International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 06 | June - 2022 Impact Factor: 7.185 ISSN: 2582-3930

COVID 19 Detection Using Deep Learning Techniques

M. SakthiVel, R. Dharun Kumar, M.K. Subramanian, M. Meera Mydeen

Electronics and Communication Engineering, Velammal College of Engineering and Technology, Madurai.

***Abstract*—** It is always desirable to detect an epidemic/pandemic in a timely and accurate manner in order to prevent its spread. There can be several approaches to detecting any illness, including deep learning models. Transparency/interpretability of a deep learning model's reasoning process in relation to health science, on the other hand, is a must. As a result, we provide Gen-ProtoPNet, an interpretable deep learning model.The distance function L2 and prototypes of spacial dimension 1 1 are used in the last two models. In our approach, we employ an extended version of the distance function L2 that allows us to categorise an input using prototypes of any sort of spatial dimension, such as square and rectangular dimensions.

.

Domain – **Deep Learning**

I. INTRODUCTION

The global pandemic of Covid-19 and its variations, such as B.1.1.7, B.1.351, and P.1, is currently ongoing [18]. Efforts to manage and limit the illness have taken several forms. The virus is being detected as part of these efforts. Many methods for detecting Covid-19 from medical photos have been proposed, see e [4], [12], [22], [23], [25], [36], [37], [40], [58].The interpretability of these models' predictions is lacking, yet the interpretability of models connected to public health is critical. The goal of this project is to develop an interpretable approach for picture classification so that we can understand why an image is categorised as it is. In this paper, we describe the generalised prototype component network (Gen-ProtoPNet), an interpretable deep learning model, and test it on a dataset of three different classes of X-rays (see Section).

For example, most of the X-ray images have some part black as a background, see Figure 2. Therefore, the use of prototypical parts of spacial dimensions 1 × 1 can give good similarity scores between patches of an input image and patches of images from wrong classes, because all the images have some part black. Another example, images of birds from different sea bird species can share same background on most part of the images. So, the use of prototypes of spacial dimension 1 × 1 can wrongly give high similarity score between patches of a test image and patches of images form wrong classes, because mostly such images have water as a background.

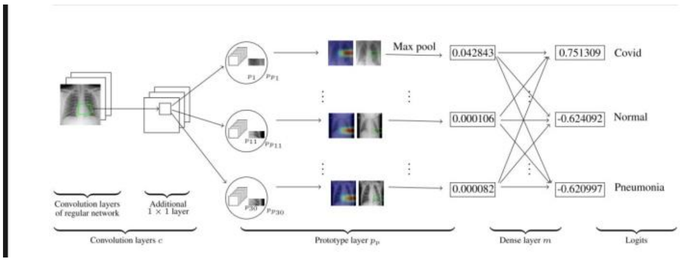
Second, photographs of items from entirely distinct classes have no tiny patches in common, thus image classification may be done only on the basis of the backdrop in the photos rather than recognising the objects themselves. Any patch of photographs of sea birds, for example, is unlike any patch of images of jungle birds. As a result, instead of recognising the birds themselves, the prototype sections of the spacial dimensions 1 1 may discern a sea bird image from a jungle bird image based on the background in the photographs.

On the other hand, employing prototype components with the largest feasible spatial dimension might impair accuracy since the prototypical part will be an image in and of itself rather than a portion of an image. As a result, there can only be a few photos that are highly similar to such a prototype whose size is comparable to the size of an image, but a smaller portion of an image can be similar to sections of many other images. Because a prototype is a representation of a component of a picture, the ideal value for the prototype's spatial dimensions is between 1 and 1.

II. PROPOSED SYSTEM:

A. *Working Principle :*

On the other hand, employing prototype components with the largest feasible spatial dimension might impair accuracy since the prototypical part will be an image in and of itself rather than a portion of an image. As a result, there can only be a few photos that are highly similar to such a prototype whose size is comparable to the size of an image, but a smaller portion of an image can be similar to sections of many other images. Because a prototype is a representation of a component of a picture, the ideal value for the prototype's spatial dimensions is between 1 and 1



B. *Methodology:*

VGG-16, VGG-19 [44], ResNet-34, ResNet-152 [17], DenseNet-121, or DenseNet-161 [20] (initialised with filters pretrained on ImageNet [9]) are the models we use to build our model. These models are referred to as baseline or basic models.

In Figure 3, we can see that the model is made up of the convolution layers of any of the three base models, followed by an extra 11 layer (we call these convolution layers c, and their parameters are all the same).

The patch p1 is the part of the original image, that is enclosed in a rectangle with green boundaries. Similarly, the patches p11 and p30 are parts of the original images given in the same row in the Figure 3. For the output z = c(x) of a test image x, the r-th prototypical unit ppr in pp calculates (with the generalized version of L2) distances between the prototypical part pr and each patch of z. These distances are inverted into similarity scores which results in an activation map of similarity scores. More the activation value, stronger the presence of a prototype in the image x. This activation map preserves the spatial relation of the convolutional output, and can be upsampled to the size of the input image to produce a heat map that identifies which part of the input image is most similar to the learned prototype [7]. These regions are enclosed in the green rectangles on the source images. The activation map is max-pooled to reduce to a single similarity score, that is, there is only one similarity score for each prototype. In the fully connected layer m, the similarity scores produced with global max-pooling are multiplied with the matrix wtm to get the logits, then these logits give prediction after normalization with softmax.

VGG-16, VGG-19 [44], ResNet-34, ResNet-152 [17], DenseNet-121, or DenseNet-161 [20] (initialised with filters pretrained on ImageNet [9]) are the models we use to build our model. These models are referred to as baseline or basic models.

In Figure 3, we can see that the model is made up of the convolution layers of any of the three base models, followed by an extra 11 layer (we call these convolution layers c, and their parameters are all the same).

A. MATHEMATICAL FORMULATION AND GEN-PROTOPNET TRAINING

The generalisation of the distance function L2 (Euclidean distance) using the basic model VGG-16 is described in this section.

We also provide the mathematical formulation and training processes of our generalised distance function method.

The normal convolution layers whose output channels have a spatial dimension of 7 7 are used to build Gen-ProtoPNet (see Section V). Assume that x is an input picture. Let z (= c(x)) have the shape (D, 7, 7), with D denoting the depth of c. (x). Consider the shape's prototype p. (D, h, w). Let I j) and (l, m) pixels of the kth tensor of z and p, respectively, be zijk and plmk.

By convolving p over z with stride, we get zp.size

equal to 1. Then zp is a tensor of the shape (D, 8 − h, 8 − w).

Therefore, each feature map of zp has (8 − h)(8 − w) pixels.

For 0 ≤ i ≤ 7 − h, 0 ≤ j ≤ 7 − w and 0 ≤ k ≤ D − 1; (i, j)

pixel (zp)ijk of the kth feature map of zp is given by:

zijkp00k + . . . + zi(j+w−1)

kp0(w−1)

k

+ z(i+1)jkp10k + . . . + z(i+1)(j+w−1) kp1(w−1)k

+ . . .

+ z(i+h−1)jkp(h−1) k + . . . + z(i+h−1)(j+w−1)kp(h−1)(w−1)k

.

(1)

Let z

2 be obtained from the Hadamard multiplication of

feature maps ofz with themselves. Let Z

2 be obtained from z

2

by convolving (overz

2

) all 1's filter of the shape of prototypes

with stride size equal to 1. Note that, Z

2

is the sum of the

patches of z

2 of the shape h × w over all feature maps.

Therefore, the shape of Z

2

is (8 − h) × (8 − w), and (i, j)

pixel Z

2

ij of Z

2

is given by:

D

X−1

k=0

z

2

ijk + z

2

i(j+1)k + . . . + z

2

i(j+w−1)k

+ z

2

(i+1)jk + z

2

(i+1)(j+1)k + . . . + z

2

(i+1)(j+w−1)k

+ . . .

Note that, z has (8 − h)(8 − w) patches of the spacial dimension h×w. Hence, the distance d 2 (Zij, p) between (i, j) patch Zij (say) of z and a prototype p is given by: d 2 (Zij, p) = Z 2 ij − 2 D X−1 k=0 (zp)ijk + D X−1 k=0 X h l=1 Xw m=1 p 2 lmk . (3) The equations 1 and 2 give the values of (zp)ijk and Z 2 ij. Thus, (zp)ijk = h P−1 l=0 wP−1 m=0 z(i+l)(j+m)kp(1+l)(1+m)k and Z 2 ij = D P−1 k=0 h P−1 l=0 wP−1 m=0 z 2 (i+l)(j+m)k . Therefore, by equation 3, d 2 (Zij, p) = X h−1 l=0 wX−1 m=0 D X−1 k=0 ||z(i+l)(j+m)k − p(1+l)(1+m)k ||2 2 . If the spacial dimension of a prototype p is 1 × 1 then h = w = 1 and d 2 (Zij, p) = D P−1 k=0 ||zijk − p11k ||2 2 , which is the square of L2 distance between a patch of z and the prototype p, where p11k ' pk . Therefore, if the spacial dimensions of p are not equal to 1 × 1 then d 2 is a generalization of the distance function L2. The distance function L2 is used in both ProtoPNet and NP-ProtoPNet to find distances of prototypes (spacial dimension 1 × 1) and the patches of images. The prototypical unit pp calculates: pp(z) = max 0≤i≤7−h, 0≤j≤7−w log d 2 (Zij, p) + 1 d 2 (Zij, p) + . Alternatively, pp(z) = max Z∈ patches(z) log_x0012_ d 2 (Z, p) + 1 d 2 (Z, p) +

. (4) The equation 4 tells us that if Z is similar to p then d 2 (Z, p) is smaller. The following three steps are performed to train our algorithm.

1) STOCHASTIC GRADIENT DESCENT (SGD) OF EVERY LAYER BEFORE DENSE LAYER At this stage of learning, Gen-ProtoPNet aim to learns important features of the image while salient parts cluster near their respective classes. To attain this aim, Gen-ProtoPNet collectively using SGD. Let X = {x1 . . . xn} be a

set of image and Y = {y1 . . . yn} is a set of corresponding labels, and D = {(xr, yr) : xr 뉫 X, yr 뉫 Y }. Our goal is to solve the following optimization problem: min P,cconv 1 n Xn r=1 CrsEnt(h 뷾 pp 뷾 c(xr), yr) + 貫1ClstCt + 貫2SepCt, (5) where ClstCt and SepCt are as follows: ClstCt = 1 n Xn r=1 min s:ps늣Pyr min Z늣patches(c(xr)) d 2 (Z, ps); (SepCt = − 1 n Xn r=1 min s:ps6∈ Pyr min Z∈ patches(c(xr)) d 2 (Z, ps). (7) The decrease in cluster cost clusters prototypical parts around their correct class, see equation 6, whereas the decrease in separation cost attempts to separate prototypical parts from their incorrect class [7], see equation 7. The decrease in the cross entropy gives better classifications, see equation 5. The coefficients λ1 is set equal to 0.8 and the coefficient λ2 belongs to the interval (0.08, 0.8). Let Pr be the set of prototypical parts of the images that belong to r-th class. For a class r, we put wt(r,s) m = 1 for all s with ps ∈ Pr and wt(r,s) m = −0.5 for all s with ps 6∈ Pr . Since similarity scores are nonnegative, in this way Gen-ProtoPNet learns a meaningful latent space [7]. 2) PUSH OF PROTOTYPICAL PARTS To see which part of the training images are used as prototypes, Gen-ProtoPNet projects every prototype ps onto the patch of the output c(x) that has smallest distance from ps , and x belong to class of ps [7]. That is, for every prototype ps of class r, Gen-ProtoPNet perform the following update: ps ←− min {Z:Z∈ patches(c(xk)) ∀k s.t. yk=r} d 2 (Z, ps). 3) OPTIMIZATION OF THE LAST LAYER To rely only on positive connections between prototypes and logits. We aim to make negative connection wt(r,s) m to 0 for all s with ps 6∈ Pr . We perform this process after fixing all the parameters of convolution layers and prototype layer, and aim to optimize [7]: min P,cconv 1 n Xn k=1 CrsEnt(h ◦ pp ◦ c(xk), yk)+λ X 3 r=1 X s:ps6∈ Pr |wt(r,s) m |.

B. SELECTION OF AN IMAGE PATCH AS A PROTOTYPE

Suppose x is the source image of a prototype pr . The patch of x that is most activated by the prototype pr is used for the visualization of pr . Its activation value must be at least 92nd percentile of all the activation values (before max-pooling) of ppr [7].

VI. EXPLANATION OF THE REASONING PROCESS OF GEN-PROTOPNET WITH AN EXAMPLE

We constructed our model over six baseline models. We trained and tested our model for 500 epochs. The model VGG-16 is used as a baseline model to run the experiments explained in this example. However, the measures of the performance of the model with the other baseline model are given in the Table 1. In the Figure 4, the test image in the first column is a member of the Covid class. In next column, each image is the test image with a rectangle (at a certain place) on it. The rectangles have green boundaries. The pixels enclosed by such a rectangle on an image in the second column correspond to the pixels on the original image in the fourth column and same

of Covid class. Since Covid is a first class these weight are entries of first row of wtm. The multiplication of the first row of the weight matrix wtm with S gives logit for the Covid class. Similarly, logit for Normal and Pneumonia classes can be obtained by multiplying second and third row of the weight matrix wtm with the matrix S, respectively. Hence, the logits for the first, second and third classes are 0.752591, −0.627040 and −0.623544, respectively. The matrix S and

VII. THE PERFORMANCE DESCRIPTION WITH CONFUSION MATRICES

The comparison of the performance of Gen-ProtoPNet with NP-ProtoPNet, ProtoPNet and the base models is made with some metrics, such as: accuracy, precision, recall and F1-score. The confusion matrices are also used to outline the performance of Gen-ProtoPNet. A confusion matrix is an array that is used to describe the performance of a classification model on a set of test data for which the true values are known [54]. True positive (TP) is the number of items correctly labeled as belonging to the positive class, that is, the items are predicted to belong to a class when they actually belong to that class. True negative (TN) is the number of items for which the model correctly predict the negative classes, that is, the items are predicted to not belonging to a class when they actually belong to other classes, see [55]. Note that, in non-binary classifications, TP and TN are the diagonal entries of the confusion matrix. False positive (FP) is the number of items incorrectly predicted as belonging to the positive class. False negative (FN) is the number of items incorrectly predicted as not belonging to the positive class, see [52]. The metrics accuracy, precision and recall in terms of the above positives and negatives are: TP + TN Total Cases , TP TP + FP and TP TP + FN , respectively. F1-score is the harmonic mean of precision and recall, that is, F1-score = 2 Precision−1 + Recall−1 , see [53]. In figures 5-10, the confusion matrices give visualization of the performance of Gen-ProtoPNet with the six baselines. Let H be any of the following six confusion matrices. Suppose (r,s) entry of the confusion matrix H is given by H[r][s]. Therefore, TP for the first class (Covid) are H[0][0], and TN are H[1][1] + H[2][2]. In addition, FP and FN for the first class are H[0][1] + H[0][2] and H[1][0] + H[2][0] respectively. Next, we describe the confusion matrix (given in Figure 5) for Gen-ProtoPNet when constructed over baseline VGG-16. Total correct predictions made by Gen-ProtoPNet with baseline VGG-16 are 749 (=242 + 119 + 388), see Figure 5. Total number of test images are 872, see Section V. Thus, the accuracy is 85.89%. The above definitions and Figure 5 give us the precision, recall and FIGURE 5. Gen-ProtoPNet with base VGG-16. F

III. RESULTS & DISCUSSION

VIII. COMPARISON OF THE PERFORMANCE OF GEN-PROTOPNET WITH THE PERFORMANCE OF NP-PROTOPNET, PROTOPNET AND THE BASELINES The convolution layers of several neural networks can be used to build the models Gen-ProtoPNet, NP-ProtoPNet and ProtoPNet. As stated in Section V, we trained and tested Gen-ProtoPNet with the baseline models over the datesets of the X-rays [13], [24]. Also, NP-ProtoPNet and ProtoPNet were examined over the same datasets and with the same base models. We trained and tested all models that are compared in Table 1 for 500 epochs. The measures of the performances of the models (GenProtoPNet, NP-ProtoPNet and ProtoPNet with the six base models) in the metrics can be found in Table 1. Also, the measures of the performance the base models themselves are given in Table 1. We explain the Table 1 with an account of the performance of each of these models with base model VGG-16. However, the measures of the performance of these models with the other five base models are also given in Table 1. In the second column of the Table 1, the spacial dimensions of prototypes corresponding to each base model are given. For example, when Gen-ProtoPNet is constructed over the base model VGG-16, and prototypes have spacial dimension 4 脳 6 then the accuracy, precision, recall and F1-score of Gen-ProtoPNet are 85.89, 0.99, 0.98 and 0.98, respectively. Similarly, the measures of the performances of the models NP-ProtoPNet and ProtoPNet with baseline model VGG-16 in the metrics accuracy, precision, recall and F1-score are 84.63, 0.97, 0.99, 0.97, and 79.93, 0.87, 0.92, 0.89, respectively. Also, the measures of the performances of VGG-16 itself (Base only) in the metrics accuracy, precision, recall and F1-score are 82.45, 0.97, 0.98 and 0.97, respectively. The performance of Gen-ProtoPNet is improved over ProtoPNet with all the base models. Also, the performance of Gen-ProtoPNet is better than the performance of NP-ProtoPNet with some baseline models, and in two cases its performance is better than the performance of the baseline models themselves.

VIII. COMPARISON OF THE PERFORMANCE OF GEN-PROTOPNET WITH THE PERFORMANCE OF NP-PROTOPNET, PROTOPNET AND THE BASELINES The convolution layers of several neural networks can be used to build the models Gen-ProtoPNet, NP-ProtoPNet and ProtoPNet. As stated in Section V, we trained and tested Gen-ProtoPNet with the baseline models over the datesets of the X-rays [13], [24]. Also, NP-ProtoPNet and ProtoPNet were examined over the same datasets and with the same base models. We trained and tested all models that are compared in Table 1 for 500 epochs. The measures of the performances of the models (GenProtoPNet, NP-ProtoPNet and ProtoPNet with the six base models) in the metrics can be found in Table 1. Also, the measures of the performance the base models themselves are given in Table 1. We explain the Table 1 with an account of the performance of each of these models with base model VGG-16. However, the measures of the performance of these models with the other five base models are also given in Table 1. In the second column of the Table 1, the spacial dimensions of prototypes corresponding to each base model are given. For example, when Gen-ProtoPNet is constructed over the base model VGG-16, and prototypes have spacial dimension 4 脳 6 then the accuracy, precision, recall and F1-score of Gen-ProtoPNet are 85.89, 0.99, 0.98 and 0.98, respectively. Similarly, the measures of the performances of the models NP-ProtoPNet and ProtoPNet with baseline model VGG-16 in the metrics accuracy, precision, recall and F1-score are 84.63, 0.97, 0.99, 0.97, and 79.93, 0.87, 0.92, 0.89, respectively. Also, the measures of the performances of VGG-16 itself (Base only) in the metrics accuracy, precision, recall and F1-score are 82.45, 0.97, 0.98 and 0.97, respectively. The performance of Gen-ProtoPNet is improved over ProtoPNet with all the base models. Also, the performance of Gen-ProtoPNet is better than the performance of NP-ProtoPNet with some baseline models, and in two cases its performance is better than the performance of the baseline models themselves.

IX. GRAPHICAL COMPARISON OF THE ACCURACIES OF THE MODELS In this section, a graphical comparison of the accuracies of Gen-ProtoPNet with the other models is provided over 100 epochs. In the figures 11-16, the curves of colors purple, yellow, blue and brown sketch the accuracies of GenProtoPNet, NP-ProtoPNet, ProtoPNet and the baselines. For example, in Figure 11, the accuracies of Gen-ProtoPNet, NP-ProtoPNet and ProtoPNet with the base model VGG-16, and the base model VGG-16 itself are depicted. The performance in accuracy of Gen-ProtoPNet is the highest with the baseline model VGG-16 and second highest for the remaining base models except VGG-19. Therefore, the curve depicting the accuracy for Gen-PrortoPNet is the highest for base model VGG-16 and second highest for the other base models.

IV. CONCLUSION

In the light of the rapidly growing COVID-19 pandemic,the need for an expeditious diagnosis of COVID-19 infection became essential. The immediate diagnosis will allow the initiation of the isolation process and adequate treatment as well. While the standard test used for the diagnosis of COVID-19 disease (RT-PCR) is usually time consuming (6 hours up to days in some centers)the need for a highly sensitive test became essential. Many studies have illustrated the utility of chest CT scan in the diagnoses of COVID-19. This paper evaluates the value of classical Deep learning techniques and the convolutional neural networks in aiding physicians to further classify patients into either COVID-19 positive or negative according to their chest CT findings, and thus facilitating their work.

REFERENCES

[1] K. H. Abdulkareem, M. A. Mohammed, A. Salim, M. Arif, O. Geman, D. Gupta, and A. Khanna, ‘‘Realizing an effective COVID-19 diagnosis system based on machine learning and IOT in smart hospital environment,’’ IEEE Internet Things J., early access, Jan. 11, 2021, doi: 10.1109/JIOT.2021.3050775.

[2] A. S. Al-Waisy, M. A. Mohammed, S. Al-Fahdawi, M. S. Maashi, B. Garcia-Zapirain, K. H. Abdulkareem, S. A. Mostafa, N. M. Kumar, and D.-N. Le, ‘‘COVID-DeepNet: Hybrid multimodal deep learning system for improving COVID-19 pneumonia detection in chest X-ray images,’’ Comput., Mater. Continua, vol. 67, no. 2, pp. 2409–2429, 2021, doi: 10.32604/cmc.2021.012955.

[3] A. S. Al-Waisy, S. Al-Fahdawi, M. A. Mohammed, K. H. Abdulkareem, S. A. Mostafa, M. S. Maashi, M. Arif, and B. Garcia-Zapirain, ‘‘COVIDCheXNet: Hybrid deep learning framework for identifying COVID-19 virus in chest X-rays images,’’ Soft Comput., pp. 1–16, Nov. 2020, doi: 10.1007/s00500-020-05424-3.

[4] M. Z. C. Azemin, R. Hassan, M. I. M. Tamrin, and M. A. M. Ali, ‘‘COVID-19 deep learning prediction model using publicly available radiologist-adjudicated chest X-ray images as training data: Preliminary findings,’’ Int. J. Biomed. Imag., vol. 2020, pp. 1–7, Aug. 2020, doi: 10.1155/2020/8828855.

[5] S. Bhattacharya, P. K. R. Maddikunta, Q.-V. Pham, T. R. Gadekallu, S. R. S. Krishnan, C. L. Chowdhary, M. Alazab, and M. J. Piran, ‘‘Deep learning and medical image processing for coronavirus (COVID-19) pandemic: A survey,’’ Sustain. Cities Soc., vol. 65, Feb. 2021, Art. no. 102589.

[6] Y. Chaudhary, M. Mehta, R. Sharma, D. Gupta, A. Khanna, and J. J. P. C. Rodrigues, ‘‘Efficient-CovidNet: Deep learning based COVID19 detection from chest X-ray images,’’ in Proc. IEEE Int. Conf. EHealth Netw., Appl. Services (HEALTHCOM), Mar. 2021, pp. 1–6, doi: 10.1109/HEALTHCOM49281.2021.9398980.

[7] C. Chen, O. Li, C. Tao, A. J. Barnett, J. Su, and C. Rudin, ‘‘This looks like that: Deep learning for interpretable image recognition,’’ in Proc. 33rd Conf. Neural Inf. Process. Syst. (NIPS), 2019, pp. 1–12.

[8] D. Dansana, R. Kumar, A. Bhattacharjee, D. J. Hemanth, D. Gupta, A. Khanna, and O. Castillo, ‘‘Early diagnosis of COVID-19-affected patients based on X-ray and computed tomography images using deep learning algorithm,’’ Soft Comput., Aug. 2020, doi: 10.1007/ s00500-020-05275-y.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, ‘‘ImageNet: A large-scale hierarchical image database,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2009, pp. 248–255.

[10] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, ‘‘Visualizing higherlayer features of a deep network,’’ presented at the Workshop Learn. Feature Hierarchies 26th Int. Conf. Mach. Learn. (ICML), Montreal, QC, Canada, Jun. 2009.

[11] J. Fu, H. Zheng, and T. Mei, ‘‘Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 4438–4446.

[12] J. P. Cohen, L. Dao, K. Roth, P. Morrison, Y. Bengio, A. F. Abbasi, B. Shen, H. K. Mahsa, M. Ghassemi, H. Li, and T. Duong, ‘‘Predicting COVID-19 pneumonia severity on chest X-ray with deep learning,’’ Cureus, vol. 12, no. 7, p. e9448, Jul. 2020