code availability This work provides both the dataset source and code required to simulate the proposed model for ease of evaluation. https://github.com/francismontalbo/covid19_diagnosis_and_analysis_of_truncated_dcnns

Declarations

Ethical approval This work did not involve humans, animals, and other living specimens. The author kept the research etiquette regarding such matters throughout the conduct of this work.

Conflict of interest The author has no conflict of interest to declare or disclose regarding this work.

References

- Afshar P et al (2021) COVID-CT-MD, COVID-19 computed tomography scan dataset applicable in machine learning and deep learning. Sci Data 8(1). https://doi.org/10.1038/s41597-021-00900-3
- Alakus T, Turkoglu I (2020) Comparison of deep learning approaches to predict COVID-19 infection. Chaos Solitons Fractals 140:110120. https://doi.org/10.1016/j.chaos.2020.110120
- Aljondi R, Alghamdi S (2020) Diagnostic value of imaging modalities for COVID-19: scoping review. J Med Internet Rese 22(8):e19673. https://doi.org/10.2196/19673
- Alsharif W, Qurashi A (2021) Effectiveness of COVID-19 diagnosis and management tools: a review. Radiography 27(2):682–687. https://doi.org/10.1016/j.radi.2020.09.010
- Bai H et al (2020) Artificial intelligence augmentation of radiologist performance in distinguishing COVID-19 from pneumonia of other origin at chest CT. Radiology 296(3):E156–E165. https://doi.org/10.1148/ radiol.2020201491
- Chattopadhay A, Sarkar A, Howlader P, Balasubramanian VN (2018) Grad-CAM++: generalized gradientbased visual explanations for deep convolutional networks. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp 839–847. https://doi.org/10.1109/WACV.2018.00097
- Chen X, Lin X (2014) Big data deep learning: challenges and perspectives. IEEE Access 2:514–525. https:// doi.org/10.1109/ACCESS.2014.2325029
- Chollet F (2017) Xception: deep learning with depthwise separable convolutions. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1800–1807. https://doi.org/10.1109/CVPR.2017.195
- Coronavirus (COVID-19) testing: What you should know | UC Davis Health, Health.ucdavis.edu, 2021. [Online]. Available: https://health.ucdavis.edu/coronavirus/coronavirus-testing.html. Accessed 27 Apr 2021
- 10. COVID-19. 2020. [Online]. Available: http://medicalsegmentation.com/covid19/. Accessed 02 Mar 2021]
- Das D, Santosh K, Pal U (2020) Truncated inception net: COVID-19 outbreak screening using chest X-rays. Phys Eng Sci Med 43(3):915–925. https://doi.org/10.1007/s13246-020-00888-x
- Do S, Song K, Chung J (2020) Basics of deep learning: a radiologist's guide to understanding published radiology articles on deep learning. Kor J Radiol 21(1):33. Available: https://doi.org/10.3348/kjr.2019.0312
- Druzhkov P, Kustikova V (2016) A survey of deep learning methods and software tools for image classification and object detection. Pattern Recognit Image Anal 26(1):9–15. https://doi.org/10.1134/ s1054661816010065
- Feurer M, Hutter F (2019) Hyperparameter optimization. Automated Machine Learning, pp 3–33. DOI: https://doi.org/10.1007/978-3-030-05318-5_1
- Gabruseva T, Poplavskiy D, Kalinin A (2020) Deep learning for automatic pneumonia detection. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp 1436– 1443. https://doi.org/10.1109/CVPRW50498.2020.00183
- Gibson E et al (2018) NiftyNet: a deep-learning platform for medical imaging. Comput Methods Prog Biomed 158:113–122. https://doi.org/10.1016/j.cmpb.2018.01.025
- 17. Giri A, Rana D (2020) Charting the challenges behind the testing of COVID-19 in developing countries: Nepal as a case study. Biosafety and Health 2(2):53–56. https://doi.org/10.1016/j.bsheal.2020.05.002
- Gunraj H, Wang L, Wong A (2020) COVIDNet-CT: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest CT images. Front Med 7. https://doi.org/10.3389/fmed.2020. 608525
- He K, Zhang X, Ren S, Sun J (2016) Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 770–778. https://doi.org/10.1109/ CVPR.2016.90
- He K, Zhang X, Ren S, Sun J (2016) Identity mappings in deep residual networks. In: ECCV. https://doi. org/10.1007/978-3-319-46493-0_38.
- Hernández Santa Cruz J (2021) An ensemble approach for multi-stage transfer learning models for COVID-19 detection from chest CT scans. Intell Based Med 5:100027. https://doi.org/10.1016/j.ibmed.2021.100027

- Hinz T, Barros P, Wermter S (2016) The effects of regularization on learning facial expressions with convolutional neural networks. Artificial Neural Networks and Machine Learning – ICANN 2016, pp 80– 87 https://doi.org/10.1007/978-3-319-44781-0_10
- Hossin M, Sulaiman MN (2015) A review on evaluation metrics for data classification evaluations. Int J Data Mining Knowl Manage Process 5(2):1–11. https://doi.org/10.5121/ijdkp.2015.5201
- Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2261–2269. https://doi. org/10.1109/CVPR.2017.243
- Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K (2016) Squeezenet: Alexnet-level accuracy with 50x fewer parameters and_i 0.5 mb model size. arXiv preprint arXiv:1602.07360. Available: https://arxiv.org/abs/1602.07360
- Ibrahim D, Elshennawy N, Sarhan A (2021) Deep-chest: multi-classification deep learning model for diagnosing COVID-19, pneumonia, and lung cancer chest diseases. Comput Biol Med 132:104348. https://doi.org/10.1016/j.compbiomed.2021.104348
- Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd International Conference on Machine Learning, in Proceedings of Machine Learning Research, pp 448–456, Available: http://proceedings.mlr.press/v37/ioffe15.html
- Istaiteh O, Owais T, Al-Madi N, Abu-Soud S (2020) Machine learning approaches for COVID-19 forecasting. 2020 International Conference on Intelligent Data Science Technologies and Applications (IDSTA), pp 50–57 https://doi.org/10.1109/IDSTA50958.2020.9264101
- Jeni LA, Cohn JF, De La Torre F (2013) Facing imbalanced data–recommendations for the use of performance metrics. 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, Geneva, pp 245-251. https://doi.org/10.1109/ACII.2013.47
- Jun M et al (2020) COVID-19 CT lung and infection segmentation dataset. Zenodo, Apr, vol 20. Available: https://zenodo.org/record/3757476#.YJfUtrUzZBc
- Kandel I, Castelli M (2020) How deeply to fine-tune a convolutional neural network: a case study using a histopathology dataset. Appl Sci 10(10):3359. https://doi.org/10.3390/app10103359
- Ketkar N (2017) Stochastic gradient descent. In: Deep learning with python. Apress, Berkeley. https://doi. org/10.1007/978-1-4842-2766-4_8
- Kingma DP, Ba JL (2015) Adam: a method for stochastic optimization. In: Proc. Int. Conf. Learn. Represent, pp 1–41. Available: https://hdl.handle.net/11245/1.505367
- Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp 1097–1105. Available: https:// papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html
- Lai Z, Deng H (2018) Medical image classification based on deep features extracted by deep model and statistic feature fusion with multilayer perceptron. Comput Intell Neurosci 2018:1–13. https://doi.org/10. 1155/2018/2061516
- LeCun Y, Haffner P, Bottou L, Bengio Y (1999) Object recognition with gradient-based learning. Shape, Contour and Grouping in Computer Vision, pp 319–345. https://doi.org/10.1007/3-540-46805-6_19
- Lee L, Liew S, Thong W (2014) A review of image segmentation methodologies in medical image. Lecture Notes in Electrical Engineering, pp. 1069–1080. https://doi.org/10.1007/978-3-319-07674-4_99
- Li L et al (2020) Using artificial intelligence to detect COVID-19 and community-acquired pneumonia based on pulmonary CT: evaluation of the diagnostic accuracy. Radiology 296(2):E65–E71. https://doi.org/ 10.1148/radiol.2020200905
- Lin M, Chen Q, Yan S (2013) Network in network. arXiv preprint arXiv:1312.4400. Available: https:// arxiv.org/abs/1312.4400
- ML Cheatsheet, c2017. [Online]. Available: https://mlcheatsheet.readthedocs.io/en/latest/loss_functions. html. Accessed 28 Mar 2021
- Mohammad-Rahimi H, Nadimi M, Ghalyanchi-Langeroudi A, Taheri M, Ghafouri-Fard S (2021) Application of machine learning in diagnosis of COVID-19 through X-Ray and CT images: a scoping review. Front Cardiovasc Med 8. https://doi.org/10.3389/fcvm.2021.638011
- Montalbo F (2021) Diagnosing Covid-19 chest x-rays with a lightweight truncated DenseNet with partial layer freezing and feature fusion. Biomed Signal Process Control 68:102583. https://doi.org/10.1016/j.bspc. 2021.102583
- Montalbo F (2021) Truncating a densely connected convolutional neural network with partial layer freezing and feature fusion for diagnosing COVID-19 from chest X-rays. MethodsX 8:101408. Available: https:// doi.org/10.1016/j.mex.2021.101408
- Morozov S et al (2020) Mosmeddata: Chest CT scans with covid-19 related findings. arXiv:2005.06465. [Online]. Available: https://arxiv.org/abs/2005.06465

- Narayan Das N, Kumar N, Kaur M, Kumar V, Singh D (2020) Automated deep transfer learning-based approach for detection of COVID-19 infection in chest X-rays. IRBM. https://doi.org/10.1016/j.irbm.2020.07.001
- Paul Cohen J, Morrison P, Dao L (2020) COVID-19 image data collection. arXiv:2003.11597. [Online]. Available: http://arxiv.org/abs/2003.11597
- Polsinelli M, Cinque L, Placidi G (2020) A light CNN for detecting COVID-19 from CT scans of the chest. Pattern Recogn Lett 140:95–100. https://doi.org/10.1016/j.patrec.2020.10.001
- Qian Y et al (2020) Safety management of nasopharyngeal specimen collection from suspected cases of coronavirus disease 2019. Int J Nurs Sci 7(2):153–156. https://doi.org/10.1016/j.ijnss.2020.03.012
- Rahimzadeh M, Attar A, Sakhaei SM (2020) A fully automated deep learning-based network for detecting covid-19 from a new and large lung ct scan dataset. medRxiv. [Online]. Available: https://www.medrxiv. org/content/early/2020/06/12/2020.06.08.20121541
- Ramachandran P, Zoph B, Le QV (2017) Searching for activation functions. arXiv preprint arXiv: 1710.05941. Available: https://arxiv.org/abs/1710.05941
- Sait U, Gokul Lal KV, Prajapati SP, Bhaumik R, Kumar T, Sanjana S, Bhalla K Curated dataset for COVID-19 posterior-anterior chest radiography images (X-Rays). Mendeley Data, V1. https://doi.org/10. 17632/9xkhgts2s6.1
- Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-CAM: visual explanations from deep networks via gradient-based localization. 2017 IEEE International Conference on Computer Vision (ICCV), pp 618–626. https://doi.org/10.1109/ICCV.2017.74
- Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: Proc. Int. Conf. Learn. Represent., pp 1–14. https://arxiv.org/abs/1409.1556
- Srivastava N et al (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958. https://doi.org/10.5555/2627435.2670313
- Subbarao K, Mahanty S (2020) Respiratory virus infections: understanding COVID-19. Immunity 52(6): 905–909. https://doi.org/10.1016/j.immuni.2020.05.004
- Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the Inception Architecture for Computer Vision. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2818– 2826. https://doi.org/10.1109/CVPR.2016.308
- Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11231
- Tan M, Le Q (2019) EfficientNet: Rethinking model scaling for convolutional neural networks. In: Proc. Int. Conf. Mach. Learn., pp 6105–6114. Available: https://proceedings.mlr.press/v97/tan19a.html
- Thomford N et al (2020) Implementing artificial intelligence and digital health in resource-limited settings? Top 10 lessons we learned in congenital heart defects and cardiology. OMICS: A Journal of Integrative Biology 24(5):264–277. https://doi.org/10.1089/omi.2019.0142
- 60. Ting KM (2017) Confusion matrix. Springer, Boston, p 260. https://doi.org/10.1007/978-1-4899-7687-1_50
- Udugama B et al (2020) Diagnosing COVID-19: the disease and tools for detection. ACS Nano 14(4): 3822–3835. https://doi.org/10.1021/acsnano.0c02624
- Wang L, Lin Z, Wong A (2020) COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. Sci Rep 10(1). https://doi.org/10.1038/s41598-020-76550-z
- Wang S et al (2021) A deep learning algorithm using CT images to screen for Corona virus disease (COVID-19). Eur Radiol. https://doi.org/10.1007/s00330-021-07715-1
- Xu B, Wang N, Chen T, Li M (2015) Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853. Available: https://arxiv.org/abs/1505.00853
- Xu X et al (2020) A deep learning system to screen novel coronavirus disease 2019 pneumonia. Engineering 6(10):1122–1129. https://doi.org/10.1016/j.eng.2020.04.010
- Yang X, He X, Zhao J, Zhang Y, Zhang S, Xie P (2020) COVID-CT dataset: a CT scan dataset about COVID-19. arXiv:2003.13865. [Online]. Available: http://arxiv.org/abs/2003.13865
- Yao Q, Wang R, Fan X, Liu J, Li Y (2020) Multi-class arrhythmia detection from 12-lead varied-length ECG using attention-based time-incremental convolutional neural network. Inf Fusion 53:174–182. https:// doi.org/10.1016/j.inffus.2019.06.024
- Ying X (2019) An overview of overfitting and its solutions. J Phys Conf Ser 1168:Art. no. 022022 Available: https://iopscience.iop.org/article/10.1088/1742-6596/1168/2/022022/meta
- Yu T, Zhu H (2020) Hyper-parameter optimization: a review of algorithms and applications, arXiv preprint arXiv:2003.05689. Available: https://arxiv.org/abs/2003.05689

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.