diagnostics

MDPI

Article

Object or Background: An Interpretable Deep Learning Model for COVID-19 Detection from CT-Scan Images

Gurmail Singh and **Kin-Choong Yow** *

Faculty of Engineering and Applied Science, University of Regina, Regina, SK S4S 0A2, Canada; Gurmail.Singh@uregina.ca

* Correspondence: Kin-Choong.Yow@uregina.ca

Abstract: The new strains of the pandemic COVID-19 are still looming. It is important to develop multiple approaches for timely and accurate detection of COVID-19 and its variants. Deep learning techniques are well proved for their efficiency in providing solutions to many social and economic problems. However, the transparency of the reasoning process of a deep learning model related to a high stake decision is a necessity. In this work, we propose an interpretable deep learning model Ps-ProtoPNet to detect COVID-19 from the medical images. Ps-ProtoPNet classifies the images by recognizing the objects rather than their background in the images. We demonstrate our model on the dataset of the chest CT-scan images. The highest accuracy that our model achieves is 99.29%.

Keywords: COVID-19; pneumonia; CT-scan; prototypical part

check for updates

Citation: Singh, G.; Yow, K.-C. Object or Background: An Interpretable Deep Learning Model for COVID-19 Detection from CT-Scan Images. *Diagnostics* **2021**, *11*, 1732. https://doi.org/10.3390/diagnostics11091732

Academic Editor: Jae-Ho Han

Received: 9 August 2021
Accepted: 14 September 2021
Published: 21 September 2021

1. Introduction

The pandemic COVID-19 is looming as a worst menace on the world populations while its several new strains are being identified. Some vaccines for COVID-19 have been developed, but the list of the variants of COVID-19 is also getting bigger. There are seven lineages of the variants of the virus, such as: B.1.1.7, B.1.351, P.1, B.1.427, B.1.429, B.1.525, B.1.617.1 and B.1.617.2 [1]. The detection of the virus is usually done with molecular tests, that is, the tests that look for the virus by detecting the presence of the virus's RNA. The molecular tests include RT-PCR, CRISPR, isothermal nucleic acid amplification, digital polymerase chain reaction, microarray analysis, and next-generation sequencing [2]. The presence of the virus can also be detected from the medical images, such as: chest X-ray and CT images. Although, RT-PCR is still a gold standard for COVID-19 testing, but deep learning techniques to identify the virus from medical images can also be helpful in certain circumstances, such as: unavailability of RT-PCR kits. A deep learning model can also be used for the pre-screening before RT-PCR testing. Many models have been proposed to detect COVID-19 from the medical images, see [3–15]. However, these models lack the interpretability/transparency of the reasoning process of their predictions. So, we propose an interpretable deep learning model: *pseudo prototypical part network* (Ps-ProtoPNet), and experiment it over the dataset of CT-scan images, see Section 2.4. Ps-ProtoPNet is closely related to ProtoPNet [16], Gen-ProtoPNet [17] and NP-Proto-PNet [18], but strikingly different from these models.

A prototype represents a patch of an image. To classify a test image, ProtoPNet compares the different parts of the test image with the learned prototypes of images from all classes. Then the decision is made based on the weighted combination of similarity scores [16]. To calculate the similarity scores between learned prototypes (with square spatial dimensions 1 × 1) and parts of the test image, ProtoPNet and NP-ProtoPNet use $L2$ distance function, whereas Gen-ProtoPNet uses a generalized version of $L2$.

In this work, we present a theorem that calculates the impact of the change in the hyperparameters of the dense layer on the logits, see Theorem 1. Ps-ProtoPNet chooses negative connections between the similarity score and logits of incorrect classes as suggested by the theorem. Also, our model uses prototypes that can have any type of spatial dimensions, that is, square and rectangular.

A model should classify an image of an *object* by identifying the object in the image instead of the *background* of the object in the image. The model that uses prototypes of smaller spatial dimensions (1 × 1) can classify an image just on the basis of the background and give higher accuracy with wrong reasoning process. For example, the most part of the images of birds of a sea specie is not similar to any patch of the images of birds of a jungle specie. So, the images from these two classes can be classified on the basis of backrounds. Another scenario, images of birds of different sea bird species can share same background water on the most part. Therefore, a model with prototypes of small spatial dimensions (1 × 1) can classify wrongly the images just on the basis of the background of the birds. On the other hand, the use of prototypes with the dimensions equal to the dimensions of an image can also reduce the accuracy because there can be only few images that are similar to the whole image, but their parts can be similar. So, we need to use optimum spatial dimensions for the prototypes. To identify an image that has not been encountered before, humans may compare patches of the image with the patches of images of the known objects. Our model's reasoning is inspired from the above reasoning, where comparison of image parts with learned prototypes is integral to the reasoning process of the model. That is, a new image is compared with learned prototypes from all classes, and it is classified to the class whose prototypes are more similar to parts of the image. We have three classes of images: Covid, Normal and Pneumonia. Therefore, a COVID-19 CT image is distinguished from the pneumonia CT images based on the greater similarity of parts of the image with the prototypes.

2. Materials and Methods

2.1. Related Work

Numerous perspectives have been emerged to explain convolution neural networks, including posthoc interpretability analysis. A neural network with *posthoc* analysis is interpreted following classifications made by a model. Activation maximization [19–25], deconvolution [26], and saliency visualization [23,27–29] are some forms of posthoc analysis approach. Nevertheless, these techniques do not throw light on the reasoning process with transparency. Another approach to make the reasoning process of the neural networks clear is attention-based interpretability that includes class activation maps (CAM) and part-based models. In this approach, a model aims to point out the parts of a test image that are its centers of attention [30–41]. These models do not point out the prototypes that are similar to parts of the test image.

Oscar et al. [42] developed a model that uses prototypes of the size of a whole image to find the similarity scores. A substantial improvement over the above work was made by Chen et al. with the development of their model ProtoPNet [16]. The models Gen-ProtoPNet [17] and NP-ProtoPNet [18] are close variations of ProtoPNet.

2.2. Data

Many datasets of medical images are publicly available [43–45]. However, we used the dataset of chest CT-scan images of normal people, COVID-19 patients and pneumonia patients [44]. This dataset has 143, 778 training images and 25,658 test images. The training dataset consists of 35,996, 25,496 and 82,286 CT-scan images of normal people, pneumonia patients and COVID-19 patients, respectively. The test dataset consists of 12, 245, 7395 and 6018 CT-scan images of normal people, pneumonia patients and COVID-19 patients. We resized the images to the dimensions 224 × 224 as required by the base models. We put these images into three classes Covid (first class), Normal (second class) and Pneumonia (third class).

2.3. Working Principal and Novelty of Ps-ProtoPNet

ProtoPNet classify an image on the basis of a weighted combination of the similarity scores [16]. For each class, a fixed number of prototypes are selected. We select 10 prototypes for each class. The model calculates the Euclidean distance of each prototype from each latent patch of the test image that has spatial dimensions equal to 1×1. Then these distances are inverted and a maximum of the inverted distances is called the similarity score of the prototype. Thus, for a given image, only one similarity score for each prototype is obtained. In the dense layer, these similarity scores are multiplied with the weights to calculates the logits. During the training process, ProtoPNet does the convex optimization of the last layer to make the certain weights zero [16].

Theorem 1 finds the impact of the change in the weights on the logits. Therefore, along with the use of prototypes with spatial dimensions bigger than 1×1, Ps-ProtoPNet uses the negative weights for similarity scores that connect to incorrect classes. Thus, for a given CT-scan image as in Figure 1, Ps-ProtoPNet identifies the parts of the image where it thinks that this part of the image looks like that prototypical part, and this part of the image does not look like that prototypical part. In addition to the positive reasoning process, Ps-ProtoPNet does not do the convex optimiza-tion of the last layer to keep the impact of the negative reasoning process on the image classification, whereas ProtoPNet model emphasizes on the positive reasoning process. The non-optimization of the last layer enabled us to write Theorem 1, because it ensures that the weights of last layer do not change during the training process. Also, it reduces the training time considerably.

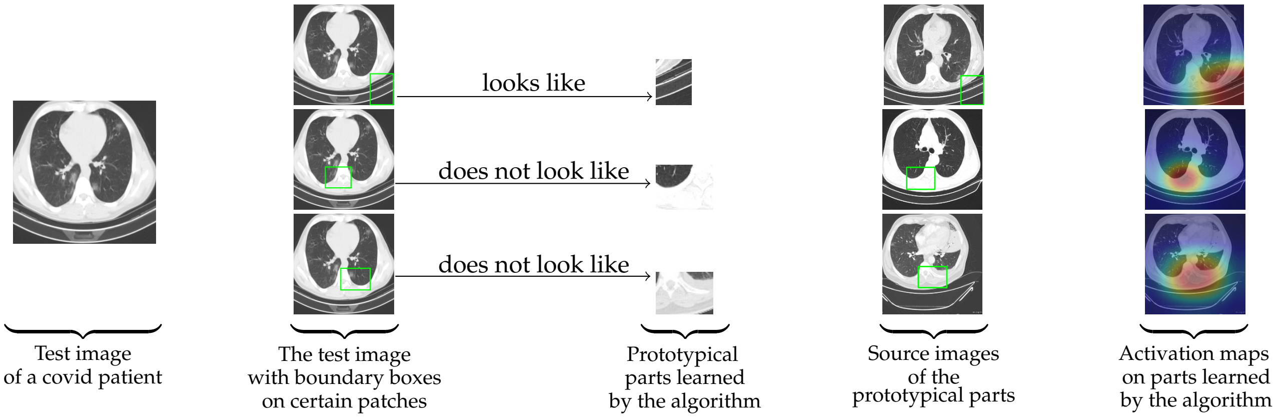
looks like
does not look like
does not look like
Test image of a covid patient
The test image with boundary boxes on certain patches
Prototypical parts learned by the algorithm
Source images of the prototypical parts
Activation maps on parts learned by the algorithm

Figure 1. For a given CT-scan image, Ps-ProtoPNet identifies the parts of the image where it thinks that this part of the image looks like that prototypical part, and this part of the image does not look like that prototypical part.

2.4. Ps-ProtoPNet Architecture

In this section, we introduce and explain the architecture and the training procedure of our model Ps-ProtoPNet in the context of CT-scan images.

We construct our network over the state-of-the-art mod-els: VGG-16, VGG-19 [46], ResNet-34, ResNet-152 [47], DenseNet-121, or DenseNet-161 [48]. In this paper, these models are called baseline or base models. The base models were pretrained on ImageNet [49]. In the Figure 2, we see that the model comprises of the convolution layers of any of the above base model that are followed by an additional 1×1 layer (we denote these convolution layers together by ℓ) and then these convolution layers are followed by a generalized [50,51] convolution layer p_p of prototypical parts and a dense layer w with weight matrix m_w. The dense layer does not have any bias. We denote the parameters of ℓ by ℓ_{conv}. The activation function Sigmoid is used for the additional convolution layer.

We provide an explanation of our model with the base model VGG-16. For an input image x, let $\ell(x)$ be the output of the convolutional layers ℓ. Therefore, the shape of $\ell(x)$ is $512 \times 7 \times 7$. Let $P^k = \{p_l^k\}_{l=1}^{m'}$ be the set of prototypes of a class k and $P = \{P^k\}_{k=1}^{n}$ is set of prototypes of all classes, where m' is the number of prototypes for each class and n

is the total number of classes. In our case, $m' = 10$ and $n = 3$, and the hyperparameter $m' = 10$ is chosen randomly. For example, $p_1^1, p_2^1, \ldots, p_{10}^1$ prototypes belong to the first class (Covid class). The shape of each prototype is $512 \times h \times w$, where $1 \times 1 < h \times w < 7 \times 7$, that is, h and w are neither simultaneously equal to 1 nor 7. Hence, every prototype can be considered as a representation of some prototypical part of some CT-scan image.

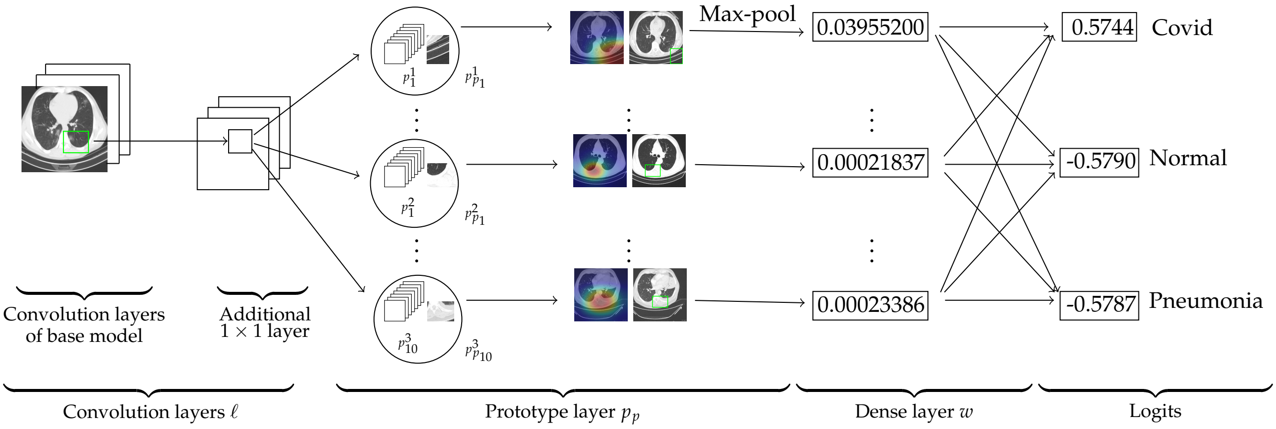

Figure 2. Ps-ProtoPNet architecture.

As explained in Section 2.3, Ps-ProtoPNet calculates the similarity scores between an input image and the prototypical parts $p_1^1 - p_{10}^1$, $p_1^2 - p_{10}^2$ and $p_1^3 - p_{10}^3$, see Figure 2. Note that, similarity score of the prototype p_1^1 (0.03955200) is greater than the similarity scores of p_1^2 (0.00021837) and p_{10}^3 (0.00023386). The complete list is given in the similarity score matrix S, see Section 2.6. The source image of the prototypes p_1^1, p_1^2 and p_{10}^3 are also given in the third column of the Figure 2. The model keeps track of spatial relation of the convolutional output and the prototypical parts, and upsamples the parts to the size of input image to point out the patch on the source images that corresponds to the prototypes. The rectangles in the source images are the parts of the source images from where the prototypical parts are taken. In layer w, the matrix S is multiplied with m_w to get the logits. The logits for the first, second and third class are 0.5744, -0.5790 and -0.5787, respectively.

2.5. The Training of Ps-ProtoPNet

We use the generalized version d of the distance function $L2$ (Euclidean distance). We consider the baseline VGG-16 to present d in this section. For a given image x, let $z = \ell(x)$. Therefore, the shape of $\ell(x)$ is $512 \times 7 \times 7$, where 512 is the depth of $\ell(x)$ and 7×7 are the spatial dimensions of $\ell(x)$. Let p be a prototype of the shape $512 \times h \times w$, where $1 \leq h, w \leq 7$, but h and w are neither simultaneously equal to 1 nor 7. Since p can be any prototype of any class, p does not have any subscript and superscript. The output z of the convolutional layers ℓ has $(8-h)(8-w)$ patches of dimensions $h \times w$. Hence, square of the distance $d(\mathcal{Z}_{ij}, p)$ between the prototype p and (i, j) patch $\mathcal{Z}_{ij}$ (say) of z is:

$$d^2(\mathcal{Z}_{ij}, p) = \sum_{l=1}^{h} \sum_{m=1}^{w} \sum_{k=1}^{512} ||z_{(i+l-1)(j+m-1)k} - p_{lmk}||_2^2. \quad (1)$$

For prototypes of spatial dimension 1×1, that is, $h = w = 1$, we have $d^2(\mathcal{Z}_{ij}, p) = \sum_{k=1}^{512} ||z_{ijk} - p_{11k}||_2^2$, which is the square of the Euclidean distance between the prototype p and a patch of z, where $p_{11k} \simeq p_k$. Therefore, the distance function d is a generalization of $L2$. The prototypical unit p_p calculates the following.

$$p_p(z) = \max_{1 \leq i \leq 8-h,\ 1 \leq j \leq 8-w} \log\left(\frac{d^2(\mathcal{Z}_{ij}, p) + 1}{d^2(\mathcal{Z}_{ij}, p) + \epsilon}\right).$$

In other words,

$$p_p(z) = \max_{\mathcal{Z} \in \text{patches}(z)} \log\left(\frac{d^2(\mathcal{Z}, p) + 1}{d^2(\mathcal{Z}, p) + \epsilon}\right). \tag{2}$$

The Equation (2) tells us that a prototype p is more similar to input image x if the inverse of the distance between a latent patch of x and p is smaller. The two training steps of our model are as follows.

2.5.1. Optimization of All Layers before the Dense Layer

Suppose $X = \{x_1 \dots x_n\}$ and $Y = \{y_1 \dots y_n\}$ are sets of images and corresponding labels, respectively. Let $D = \{(x_i, y_i) : x_i \in X, y_i \in Y\}$. Our objective function is:

$$\min_{P, \ell_{conv}} \frac{1}{n} \sum_{i=1}^{n} \text{CrosEnt}(h \circ p_p \circ \ell(x_i), y_i) + \lambda_1 \text{ClstCst} + \lambda_2 \text{SepCst}, \tag{3}$$

where ClstCst and SepCst are:

$$\text{ClstCst} = \frac{1}{n} \sum_{i=1}^{n} \min_{j: p_j \in P_{y_i}} \min_{\mathcal{Z} \in \text{patches}(\ell(x_i))} d^2(\mathcal{Z}, p_j); \tag{4}$$

$$\text{SepCst} = -\frac{1}{n} \sum_{i=1}^{n} \min_{j: p_j \notin P_{y_i}} \min_{\mathcal{Z} \in \text{patches}(\ell(x_i))} d^2(\mathcal{Z}, p_j). \tag{5}$$

The Equation (4) tells us that the decrease in the cluster cost (ClstCst) leads to clustering of prototypes surrounding their respective classes. However, the Equation (5) suggests that the decrease in separation cost (SepCst) keeps prototypes away from their incorrect classes [16]. The drop in cross entropy leads to improved classifications, see the objective function (3). The hyperparameters λ_1 and λ_2 are selected from the set $\{0.4, 0.5, 0.7, 0.8, 0.9\}$ using cross validation. Since m_w is the weight matrix for the last layer, $m_w^{(i,j)}$ is the weight assigned to the connection between similarity score of jth prototype and logit of ith class. Theorem 1 finds the impact of the selection of the weights $m_w^{(i,j)}$ on the logits. Therefore, for a class k, we put $m_w^{(i,j)} = 1$ for all j with $p_j^i \in P^i$, and for all $p_j^k \notin P^i$ with $k \neq i$, $m_w^{(k,j)}$ is chosen from the set $\{-1, -0.9, -0.7, -0.5, -0.2, -0.1\}$. Since the distance function is nonnegative, the optimization of all layers except the last layer with the optimizer SGD helps Ps-ProtoPNet to learn important latent space.

2.5.2. Push of Prototypical Parts

At this step, Ps-ProtoPNet pushes/projects the prototypes onto the patches of the output $\ell(x)$ of an image x that have smallest distances from the prototypes. That is, Ps-ProtoPNet performs the following update:

$$p_j^k \longleftarrow \arg \min_{\{\mathcal{Z}: \mathcal{Z} \in \text{patches}(\ell(x_i))\ \forall i \text{ s.t. } y_i = k\}} d(\mathcal{Z}, p_j^k).$$

Therefore, prototype layer gets updated prototypical parts that are more closer to their respective classes [16]. The patch of x that is the most similar to p is used for visualization of p. The activation value of the prototype must be at least 94th percentile of all the activation values of p_p [16].

2.6. *Explanation of Ps-ProtoPNet with an Example*

The test image in the first column of Figure 3 belongs to the first class (Covid). In the second column, the test image has some patches enclosed in green rectangles. These patches give the highest similarity score to the corresponding prototypes in the third column. The prototypes in the third column are taken from the corresponding source images in the fourth column. The rectangles on the source image pin-point the patches from where the corresponding prototypes are taken. The fifth column has similarity scores of the prototypes and sixth column has the weights. The entries of the seventh column are obtained by multiplying the similarity scores with the corresponding weights. The logit (0.5744) of the first class is the sum of entries of the seventh column. The logit for the first class can also be obtained from the multiplication of the first row of weight matrix m_w with the similarity score matrix S. Similarly, the logit for the second class (-0.5790) and third class (-0.5787) can be obtained by multiplying second and third row of the weight matrix with the similarity score matrix S.

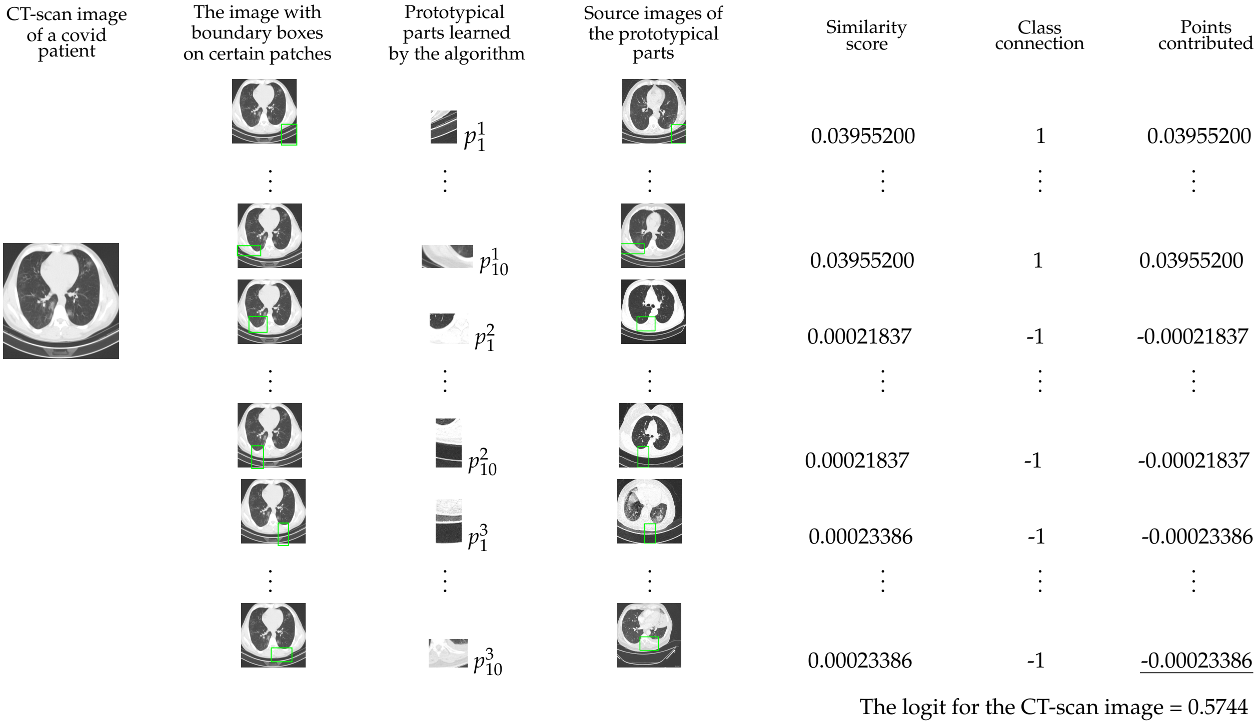
CT-scan image of a covid patient
The image with boundary boxes on certain patches
Prototypical parts learned by the algorithm
Source images of the prototypical parts
Similarity score
Class connection
Points contributed
p_1^1 0.03955200 1 0.03955200
p_{10}^1 0.03955200 1 0.03955200
p_1^2 0.00021837 -1 -0.00021837
p_{10}^2 0.00021837 -1 -0.00021837
p_1^3 0.00023386 -1 -0.00023386
p_{10}^3 0.00023386 -1 -0.00023386
The logit for the CT-scan image = 0.5744

Figure 3. Explanation of the classification process of the model.

The transpose of the weight matrix m_w and similarity scores matrix S that we obtain from our experiments are as follows:

$$m_w^T = \begin{bmatrix} 1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & 1 & -1 \\ -1 & 1 & -1 \\ -1 & 1 & -1 \\ -1 & 1 & -1 \\ -1 & 1 & -1 \\ -1 & 1 & -1 \\ -1 & 1 & -1 \\ -1 & 1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ -1 & -1 & 1 \\ -1 & -1 & 1 \\ -1 & -1 & 1 \\ -1 & -1 & 1 \\ -1 & -1 & 1 \\ -1 & -1 & 1 \\ -1 & -1 & 1 \\ -1 & -1 & 1 \\ -1 & -1 & 1 \end{bmatrix} \quad \text{and} \quad S = \begin{bmatrix} 0.03955200 \\ 0.03955200 \\ 0.03955200 \\ 0.03955200 \\ 0.03955200 \\ 0.03955200 \\ 0.03955200 \\ 0.03955200 \\ 0.22292000 \\ 0.03955200 \\ 0.00021837 \\ 0.00021837 \\ 0.00021837 \\ 0.00021837 \\ 0.00021837 \\ 0.00021837 \\ 0.00021837 \\ 0.00021837 \\ 0.00021837 \\ 0.00021813 \\ 0.00023386 \\ 0.00023386 \\ 0.00023374 \\ 0.00023386 \\ 0.00023386 \\ 0.00023386 \\ 0.00023386 \\ 0.00023386 \\ 0.00023374 \\ 0.00023386 \end{bmatrix}.$$

3. Results

In this section we present the metrics given by our model and compare the performance of our model with the performance of the other models.

3.1. The Metrics and Confusion Matrices

For a given class, true positive (TP) and true negative (TN) are the number of items correctly predicted as belonging to the class and not belonging to the class, respectively, see [52]. False positives (FP) and false negatives (FN) are the number of items incorrectly predicted as belonging to the class and not belonging to the class, respectively, see [53]. The metrics accuracy, precision, recall and F1-score are [54–56]:

$$\text{Accuracy} = \frac{TP + TN}{\text{Total Cases}}, \ \text{Precision} = \frac{TP}{TP + FP}. \tag{6}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \ \text{F1-score} = \frac{2}{\text{Precision}^{-1} + \text{Recall}^{-1}}. \tag{7}$$

In Figures 4–9, the confusion matrices of Ps-ProtoPNet with the base models are given. For example, in Figure 4, the confusion matrix N (say) of Ps-ProtoPNet with base model VGG-16 is provided. Thus, the numbers $N[0][0]$, $N[1][1] + N[2][2]$, $N[0][1] + N[0][2]$ and $N[1][0] + N[2][0]$ denote the true positives TP, true negatives TN, false positives FP and false negatives FN of the Covid class. Therefore, by Equations (6) and (7), the accuracy for Ps-ProtoPNet is 98.83%, and the precision, recall and F1-score are equal to 0.96, 0.98 and 0.97, respectively.

Actual class

		Covid	Normal	Pneumonia
Predicted Class	Covid	5997	178	11
	Normal	12	11980	2
	Pneumonia	9	87	7382

Figure 4. Ps-ProtoPNet with base VGG-16.

Actual class

		Covid	Normal	Pneumonia
Predicted Class	Covid	5987	165	5
	Normal	6	11904	1
	Pneumonia	25	176	7389

Figure 5. Ps-ProtoPNet with base VGG-19.

Actual class

		Covid	Normal	Pneumonia
Predicted Class	Covid	5964	153	29
	Normal	25	12072	8
	Pneumonia	29	20	7358

Figure 6. Ps-ProtoPNet with ResNet-34.

		Actual class		
		Covid	Normal	Pneumonia
Predicted Class	Covid	5924	127	9
	Normal	53	12071	12
	Pneumonia	41	47	7374

Figure 7. Ps-ProtoPNet with ResNet-152.

		Actual class		
		Covid	Normal	Pneumonia
Predicted Class	Covid	5980	114	4
	Normal	16	12094	3
	Pneumonia	22	37	7388

Figure 8. Ps-ProtoPNet with DenseNet-121.

		Actual class		
		Covid	Normal	Pneumonia
Predicted Class	Covid	5989	174	18
	Normal	20	12043	2
	Pneumonia	9	28	7375

Figure 9. Ps-ProtoPNet with DenseNet-161.

3.2. The Performance Comparison of the Models

The models Ps-ProtoPNet, Gen-ProtoPNet, NP-ProtoPNet and ProtoPNet are constructed over the convolution layers of base models. We trained and tested these models over the dataset of CT-scan images [44]. Although, the accuracies of these models stabilize before 30 epochs (see Section 3.3), but we trained and tested the models for 100 epochs.

The comparison of the performance in the metrics is given in the Table 1. We observe from the third column of Table 1 that when we construct our model over the convolutional layers of VGG-16, and use the prototypes of spatial dimensions 3×4 then the accuracy, precision, recall and F1-score given by Ps-ProtoPNet are 98.83, 0.96, 0.98 and 0.97, respectively. The accuracy, precision, recall and F1-score given by the models Gen-ProtoPNet, NP-ProtoPNet and ProtoPNet with baseline VGG-16 are 95.85, 0.93, 0.95 and 0.94; 98.23, 0.93, 0.95 and 0.94; and 90.84, 0.89, 0.91 and 0.90, respectively. The accuracy, precision, recall and F1-score given by VGG-16 itself (Base only) are 99.03, 0.98, 0.99 and 0.98, respectively. Also, we observe from the Table 1 that the performance of Ps-ProtoPNet is the highest after base models.

Table 1. The performances comparison of the models while experimented over the dataset of the CT-scan images.

Base (B)	**Metric**	**Ps-ProtoPNet**	**Gen-ProtoPNet [14]**	**NP-ProtoPNet [17]**	**ProtoPNet [5]**	**B Only**
VGG-16		3 × 4				
	accuracy	98.83	95.85	98.23	90.84	99.03
	precision	0.96	0.93	0.93	0.89	0.98
	recall	0.98	0.95	0.95	0.91	0.99
	F1-score	0.97	0.94	0.94	0.90	0.98
VGG-19		3 × 6				
	accuracy	98.53	98.17	98.23	96.54	98.71
	precision	0.97	0.95	0.91	0.93	0.98
	recall	0.99	0.99	0.96	0.95	0.99
	F1-score	0.98	0.97	0.93	0.94	0.98
ResNet-34		3 × 3				
	accuracy	98.97 ± 0.05	98.40 ± 0.12	98.45 ± 0.07	97.05 ± 0.06	99.24 ± 0.10
	precision	0.97	0.96	0.96	0.95	0.99
	recall	0.99	0.99	0.99	0.96	0.99
	F1-score	0.98	0.97	0.97	0.96	0.99
ResNet-152		2 × 3				
	accuracy	98.85 ± 0.04	95.90 ± 0.09	98.48 ± 0.06	88.20 ± 0.08	99.40 ± 0.05
	precision	0.97	0.93	0.99	0.87	0.99
	recall	0.98	0.93	0.99	0.87	0.99
	F1-score	0.97	0.93	0.99	0.87	0.99
DenseNet-121		3 × 5				
	accuracy	99.24 ± 0.05	98.97± 0.02	98.83 ± 0.10	98.81 ± 0.07	99.32 ± 0.03
	precision	0.98	0.98	0.99	0.98	0.99
	recall	0.99	0.99	0.98	0.98	0.99
	F1-score	0.98	0.98	0.98	0.98	0.99
DenseNet-161		2 × 2				
	accuracy	99.02 ± 0.03	98.87 ± 0.02	98.88 ± 0.03	98.76 ± 0.07	99.41 ± 0.07
	precision	0.96	0.98	0.97	0.97	0.99
	recall	0.99	0.99	0.99	0.99	0.99
	F1-score	0.97	0.98	0.97	0.98	0.99

3.3. The Graphical Comparison of the Accuracies

In the Figures 10–15, the accuracies given by Ps-ProtoPNet are graphically compared with the accuracies given by the other models. As mentioned in Section 3.2, the accuracies of these models stabilize before 30 epochs, but we trained and tested the models for 100 epochs over the dataset of CT-scan images [44]. In Figure 10, the comparison of the accuracies given by the models with baseline VGG-16 is provided. The curves of colors green, purple, yellow, brown and blue sketch the accuracies of Ps-ProtoPNet, Gen-ProtoPNet, NP-ProtoPNet, ProtoPNet and VGG-16, respectively. Although, it is hard to see the difference between the accuracies in the Figures 10–15, but the figures clearly show the difference between the accuracies before they stabilize.

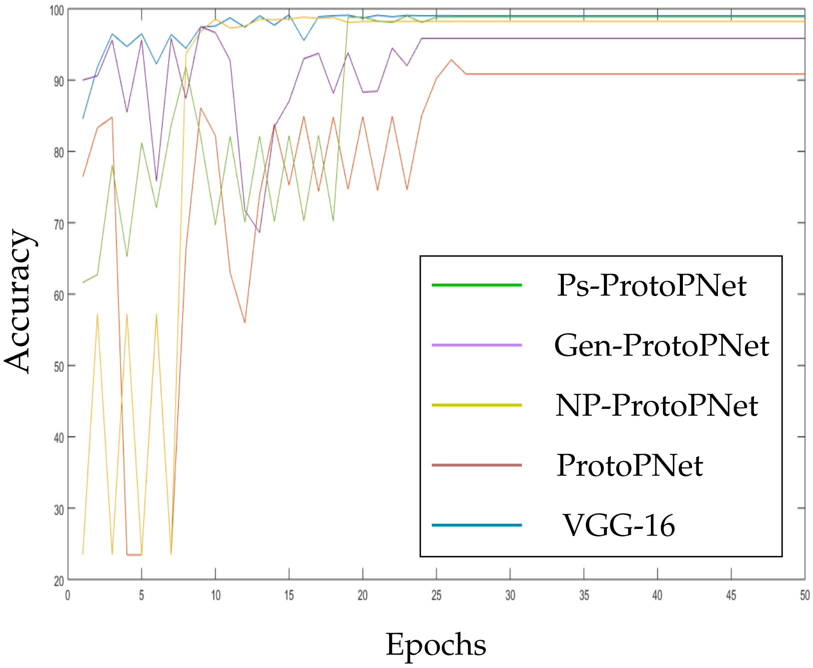

Figure 10. Ps-ProtoPNet with baseline VGG-16.

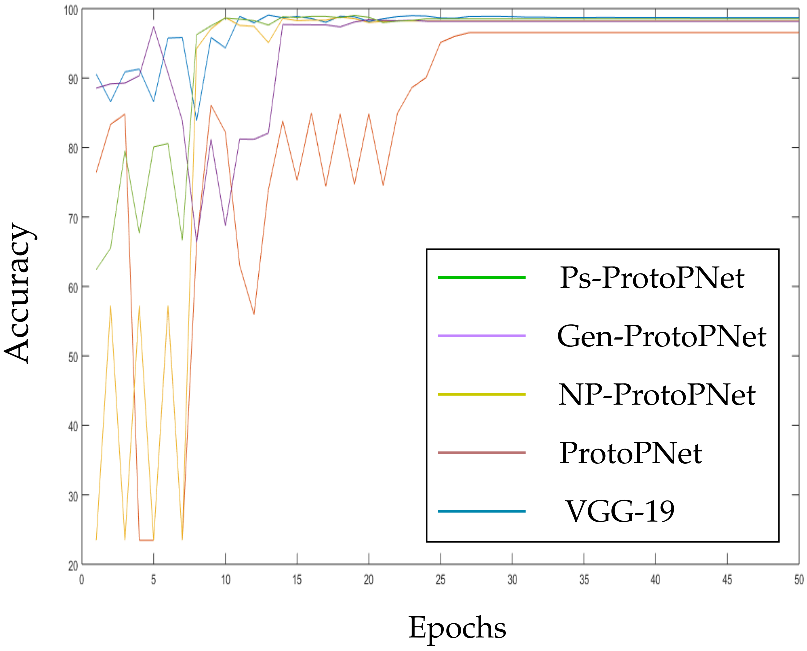

Figure 11. Ps-ProtoPNet with baseline VGG-19.

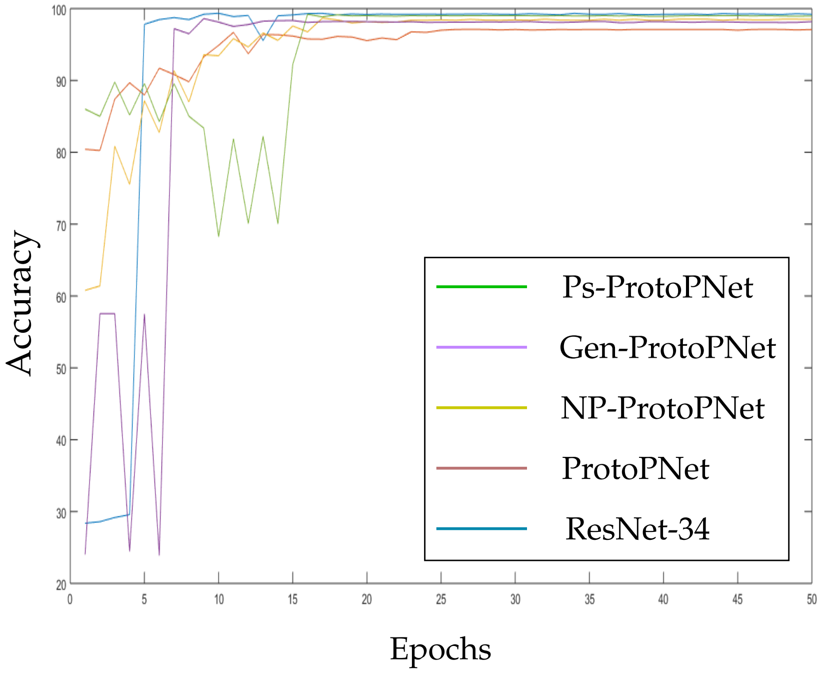

Figure 12. Ps-ProtoPNet with baseline ResNet-34.

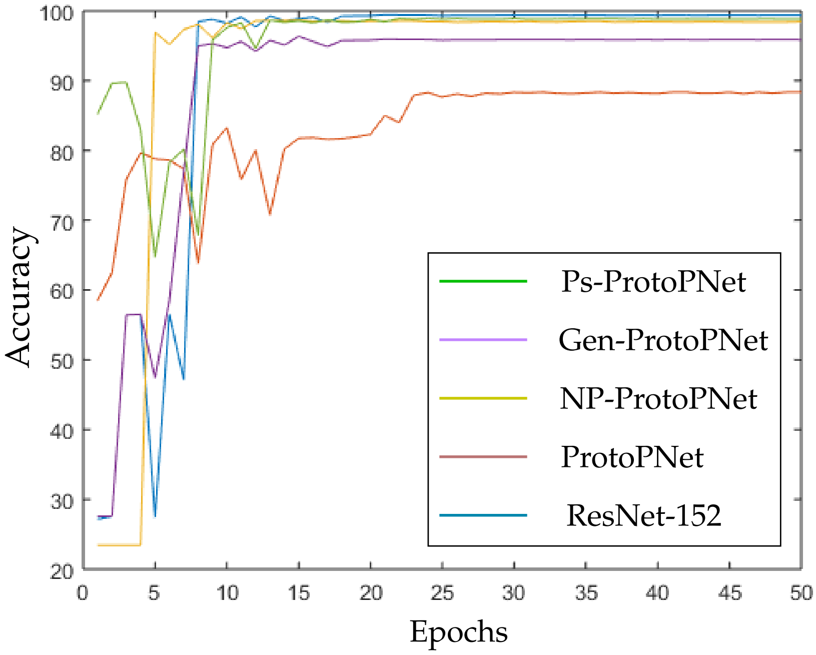

Figure 13. Ps-ProtoPNet with baseline ResNet-152.

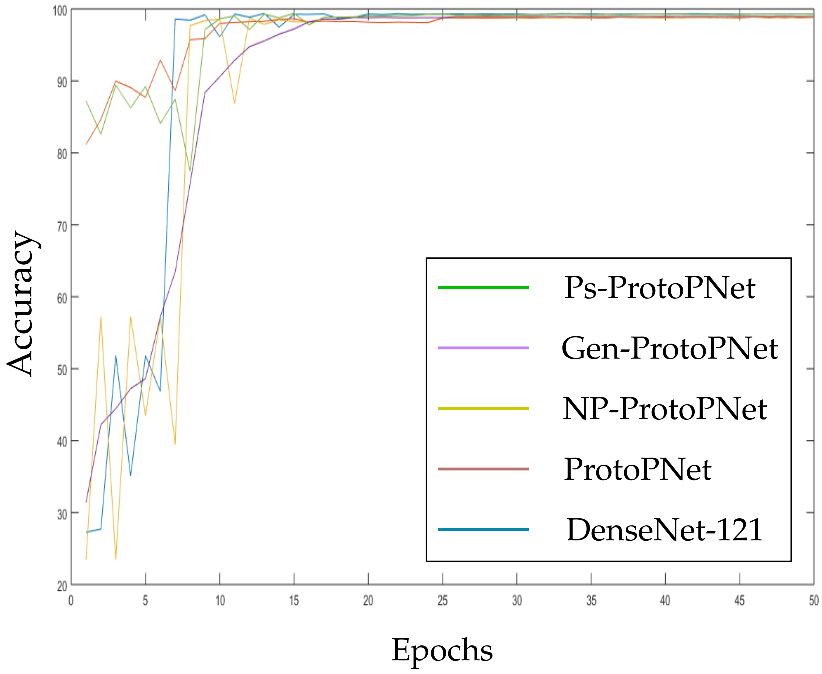

Figure 14. Ps-ProtoPNet with baseline DenseNet-121.

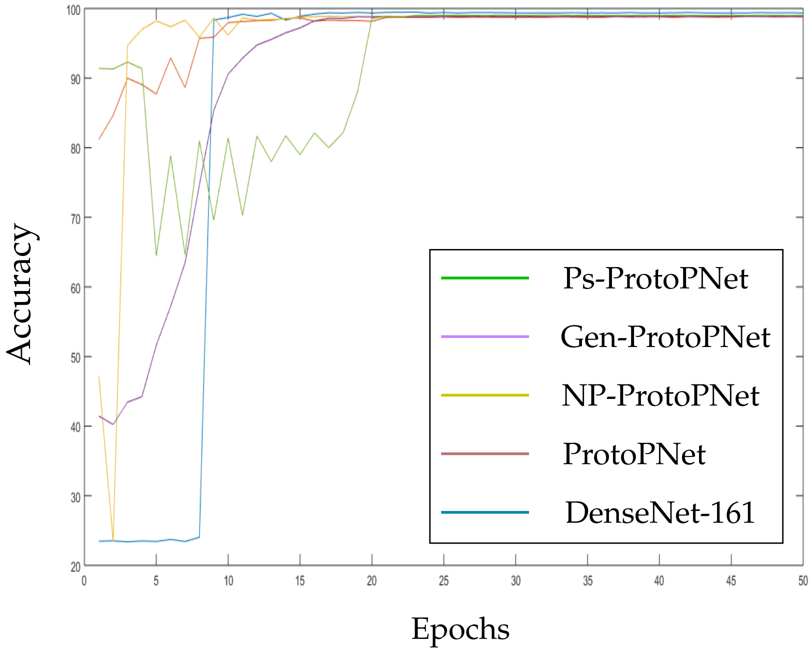

Figure 15. Ps-ProtoPNet with baseline DenseNet-161.

3.4. The Test of Hypothesis for the Accuracies

Since accuracy is the proportion of correctly classified images among all the test images, we can apply the test of hypothesis concerning system of two proportions. Let n be the size of test dataset, and the number of images correctly classified by model 1 and 2 are x_1 and x_2, respectively. Let $\tilde{p}_1 = x_1/n$ and $\tilde{p}_2 = x_2/n$. The statistic for test concerning difference between two proportions is given by [57]:

$$Z = \frac{\tilde{p}_1 - \tilde{p}_2}{\sqrt{2\tilde{p}(1-\tilde{p})/n}}, \quad \text{where } \tilde{p} = (x_1 + x_2)/2n. \tag{8}$$

Let p_1 and p_2 be the accuracies given by model 1 and 2. Therefore, our hypothesis is as follows:

$H_0 : (p_1 - p_2) = 0$ (null hypothesis)

$H_a : (p_1 - p_2) \neq 0$ (alternative hypothesis)

We test the hypothesis for the level of confidence (α) = 0.05. Since the hypothesis is two-tailed, the p-value must be less than 0.025 to reject the null hypothesis. In the above hypotheses, p_1 is the accuracy given by Ps-ProtoPNet and p_2 represents the accuracies given by Gen-ProtoPNet, NP-ProtoPNet, ProtoPNet and base models. We obtain the values of test statistic Z from the above formula given by the Equation (8). Then the corresponding p-values are obtained from the standard normal table (Z-table). The complete list of p-values is given in the Table 2. For example, when VGG-16 is used as a base model, the p-values obtained from the accuracy given by Gen-ProtoPNet in pairs with accuracies given by Gen-ProtoPNet, NP-ProtoPNet and ProtoPNet are $0.0002, 0.0002, 0.0002$ and 0.0367, respectively. Since $\alpha = 0.05$, we reject the null hypothesis for all the p-values listed in the Table 2 except the five p-values written in bold. The p-values in bold in the last column means the accuracies given by Ps-ProtoPNet are not statistically different from accuracies given by the three base models. However, we can say with 95% confidence that the accuracies given by Ps-ProtoPNet are better than the corresponding accuracies given by Gen-ProtoPNet, NP-ProtoPNet and ProtoPNet except in the two cases.

Table 2. The p-values obtained with the test of hypothesis for system of two proportions (accuracies) between our proposed model, Ps-ProtoPNet, and each of the other model.

Base (B)	Gen-ProtoPNet [17]	NP-ProtoPNet [18]	ProtoPNet [16]	B Only
VGG-16	0.0002	0.0002	0.0002	**0.0367**
VGG-19	0.0007	0.0036	0.0002	**0.0409**
ResNet-34	0.0002	0.0002	0.0002	0.0002
ResNet-152	0.0002	0.0002	0.0002	0.0002
DenseNet-121	0.0002	0.0002	0.0002	**0.0582**
DenseNet-161	**0.0467**	**0.0582**	0.0075	0.0002

3.5. The Impact of Change in the Hyperparameters of the Last Layer

In this section, we prove a theorem analogous to [16], Theorem 2.1. Our experiments show that $w_m^{(k,j)}$ can hardly be made equal to 0 for $p_j^k \notin P^i$ during the training, an assumption made in [16], Theorem 2.1. Therefore, we don't assume this condition.

Theorem 1. *Let $h \circ p_p \circ \ell$ be a Ps-ProtoPNet. For a class k, let b_l^k and a_l^k be the values of l-th prototype for class k before the projection of p_l^k and after the projection of p_l^k, respectively. Let x be an input image that is correctly classified by Ps-ProtoPNet before the projection, and k be the correct class label of x. Suppose that:*

A1 $z_l^k = arg\min_{\tilde{z} \in patches(\ell(x))} d(\tilde{z}, a_l^k)$;

A2 *there exists some* δ *with* $0 < \delta < 1$ *such that:*

A2a *for all incorrect classes* $k' \neq k$ *and* $l \in \{1, \ldots, m_{k'}\}$, *we have* $d(a_l^{k'}, b_l^{k'}) \leq \theta d(z_l^{k'}, b_l^{k'}) - \sqrt{\epsilon}$, *where* ϵ *is given by* $p_p(z) = \max_{\mathcal{Z} \in patches(z)} \log\left(\frac{d^2(\mathcal{Z}, p) + 1}{d^2(\mathcal{Z}, p) + \epsilon}\right)$ *and* $\theta = \min(\sqrt{1+\delta} - 1, 1 - \frac{1}{\sqrt{2-\delta}})$;

A2b *for all* $l \in \{1, \ldots, m_k\}$, *we have* $d(a_l^k, b_l^k) \leq (\sqrt{1+\delta} - 1) d(z_l^k, b_l^k)$ *and* $d(z_l^k, b_l^k) \leq \sqrt{1-\delta}$.

Then after projection, the output logit for the correct class k can decrease at most by $\Delta = m' \log(1+\delta)(2-\delta)(1+\frac{1}{r}(n-1))$, *where* $-1/r$ *is the weight assigned to incorrect classes, and r is a positive real number.*

Proof of Theorem 1. For any class k, let $L_k(x, \{p_l^k\}_{l=1}^{m'})$ be the output logit for input image x, where $\{p_l^k\}_{l=1}^{m'}$ denote the prototypes of class k. Since negative connections between similarities scores of incorrect classes and logits are equal to $-1/r$,

$$L_k(x, \{p_l^k\}_{l=1}^{m'}) \quad = \quad \sum_{l=1}^{m'} \log\left(\frac{d^2(z_l^k, p_l^k) + 1}{d^2(z_l^k, p_l^k) + \epsilon}\right) - \frac{1}{r} \sum_{k' \neq k} \sum_{l=1}^{m'} \log\left(\frac{d^2(z_l^{k'}, p_l^{k'}) + 1}{d^2(z_l^{k'}, p_l^{k'}) + \epsilon}\right).$$

Let Δ_k be the difference between the output logit of class k before and after the projection of pr-ototypes $\{p_l^k\}_{l=1}^{m'}$ to their nearest latent training patches. Suppose $L_k(x, \{b_l^k\}_{l=1}^{m'})$ and $L_k(x, \{a_l^k\}_{l=1}^{m'})$ denotes the logits before the projection and after the projection, respectively. Therefore, we have

$$\begin{aligned}\Delta_k &= L_k(x, \{a_l^k\}_{l=1}^{m'}) - L_k(x, \{b_l^k\}_{l=1}^{m'}) \\ &= \sum_{l=1}^{m'} \log\left(\frac{d^2(z_l^k, a_l^k) + 1}{d^2(z_l^k, b_l^k) + 1} \cdot \frac{d^2(z_l^k, b_l^k) + \epsilon}{d^2(z_l^k, a_l^k) + \epsilon}\right) - \frac{1}{r} \sum_{k' \neq k} \sum_{l=1}^{m'} \log\left(\frac{d^2(z_l^{k'}, a_l^{k'}) + 1}{d^2(z_l^{k'}, b_l^{k'}) + 1} \cdot \frac{d^2(z_l^{k'}, b_l^{k'}) + \epsilon}{d^2(z_l^{k'}, a_l^{k'}) + \epsilon}\right).\end{aligned}$$

Suppose that,

$$\Psi_l^k = \frac{d^2(z_l^k, a_l^k) + 1}{d^2(z_l^k, b_l^k) + 1} \times \frac{d^2(z_l^k, b_l^k) + \epsilon}{d^2(z_l^k, a_l^k) + \epsilon}, \text{ and } \Psi_l^{k'} = \frac{d^2(z_l^{k'}, a_l^{k'}) + 1}{d^2(z_l^{k'}, b_l^{k'}) + 1} \times \frac{d^2(z_l^{k'}, b_l^{k'}) + \epsilon}{d^2(z_l^{k'}, a_l^{k'}) + \epsilon}. \tag{9}$$

Therefore,

$$\Delta_k = \sum_{l=1}^{m'} \log \Psi_l^k - \sum_{k' \neq k} \sum_{l=1}^{m'} \log \Psi_l^{k'}. \tag{10}$$

From the inequality given in the assumption (A2b), we have

$$\frac{d^2(z_l^k, a_l^k) + 1}{d^2(z_l^k, b_l^k) + 1} \geq \frac{1}{d^2(z_l^k, b_l^k) + 1} \geq \frac{1}{2 - \delta}. \tag{11}$$

By the triangle inequality, we have $d(z_l^k, a_l^k) \leq d(z_l^k, b_l^k) + d(a_l^k, b_l^k)$. Consequently,

$$\frac{d^2(z_l^k, b_l^k) + \epsilon}{d^2(z_l^k, a_l^k) + \epsilon} \geq \frac{d^2(z_l^k, b_l^k) + \epsilon}{(d(z_l^k, b_l^k) + d(a_l^k, b_l^k))^2 + \epsilon}. \tag{12}$$

Again, by $(A2b)$, we have

$$d(a_l^k, b_l^k) \leq (\sqrt{1 + \delta} - 1) d(z_l^k, b_l^k), \text{ that is,} \quad d(a_l^k, b_l^k) + d(z_l^k, b_l^k) \leq d(z_l^k, b_l^k)\sqrt{1 + \delta}.$$

On squaring both sides of the above inequality and then adding ϵ to both sides of the inequality, we obtain

$$(d(a_l^k, b_l^k) + d(z_l^k, b_l^k))^2 + \epsilon \quad \leq \quad (1 + \delta) d^2(z_l^k, b_l^k) + \epsilon \leq (1 + \delta)(d^2(z_l^k, b_l^k) + \epsilon).$$

On rearranging the above inequality, we have

$$\frac{d^2(z_l^k, b_l^k) + \epsilon}{(d(a_l^k, b_l^k) + d(z_l^k, b_l^k))^2 + \epsilon} \geq (1 + \delta). \tag{13}$$

Therefore, by inequalities (12) and (13), be obtain

$$\frac{d^2(z_l^k, b_l^k) + \epsilon}{d^2(z_l^k, a_l^k) + \epsilon} \geq \frac{(d^2(z_l^k, b_l^k) + \epsilon}{(d(a_l^k, b_l^k) + d(z_l^k, b_l^k))^2 + \epsilon} \geq (1 + \delta). \tag{14}$$

Hence, by Equations (11) and (14), we have

$$\Psi_l^k = \frac{d^2(z_l^k, a_l^k) + 1}{d^2(z_l^k, b_l^k) + 1} \times \frac{d^2(z_l^k, b_l^k) + \epsilon}{d^2(z_l^k, a_l^k) + \epsilon} \geq \frac{1}{(1+\delta)(2-\delta)}. \tag{15}$$

Now we derive an upper bound of $\Psi_l^{k'}$, where $k' \neq k$. Using the triangle inequality, we obtain

$$d^2(z_l^{k'}, a_l^{k'}) \leq (d(z_l^{k'}, b_l^{k'}) + d(a_l^{k'}, b_l^{k'}))^2 + 1.$$

Therefore,

$$\frac{d^2(z_l^{k'}, a_l^{k'}) + 1}{d^2(z_l^{k'}, b_l^{k'}) + 1} \leq \frac{(d(z_l^{k'}, b_l^{k'}) + d(a_l^{k'}, b_l^{k'}))^2 + 1}{d^2(z_l^{k'}, b_l^{k'}) + 1}. \tag{16}$$

By assumption $(A2a)$, we have

$$d(a_l^{k'}, b_l^{k'}) \leq (\sqrt{1+\delta} - 1)d(z_l^{k'}, b_l^{k'}) - \sqrt{\epsilon} \leq (\sqrt{1+\delta} - 1)d(z_l^{k'}, b_l^{k'}). \tag{17}$$

By the inequality (17), we have

$$\begin{aligned} (d(z_l^{k'}, b_l^{k'}) + d(a_l^{k'}, b_l^{k'}))^2 &\leq (d(z_l^{k'}, b_l^{k'}) + (\sqrt{1+\delta} - 1)d(z_l^{k'}, b_l^{k'}))^2 \\ &= ((\sqrt{1+\delta})d(z_l^{k'}, b_l^{k'}))^2 = (1+\delta)d^2(z_l^{k'}, b_l^{k'}). \end{aligned} \tag{18}$$

Using the inequality (18), we obtain

$$\begin{aligned} \frac{d(z_l^{k'}, b_l^{k'}) + d(a_l^{k'}, b_l^{k'}))^2 + 1}{d(z_l^{k'}, b_l^{k'})^2 + 1} &\leq \frac{(1+\delta)d(z_l^{k'}, b_l^{k'})^2 + 1}{d(z_l^{k'}, b_l^{k'})^2 + 1} \\ &\leq \frac{(1+\delta)d(z_l^{k'}, b_l^{k'})^2 + 1 + \delta}{d(z_l^{k'}, b_l^{k'})^2 + 1} = 1 + \delta. \end{aligned} \tag{19}$$

On combining the inequalities (16) and (19), we have

$$\frac{d^2(z_l^{k'}, a_l^{k'}) + 1}{d^2(z_l^{k'}, b_l^{k'}) + 1} \leq 1 + \delta. \tag{20}$$

Again, by the triangle inequality, we have

$$d(z_l^{k'}, a_l^{k'}) \geq d(z_l^{k'}, b_l^{k'}) - d(a_l^{k'}, b_l^{k'}). \tag{21}$$

Also, inequality in the assumption (A2a) implies $d(z_l^{k'}, b_l^{k'}) - d(a_l^{k'}, b_l^{k'}) > 0$. Therefore, the inequality (21) and the positivity of the expression $d(z_l^{k'}, b_l^{k'}) - d(a_l^{k'}, b_l^{k'})$ give:

$$\frac{d^2(z_l^{k'}, b_l^{k'}) + \epsilon}{d^2(z_l^{k'}, a_l^{k'}) + \epsilon} \leq \frac{d^2(z_l^{k'}, b_l^{k'}) + \epsilon}{(d(z_l^{k'}, b_l^{k'}) - d(a_l^{k'}, b_l^{k'}))^2 + \epsilon} \leq \left(\frac{d(z_l^{k'}, b_l^{k'}) + \sqrt{\epsilon}}{d(z_l^{k'}, b_l^{k'}) - d(a_l^{k'}, b_l^{k'})} \right)^2. \tag{22}$$

Again, by using the assumption (A2a), we have

$$d(a_l^{k'}, b_l^{k'}) \leq \left(1 - \frac{1}{\sqrt{2-\delta}}\right) d(z_l^{k'}, b_l^{k'}) - \sqrt{\epsilon}.$$

On simplifying the above inequality, we obtain

$$\frac{1}{\sqrt{2-\delta}} d(z_l^{k'}, b_l^{k'}) + \sqrt{\epsilon} \leq d(z_l^{k'}, b_l^{k'}) - d(a_l^{k'}, b_l^{k'}).$$

Therefore,

$$\frac{1}{\sqrt{2-\delta}}d(z_l^{k'}, b_l^{k'}) + \frac{\sqrt{\epsilon}}{\sqrt{2-\delta}} \quad \leq \quad \frac{1}{\sqrt{2-\delta}}d(z_l^{k'}, b_l^{k'}) + \sqrt{\epsilon} \leq d(z_l^{k'}, b_l^{k'}) - d(a_l^{k'}, b_l^{k'}).$$

The above inequality gives:

$$\frac{d(z_l^{k'}, b_l^{k'}) + \sqrt{\epsilon}}{d(z_l^{k'}, b_l^{k'}) - d(a_l^{k'}, b_l^{k'})} \leq \sqrt{2-\delta}. \tag{23}$$

Combining inequalities (22) and (23), we obtain

$$\frac{d(z_l^{k'}, b_l^{k'}) + \epsilon}{d(z_l^{k'}, a_l^{k'}) + \epsilon} \leq (\sqrt{2-\delta})^2 = 2-\delta. \tag{24}$$

The inequalities (21) and (24) give us

$$\Psi_l^{k'} = \frac{d^2(z_l^{k'}, a_l^{k'}) + 1}{d^2(z_l^k, b_l^k) + 1} \times \frac{d^2(z_l^k, b_l^k) + \epsilon}{d^2(z_l^{k'}, a_l^{k'}) + \epsilon} \leq (1+\delta)(2-\delta). \tag{25}$$

Therefore, by Equations (9), (10), and inequalities (14) and (25), we have

$$\Psi_l^k \geq \frac{1}{(1+\delta)(2-\delta)} \text{ and } \Psi_l^{k'} \leq (1+\delta)(2-\delta).$$

Since log is an increasing function, we have

$$\log \Psi_l^k \geq \log \frac{1}{(1+\delta)(2-\delta)} \text{ and } \log \Psi_l^{k'} \leq \log(1+\delta)(2-\delta).$$

Therefore,

$$\log \Psi_l^k \geq -\log(1+\delta)(2-\delta) \text{ and } -\log \Psi_l^{k'} \geq -\log(1+\delta)(2-\delta).$$

By the Equation (10), we have

$$\begin{aligned} \Delta_k &\geq -\sum_{l=1}^{m'} \log(1+\delta)(2-\delta) - \frac{1}{r}\sum_{k'\neq k}\sum_{l=1}^{m'} \log(1+\delta)(2-\delta) \\ &\geq -m' \log(1+\delta)(2-\delta) - \frac{1}{r}\sum_{k'\neq k} m' \log(1+\delta)(2-\delta) \\ &\geq -m' \log(1+\delta)(2-\delta)(1+\frac{1}{r}\sum_{k'\neq k}). \end{aligned} \tag{26}$$

Note that, $\sum_{k'\neq k} = n-1$, thus

$$-\Delta_k \leq m' \log(1+\delta)(2-\delta)(1+\frac{1}{r}(n-1)). \tag{27}$$

The -ve sign indicates the decrease in the logit after the projection of a prototype. □

4. Discussion

Ps-ProtoPNet is closely related to three interpretable deep learning models ProtoPNet, Gen-PrtoPNet and NP-ProtoPNet, but strikingly different from them as explained the Section 2.3. Ps-ProtoPNet uses a generalized version of the distance function *L*2 along with the non-optimization of the last layer. The non-optimization of the last layer helps to preserve the negative connection of the logits with incorrect classes that further helped to establish Theorem 1. Moreover, the non-optimization of the last layer helped us to include the negative reasoning process along with positive reasoning process.

5. Conclusions

The non-optimization of the last layer and the use of prototypes with rectangular spatial dimensions and square spatial dimensions greater than 1 × 1 helped our model to improve its performance over NP-ProtoPNet, Gen-ProtoPNet and ProtoPNet.

Author Contributions: G.S. is the first author of this article. K.-C.Y. is the corresponding author of this article. Yow reviewed and supervised this project. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), funding reference number DDG-2020-00034. Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), numéro de référence DDG-2020-00034.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study are openly available [44].

Acknowledgments: The authors are grateful to the Faculty of Engineering and Applied Sciences at the University of Regina for making arrangement of a deep learning server for them to run their experiments.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Wikipedia. Variants of SARS-CoV-2. Available online: https://en.wikipedia.org/wiki/Variants_of_SARS-CoV-2#Variants_of_Interest_(WHO) (accessed on 30 June 2021).
2. Wikipedia. COVID-19 Testing. Available online: https://en.wikipedia.org/wiki/COVID-19_testing (accessed on 24 August 2021).
3. Al-Waisy, A.S.; Mohammed, M.A.; Al-Fahdawi, S.; Maashi, M.S.; Garcia-Zapirain, B.; Abdulkareem, K.H.; Mostafa, S.A.; Kumar, N.M.; Le, D.-N. COVID-DeepNet: Hybrid Multimodal Deep Learning System for Improving COVID-19 Pneumonia Detection in Chest X-ray Images. *Comput. Mater. Contin.* **2021**, *67*, 2409–2429. [CrossRef]
4. Al-Waisy, A.S.; Al-Fahdawi, S.; Mohammed, M.A.; Abdulkareem, K.H.; Mostafa, S.A.; Maashi, M.S.; Arif, M.; Garcia-Zapirain, B. COVID-CheXNet: Hybrid deep learning framework for identifying COVID-19 virus in chest X-rays images. *Soft Comput.* **2020**, 1–16. [CrossRef]
5. Azemin, M.Z.C.; Hassan, R.; Tamrin, M.I.M.; Ali, M.A.M. COVID-19 Deep Learning Prediction Model Using Publicly Available Radiologist-Adjudicated Chest X-Ray Images as Training Data: Preliminary Findings. *Hindawi Int. J. Biomed. Imaging* **2020**, *2020*, 8828855. [CrossRef] [PubMed]
6. Chaudhary, Y.; Mehta, M.; Sharma, R.; Gupta, D.; Khanna, A.; Rodrigues, J.J.P.C. Efficient-CovidNet: Deep Learning Based COVID-19 Detection From Chest X-Ray Images. In Proceedings of the 2020 IEEE 22nd International Conference on e-Health Networking, Applications and Services, Shenzhen, China, 1–2 March 2020. [CrossRef]
7. Cohen, J.P.; Dao, L.; Roth, K.; Morrison, P.; Bengio, Y.; Abbasi, A.; Shen, B.; Mahsa, H.; Ghassemi, M.; Li, H. Predicting COVID-19 Pneumonia Severity on Chest X-ray With Deep Learning. *Cureus* **2020**. [CrossRef] [PubMed]
8. Gunraj, H.; Wang, L.; Wong, A. COVIDNet-CT: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases From Chest CT Images. *Front. Med.* **2020**, *7*, 1025. [CrossRef] [PubMed]
9. Jain, G.; Mittal, D.; Thakur, D.; Mittal, M. A deep learning approach to detect COVID-19 coronavirus with X-Ray images. *Biocybern. Biomed. Eng.* **2020**, *40*, 1391–1405. [CrossRef] [PubMed]
10. Jain, R.; Gupta, M.; Taneja, S.; Hemanth, D.J. Deep learning based detection and analysis of COVID-19 on chest X-ray images. *Appl. Intell.* **2021**. [CrossRef]
11. Kumar, R.; Arora1, R.; Bansal, V.; Sahayasheela, V.; Buckchash, H.; Imran, J.; Narayanan, N.; Pandian, G.N.; Raman1, B. Accurate Prediction of COVID-19 using Chest X-Ray Images through Deep Feature Learning model with SMOTE and Machine Learning Classifiers. *medRxiv* **2020**. [CrossRef]
12. Ozturk, T.; Talo, M.; Yildirim, E.A.; Baloglu, U.B.; Yildirime, O.; Acharya, U.R. Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Comput. Biol. Med.* **2020**, *121*, 103792. [CrossRef] [PubMed]

13. Reddy, G.T.; Bhattacharya, S.; Ramakrishnan, S.S.; Chowdhary, C.L.; Hakak, S.; Kaluri, R.; Reddy, M.P.K. An ensemble based machine learning model for diabetic retinopathy classification. In Proceedings of the 2020 international conference on emerging trends in information technology and engineering (ic-ETITE), Vellore, India, 26 December 2020; pp. 1–6.
14. Sharma, A.; Rani, S.; Gupta, D. Artificial Intelligence-Based Classification of Chest X-Ray Images into COVID-19 and Other Infectious Diseases. *Hindawi Int. J. Biomed. Imaging* **2020**, *2020*, 8889023. [CrossRef]
15. Zebin, T.; Rezvy, S. COVID-19 detection and disease progression visualization: Deep learning on chest X-rays for classification and coarse localization. *Appl. Intell.* **2020**. [CrossRef]
16. Chen, C.; Li, O.; Tao, C.; Barnett, A.J.; Su, J.; Rudin, C. This Looks Like That: Deep Learning for Interpretable Image Recognition. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019.
17. Singh, G.; Yow, K.-C. An Interpretable Deep Learning Model for COVID-19 Detection With Chest X-Ray Images. *IEEE Access* **2021**, *9*, 85198–85208. [CrossRef]
18. Singh, G.; Yow, K.-C. These Do Not Look Like Those: An Interpretable Deep Learning Model. *IEEE Access* **2021**, *9*, 41482–41493. [CrossRef]
19. Erhan, D.; Bengio, Y.; Courville, A.; Vincent, P. Visualizing Higher-Layer Features of a Deep Network. Technical Report 1341, the University of Montreal, June 2009. Also presented at theWorkshop on Learning Feature Hierarchies. In Proceedings of the 26th International Conference on Machine Learning (ICML 2009), Montreal, QC, Canada, 14–18 June 2009.
20. Hinton, G.E. A Practical Guide to Training Restricted Boltzmann Machines. In *Neural Networks: Tricks of the Trade*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 599–619.
21. Lee, H.; Grosse, R.; Ranganath, R.; Ng, A.Y. Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. In Proceedings of the 26th International Conference on Machine Learning (ICML), Montreal, QC, Canada, 14–18 June 2009; pp. 609–616.
22. Nguyen, A.; Dosovitskiy, A.; Yosinski, J.; Brox, T.; Clune, J. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems 29 (NIPS)*; NIPS: Grenada, Spain, 2016; pp. 3387–3395.
23. Simonyan, K.; Vedaldi, A.; Zisserman, A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In Proceedings of the Workshop at the 2nd International Conference on Learning Representations (ICLR Workshop), Banff, AB, Canada, 14–16 April 2014.
24. Oord, A.v.; Kalchbrenner, N.; Kavukcuoglu, K. Pixel Recurrent Neural Networks. In Proceedings of the 33rd International Conference on Machine Learning (ICML), New York, NY, USA, 19–24 June 2016; pp. 1747–1756.
25. Yosinski, J.; Clune, J.; Fuchs, T.; Lipson, H. Understanding Neural Networks through Deep Visualization. In Proceedings of the Deep Learning Workshop at the 32nd International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015.
26. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 5–12 September 2014; pp. 818–833.
27. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
28. Smilkov, D.; Thorat, N.; Kim, B.; ViÃ©gas, F.; Wattenberg, M. SmoothGrad: Removing noise by adding noise. *arXiv* **2017**, arXiv:1706.03825
29. Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic Attribution for Deep Networks. In Proceedings of the 34th International Conference on Machine Learning (ICML), San Diego, CA, USA, 7–9 May 2017; pp. 3319–3328.
30. Fu, J.; Zheng, H.; Mei, T. Look Closer to See Better: Recurrent Attention Convolutional Neural Network for Fine-grained Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 26 July 2017; pp. 4438-4446.
31. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Washington, DC, USA, 7–13 December 2015; pp. 1440–1448.
32. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 28 June 2014; pp. 580–587.
33. Huang, S.; Xu, Z.; Tao, D.; Zhang, Y. Part-Stacked CNN for Fine-Grained Visual Categorization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 30 June 2016; pp. 1173–1182.
34. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS), Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
35. Simon, M.; Rodner, E. Neural Activation Constellations: Unsupervised Part Model Discovery with Convolutional Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1143–1151.
36. Uijlings, J.R.; Sande, K.E.V.D.; Gevers, T.; Smeulders, A.W. Selective Search for Object Recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [CrossRef]

37. Xiao, T.; Xu, Y.; Yang, K.; Zhang, J.; Peng, Y.; Zhang, Z. The Application of Two-Level Attention Models in Deep Convolutional Neural Network for Fine-grained Image Classification. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference, Boston, MA, USA, 12 June 2015; pp. 842–850.
38. Zhang, N.; Donahue, J.; Girshick, R.; Darrell, T. Part-based R-CNNs for Fine-grained Category Detection. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 5–12 September 2014; pp. 834–849.
39. Zheng, H.; Fu, J.; Mei, T.; Luo, J. Learning Multi-Attention Convolutional Neural Network for Fine- Grained Image Recognition. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 Octoter 2017; pp. 5209–5217.
40. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning Deep Features for Discriminative Localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2921–2929.
41. Zhou, B.; Sun, Y.; Bau, D.; Torralba, A. Interpretable Basis Decomposition for Visual Explanation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 119–134.
42. Li, O.; Liu, H.; Chen, C.; Rudin, C. Deep Learning for Case-Based Reasoning through Prototypes: A Neural Network that Explains Its Predictions. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI), New Orleans, LA, USA, 2–7 February 2018.
43. European Institute for Biomedical Imaging Research. COVID-19 Imaging Datasets. Available online: https://www.eibir.org/COVID-19-imaging-datasets/ (accessed on 23 August 2021).
44. Kaggle. COVIDx CT-2 Dataset. Available online: https://www.kaggle.com/hgunraj/covidxct (accessed on 7 June 2021).
45. Zaffino, P.; Marzullo, A.; Moccia, S.; Calimeri, F.; Momi, E.D.; Bertucci, B.; Arcuri, P.P.; Spadea, M.F. An Open-Source COVID-19 CT Dataset with Automatic Lung Tissue Classification for Radiomics. *Bioengineering* **2021**, *8*, 26. [CrossRef]
46. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
47. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vegas, NV, USA, 30 June 2016; pp. 770–778.
48. Huang, G.; Liu, Z.; Maaten, L.v.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
49. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Juan, PR, USA, 17–19 June 2009; pp. 248–255.
50. Ghiasi-Shirazi, K. Generalizing the Convolution Operator in Convolutional Neural Networks. *Neural Process. Lett.* **2019**. [CrossRef]
51. Nalaie, K.; Ghiasi-Shirazi, K.; Akbarzadeh-T, M.R. Efficient Implementation of a Generalized Convolutional Neural Networks based on Weighted Euclidean Distance. In Proceedings of the 2017 7th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 26–27 October 2017; pp. 211–216.
52. Wikipedia. Sensitivity and Specificity. Available online: https://en.wikipedia.org/wiki/Sensitivity_and_specificity (accessed on 2 April 2021).
53. Wikipedia. Precision and Reacall. Available online: https://en.wikipedia.org/wiki/Precision_and_recall (accessed on 2 April 2021).
54. Wikipedia. F-Score. Available online: https://en.wikipedia.org/wiki/F-score (accessed on 2 April 2021).
55. Wikipedia. Accuracy and Precision. Available online: https://en.wikipedia.org/wiki/Accuracy_and_precision (accessed on 2 April 2021).
56. Wikipedia. Confusion Matrix. Available online: https://wikipedia.org/wiki/Confusion_matrix (accessed on 2 April 2021).
57. Johnson, R.A. *Miller and Freund's Probability and Statistics for Engineers*, 9th ed.; Prentice Hall International: Harlow, UK, 2011.