2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW) | 978-1-6654-0191-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCVW54120.2021.00051

Advanced 3D Deep Non-Local Embedded System for Self-Augmented X-Ray-based COVID-19 Assessment

F. Rundo*, A. Genovese†, R. Leotta‡, F. Scotti†, V. Piuri†, S. Battiato‡
* STMicroelectronics, ADG Central R&D, Catania, Italy francesco.rundo@st.com
† Università degli Studi di Milano, Department of Computer Science, Milan, Italy
{firstname.lastname}@unimi.it
‡ University of Catania, IPLAB - DMI, Catania, Italy leotta.rob@gmail.com, battiato@dmi.unict.it

Abstract

COVID-19 diagnosis using chest x-ray (CXR) imaging has a greater sensitivity and faster acquisition procedures than the Real-Time Polimerase Chain Reaction (RT-PCR) test, also requiring radiology machinery that is cheap and widely available. To process the CXR images, methods based on Deep Learning (DL) are being increasingly used, often in combination with data augmentation techniques. However, no method in the literature performs data augmentation in which the augmented training samples are processed collectively as a multi-channel image. Furthermore, no approach has yet considered a combination of attention-based networks with Convolutional Neural Networks (CNN) for COVID-19 detection. In this paper, we propose the first method for COVID-19 detection from CXR images that uses an innovative self-augmentation scheme based on reinforcement learning, which combines all the augmented images in a 3D deep volume and processes them together using a novel non-local deep CNN, which integrates convolutional and attention layers based on non-local blocks. Results on publicly-available databases exhibit a greater accuracy than the state of the art, also showing that the regions of CXR images influencing the decision are consistent with radiologists' observations.

1. Introduction

The standard procedure for COVID-19 detection uses the Real-Time Polimerase Chain Reaction (RT-PCR), a biological test that is time-consuming, expensive, and suffers from false negatives [28]. To overcome such drawbacks, recent studies have shown the potential of computed tomography (CT) and chest x-ray (CXR) imaging in discriminating between healthy and sick individuals [28]. In fact, with respect to RT-PCR, CT scans and CXR exhibit a higher sensitivity, faster acquisition times, and do not need costly and expendable testing kits. Especially, CXR uses imaging technologies that are cheap and currently available even in less developed countries [3, 28].

To process the samples obtained with CXR imaging, techniques based on Deep Learning (DL) are being increasingly used, with the purpose of obtaining an accurate and automatic classification that can help physicians in performing the diagnosis. In fact, DL-based methods have a high accuracy and the capability of automatically learning data representations, without the need for a handcrafted feature extraction step [10]. The main issue with DL-based methods is that they suffer from reduced accuracy when datasets have a limited dimensionality, which is often the case of the datasets of CXR images of individuals affected by COVID-19, due to the limited time frame in which it was possible to capture data ($\approx$ 1.5 years). To overcome this issue, most methods in the literature adopt transfer learning or data augmentation procedures [3, 13]. However, traditional data augmentation procedures consist in randomly applying a transformation to each input training image (e.g., rotation, flipping), which is then processed individually. No method in the literature has yet considered a data augmentation procedure in which the augmented images are collectively processed as a multi-channel image. Moreover, there are no approaches in the literature for COVID-19 detection that consider attention-based networks [33], which have shown state-of-the-art accuracy in several fields related to object detection and classification [7].

In this paper, we propose a novel method based on DL for the classification of CXR images individuals as healthy, COVID-19, or viral-induced pneumonia. Our method introduces the Spatio-Temporal Feature Generation, an original approach based on reinforcement learning for inline self-augmentation of training images. The augmented images are concatenated to form a 3D deep volume, consisting in a multiple-channel image, in which each channel represents a self-augmented image. The 3D deep volume is then processed by the 3D Non-Local DenseNet, a novel Convolutional Neural Network (CNN), which combines dense convolutional layers with attention layers based on non-local

DOI 10.1109/ICCVW54120.2021.00051

blocks. In addition, the proposed approach consists in an embedded system [9] with an end-to-end pipeline that first segments the lung region from the CXR image, then augments the training images, and lastly performs the classification.

We validated our methodology on a public dataset of CXR images containing both healthy, viral induced pneumonia and COVID-19-affected individuals, obtaining greater accuracy than the methods in the literature.

2. Related Works

Based on the type of architecture and the kind of transfer learning used, it is possible to divide DL-based approaches for COVID-19 detection using CXR images into four categories (similarly to the classification proposed in [3]): *i)* CNNs pretrained on ImageNet and shallow ML classifiers; *ii)* CNNs pretrained on ImageNet and fine tuning; *iii)* CNNs pretrained on CXR images and fine tuning; *iv)* CNNs trained from scratch. In addition, we review CNN-based methods for the segmentation of the lung region in CXR images.

CNNs Pretrained on ImageNet and Shallow ML

The works proposed in [13, 20] introduce an accuracy baseline using different kinds of transfer learning available in the literature, obtained by combining CNNs pretrained on ImageNet and a shallow classifier. The method proposed in [15] addresses the problem of limited availability of training images by adopting a CNN combining the ResNet architecture with an in-line data augmentation module. A shallow classifier is then used to process the features extracted by the CNN. The approach proposed in [27] also considers the combination of pretrained CNNs in combination with shallow ML classifiers, then introduces an ensemble of models to perform the classification and analyzes the variability in the prediction.

CNNs Pretrained on ImageNet and Fine Tuning

The method proposed in [4] introduces an activation function that improves on the sigmoid function in the cases of unbalanced datasets, such as the situations in which there are large quantities of CXR images but only a few COVID-19 samples.

Rather than modifying an existing architecture, the methods proposed in [32, 35] introduce the COVID-Net, a lightweight custom architecture for the detection of COVID-19 that is pretrained on the ImageNet database and then fine tuned on CXR images. A custom architecture is also described in [1, 31], based on regularizing the latent space and improving the generalization capability, by respectively using k-means clustering and a generative adversarial network.

Lastly, the work described in [39] proposes a two-stage methodology, in which one CNN segments the lung region using the U-Net architecture [22] while the second CNN includes a spatial attention map to predict the class of the image. A similar approach is introduced in [19], with the difference that the method splits each image into patches for the purpose of data augmentation, then aggregates the predictions from all the patches to provide a final decision.

CNNs Pretrained on CXR and Fine Tuning

The method introduced in [2] applies the capsule network architecture, pretrained on CXR images, for the classification of COVID-19 samples. Capsule networks often generalize well even in the case of small datasets [11]. A different architecture is used in the works described in [16, 29], which consider a siamese network pretrained on general-purpose CXR images and fine tuned with metric learning, with the purpose of maximizing the inter-class distance in the latent space.

Lastly, the approach presented in [41] addresses the problem of domain shift between different datasets, such as between general-purpose CXR images and COVID-19 samples, by using a semi-supervised method to regularize the latent space.

CNNs Trained From Scratch

The work described in [38] considers a two-level CNN architecture that is trained from scratch using CXR images and that can be based on existing CNNs, with one CNN used to classify the CXR image as healthy or COVID-19 and one CNN determining which of the two lungs the disease has affected. Differently than [38], the approach presented in [18] aims at obtaining completely new CNN architectures, by proposing an evolutionary algorithm that searches the best neural architecture.

CNN-based Lung Region Segmentation

The methods described in [31,34] perform the segmentation using a U-Net and transfer learning, while the approach proposed in [39] combines the U-Net with a data augmentation scheme. Similarly, the method described in [40] introduces a variation of the U-Net for segmenting the lung region from CXR images.

Differently, the approach introduced in [19] uses a fully-convolutional version of the DenseNet, while the work proposed in [38] considers an attention-based mechanism to locate the lung region in the image.

3. COVID-19 Assessment through Medical Imaging

As introduced, one of the most widely employed methods to early diagnose pulmonary progression of the Sars-Cov-2 infection is based on chest imaging analysis. It is reiterated that the target of this proposal is the design of a pipeline allowing CXR-based mass screening of the population subject to COVID-19. We remark that CXR methodology is effective and sustainable in the medical-health field. In the next paragraphs the method herein proposed will be exploited and analyzed.

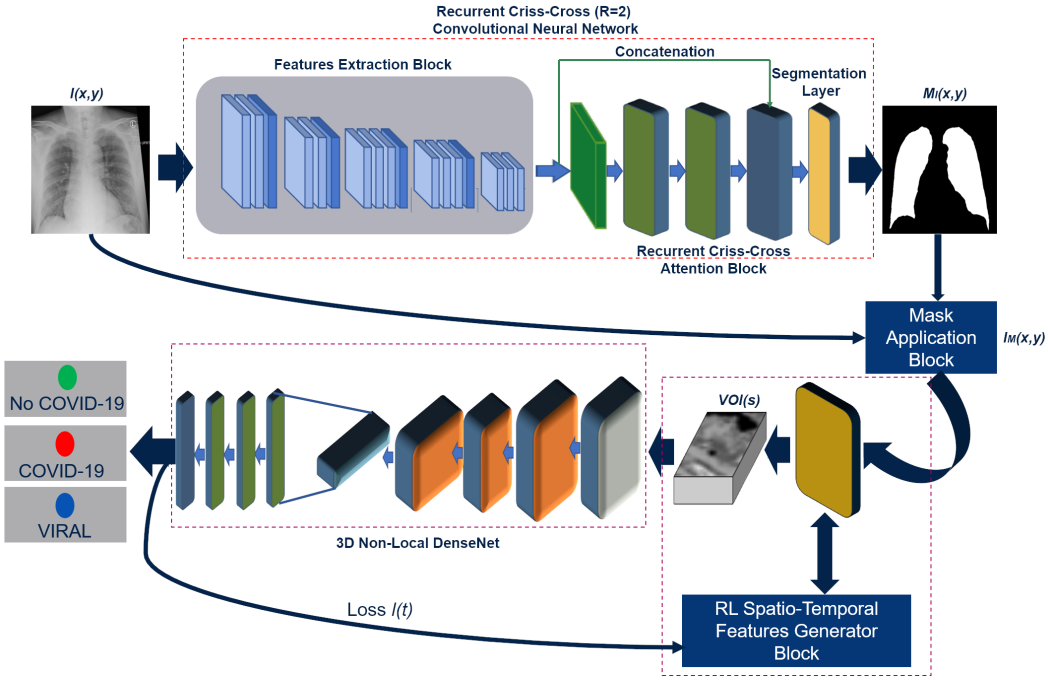

Figure 1: The proposed overall full pipeline scheme

3.1. The proposed pipeline

The overall scheme of the proposed pipeline is reported in Fig. 1. Basically, the input of the designed pipeline is the CXR image of the subject. This will be pre-processed and fed as input to the first block of the pipeline that is the "Features Extraction Block". This sub-system is able to generate the features subsequently processed by the recurrent self-attention block based on the use of the Criss-Cross algorithm [12]. The output of this block contains the lung lobes segmentation mask which will be applied to segment the source input CXR image using the "Mask Application Block". The so segmented portion of CXR containing the lobes of the lungs will then be further processed by the "RL Spatio Temporal Features Generator Block" which will perform a 2D to 3D translation using a model configured by a genetic-driven reinforcement learning algorithm. Out of this block the so generated 3D feature maps will be classified by the "3D Non-Local DenseNet" architecture which, through a densely connected architecture embedding Self-Attention mechanisms, will classify these features by associating the diagnosis of possible infection from COVID-19 or less. Each of the introduced blocks will be described in the next sections.

3.2. The Features Extraction Block

The target of this block is the extraction of the discriminating features from the patient's CXR image to be processed by the Recurrent Criss-Cross block. More in detail, the input CXR image $I(x, y)$ is passed through the designed Deep Fully Convolutional Network [17] based on ResNet-50 backbone which will produce the feature map Ψ with the spatial size of $H_s \times W_s$. In order to provide more detailed feature maps, the last two downsampling operations have been removed and dilation convolutions in the subsequent convolutional layers [12] were employed, leading to enlarge the dimension of the modified feature map Ψ to 1/8 of the input image $I(x, y)$. Therefore, from Ψ, we obtain a dimension reduced feature map Ψ_r. Then, Ψ_r is fed into the Recurrent Criss-Cross Attention module to generate a new feature map Φ which aggregates contextual information for each pixel in its criss-cross path [17].

3.3. The Recurrent Criss-Cross Attention Block

To improve full inside-image dependencies over local feature representations leveraging the well known attention mechanism [33], the authors enhanced the proposed pipeline embedding a Criss-Cross attention module [12]. As introduced in [12] the Criss-Cross attention module is able to collect contextual information in horizontal and vertical directions to enhance pixel-wise representative capability of the whole deep pipeline. More in detail, for each source feature map Ψ_r, an innovative Criss-Cross attention module computes the contextual information of all the correlated pixels on its Criss-Cross path [12]. This attention algorithm combined with further recurrent operations allows the Criss-Cross method to leverage the embedded image dependencies during the learning session of the deep network [12]. Let us formalize the attention processing embedded in the Criss-Cross module: given a local feature map $\Psi_r \in R^{C \times W \times H}$, where C is the original num-

ber of channels while $W \times H$ represents the spatial size of the feature map Ψ_r, the Criss-Cross layer applies two preliminary 1×1 convolutional layers on H in order to generate two feature maps F_1 and F_2, which belong to $R^{C' \times W \times H}$ and in which *C'* represents the reduced number of channels with respect to source C. Let define an *Affinity* function able to generate the so called Attention Map $A_M \in R^{(H+W-1)\times(W\times H)}$. The corresponding affinity operation is detailed. For each position u in the spatial dimension of F_1, we extract a vector $F_{1,u} \in R^C$. Similarly, we define the set $\Omega_u \in R^{(H+W-1)\times C}$ by extracting feature vectors from F_2 at the same position *u*, so that, $\Omega_{i,u} \in R^{C'}$ is the i-th element of Ω_u. Taking into account the above operations, we can define the introduced *Affinity* operation as follows:

$$\delta^A_{i,u} = F_{1,u}\Omega^T_{i,u} \tag{1}$$

where $\delta^A_{i,u} \in D$ is the affinity potential i.e. the correlation intensity between features $F_{1,u}$ and $\Omega_{i,u}$, for each $i = [1, ..., H + W - 1]$, and $D \in R^{(H+W-1)\times(W\times H)}$. At this stage, we further apply a softmax layer on D over the previously computed channel dimension to calculate the attention map A_M. Finally, another convolutional layer with a 1×1 kernel will be applied on the feature map H to generate the re-mapped feature $\vartheta \in R^{C\times W\times H}$ to be used for spatial adaptation. At each position u in the spatial dimension of ϑ, we can define a vector $\vartheta_u \in R^C$ and a set $\Phi_u \in R^{(H+W-1)\times C}$. The set Φ_u is a collection of feature vectors in ϑ having the same row or column at the processed position *u*. At the end, the so processed contextual information will be obtained by a further function i.e. the *Aggregation* operation defined as follows:

$$H'_u = \sum_{i=0}^{H+W-1} A^{i,u}_M \Phi_{i,u} + H_u \tag{2}$$

where H'_u is a feature vector in $H' \in R^{C\times W\times H}$ at position *u* while $A^{i,u}_M$ is a scalar value at channel *i* and position *u* in A_M. The so defined contextual information H'_u is then added to the given input feature map Ψ_r to augment the pixel-wise representation and aggregating context information. The result of this pixel-wise augmentation algorithm is the enhanced feature map Φ. More details about Criss-Cross features processing are presented in [12]. The so introduced Criss-Cross attention module is able to capture contextual information in horizontal and vertical directions but the connections between one pixel and its around (pixel neighborhood) are not covered. To overcome this issue, a Recurrent Criss-Cross processing has been proposed in [12]. The Recurrent Criss-Cross approach allows the single Criss-Cross operations to be unrolled into R loops. For our purpose, we configured $R = 2$. At the end, the so processed feature map Φ will be fed into the segmentation layer as implemented in [12] to predict the final segmentation mask $M(x, y)$.

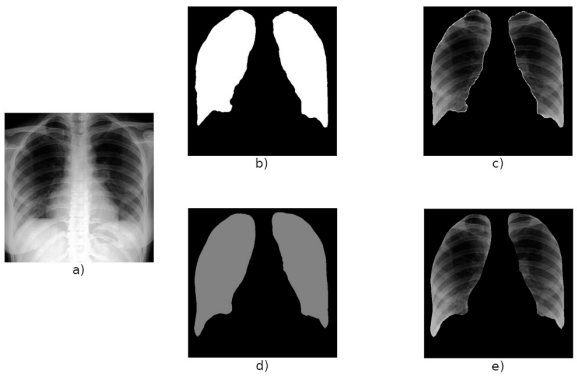

Figure 2: Chest X-Ray (CXR) Image Segmentation Process: a) Source CXR image; b) Segmentation Mask Ground Truth (GT); c) Overlay with GT; d) Predicted Segmentation Mask; e) Overlay with Predicted Mask

The "Mask Application block" will be employed to apply the so generated segmentation mask $M(x, y)$ to the input CXR image $I(x, y)$ in order to extract the lung lobes from the planar radiographic image. The resulting segmented CXR image $I_M(x, y)$ will be reduced to 256×256 through bi-cubic resizing. In Fig. 2*d* we report an instance of the so generated CXR lungs mask $M(x, y)$ compared with the Ground Truth mask reported in Fig. 2*b*. The corresponding segmented lung lobes are reported in Fig. 2*c* and Fig. 2*e* respectively.

3.4. The RL Spatio-Temporal Features Generator Block

The target of this block is the data augmentation by using a 2D-to-3D intelligent generation of the discriminating features retrieved from the input segmented CXR lung lobes. As previously described, the radiographic image contains few discriminating features, the more the COVID-19 disease is at the beginning. Certainly CT-scan based imaging would be more discriminating but significantly more invasive than CXR therefore not applicable as mass screening methodology. For this reason it was needed to apply an intelligent features augmentation method. To this end we have implemented a features generation/augmentation mechanism based on the use of Cellular Non-Linear networks [23] combined with a reinforcement learning approach driven by a genetic-like optimization routine. Basically, we have designed ad-hoc generative model in order to enhance the discriminating features of the input CXR image. This block is now detailed.

The proposed generative model embeds ad-hoc configured transient-response 2D Cellular Non-linear Networks (2D-CNN) [23]. The paradigm of the transient-response 2D-CNN allows such spatio-temporal processing of the input data. In the designed 2D-CNN, the cell denotes the basic unit [23]. Each cell of the 2D-CNN is connected only

to its neighbor cells [6, 23]. The so designed 2D-CNN's cells can interact directly with each other within the defined neighborhood [6]. Moreover, the cells not directly connected together (out of the neighborhood) may affect each other indirectly because of the propagation effects of the dynamics of 2D-CNN specified below. Specifically, in our proposed transient-response 2D-CNN every single cell of the designed network dynamically evolves from the initial state along a well defined trajectory that converges —in a time-transient session— to a specific steady-state equilibrium point [8, 23, 25, 36]. Formally, the proposed transient-response 2D-CNN mathematical model is defined as follows:

$$
\begin{aligned}
C\frac{dx_{ij}(t)}{dt} = & -\frac{1}{R_x}x_{ij}+ \\
& + \sum_{C(k,l)\in N_r(i,j)} A_1(i,j;k,l)y_{kl}(t)+ \\
& + \sum_{C(k,l)\in N_r(i,j)} A_2(i,j;k,l)u_{kl}(t)+ \\
& + \sum_{C(k,l)\in N_r(i,j)} A_3(i,j;k,l)x_{kl}(t)+ \\
& + \sum_{C(k,l)\in N_r(i,j)} D_1(i,j;k,l)(y_{ij}(t),y_{kl}(t))+ \\
& + K_b \\
& 1\leq i\leq M, 1\leq j\leq N
\end{aligned} \tag{3}
$$

$$
y_{ij}(t) = \frac{1}{2}(|x_{ij}(t)+1| - |x_{ij}(t)-1|) \tag{4}
$$

$$
\begin{aligned}
N_r(i,j) = & \{C_r(k,l); (max(|k-i|,|i-j|)\leq r)\} \\
& (1\leq k\leq M, 1\leq l\leq N)
\end{aligned} \tag{5}
$$

The Eqs. (3)-(5) characterize the space-time dynamics of the cells (and related neighborhood) of the implemented 2D-CNN. As introduced, we developed a 2D-CNN which takes the segmented CXR lung lobes $I_M(x,y)$ as input u_{kl} and state x_{kl}. This means that each pixel of the segmented CXR image will be fed as the input u_{kl} and state x_{kl} of each cell of the designed 2D-CNN. Specifically, we will consider a network of dimensions equal to that of the $I_M(x,y)$ ie 256×256. In Eqs. (3)-(5), the $Nr(i,j)$ represents the neighborhood of each 2D-CNN cell $C(i,j)$, taking into account a radius r. The variable $y_{ij}(t)$ represents the output generated hierarchical feature. The matrices $A_1(i,j;k,l)$, $A_2(i,j;k,l)$, $A_3(i,j;k,l)$, $D_1(i,j;k,l)$ represent the so called *cloning templates* while K_b denotes a bias coefficient. In the pipeline herein described, we configured the cloning templates as 3×3 matrices with a 1×1 scalar bias, all randomly initialized. Clearly, for each setup of the cloning templates and bias, the proposed 2D-CNN will generate a feature by applying the model referred to in the Eqs. (3)-(5), converging to a steady-state after its transient evolution. Therefore we investigated to understand how many setups were necessary to generate a set of discriminating features from each segmented CXR. After several heuristic tests, we have ascertained that an excellent trade-off in terms of performance and computational costs is represented by a framework of 32 separate setups of cloning templates and bias. This means in a nutshell, that for each segmented CXR a set of 32 discriminating features will be generated by the designed 2D-CNN, each through a transient evolution of the model reported in Eqs. (3)-(5) and with ad-hoc configuration of the cloning templates and bias. Moreover, for each of the defined 32 setups we randomly initialized a further set of 3×3 binary masks $A_1^{B^v}(i,j;k,l)$ and $A_2^{B^v}(i,j;k,l)$ for the $v-th$ template matrices $A_1^v(i,j;k,l)$ and $A_2^v(i,j;k,l)$ with $v=1,2,...,32$. From the tests carried out we noticed that no significant gain in terms of performance was obtained by extending the update of the coefficients also to the matrices templates $A_3(i,j;k,l)$, $D_1(i,j;k,l)$, but only a higher computational cost. Therefore, once initialized, these matrices are no longer updated during the training phase. As reported in Fig. 1, during the training of the overall pipeline, the temporal dynamics of the overall loss $L(t)$ will be retro-propagated to this block, defining the elements of the matrices $A_1^v(i,j;k,l)$ and $A_2^v(i,j;k,l)$. The configuration of the matrices is performed by using the proposed reinforcement learning algorithm. More in detail, we determined the optimal policy P_o that optimizes the cumulative discount reward R:

$$
P_0 = argmax_{P_0}\ E[\sum_{t\geq 0}\gamma^t R(\cdot|s_t,a_t)|P_0] \tag{6}
$$

where γ denotes a proper discounted coefficient in $(0,1)$. In order to evaluate the state s_t (which represents a specific setup of the $v-th$ cloning templates and bias) and the goodness of a coupled state-action (s_t,a_t), we defined the correlated value function $V^{P_0}(s_t)$ and the Q-value function $Q^{P_0}(s_t,a_t)$ respectively:

$$
V^{P_0}(s_t) = E[\sum_{t\geq 0}\gamma^t R(.|s_t)|P_0] \tag{7}
$$

$$
Q^{P_0}(s_t,a_t) = E[\sum_{t\geq 0}\gamma^t R(.|s_t,a_t)|P_0\ ] \tag{8}
$$

In order to find an optimal policy P_o for assigning the coefficients of the cloning templates and the bias that would maximize the ability of the whole pipeline to correctly discriminate the 2D-CNN generated features we decided to correlate the reward function with the retro-propagated loss $L(t)$ of the downstream deep classifier as reported in Fig. 1. Specifically, the reward function will be defined as follows:

$$
\begin{aligned}
R = & -(\frac{\partial L(A_m^v(\cdot),D_1^v(\cdot),K_b^v(\cdot),A_p^{B^v}(\cdot),B_v,v,t)}{\partial t})^2 \\
& m=1,2,3; p=1,2; v=1,2,...,32
\end{aligned} \tag{9}
$$

where $L(\cdot)$ denotes the loss of the overall pipeline which depends on the state s_t (2D-CNN setup: $A_1(i,j;k,l)$, $A_2(i,j;k,l)$, $A_3(i,j;k,l)$, $D_1(i,j;k,l)$ and bias K_b) and the actions a_t while the policy P_0 is defined by the related update of the $A_1^{B^v}(i,j;k,l)$, $A_2^{B^v}(i,j;k,l)$ and B_v masks (representing the setup among the defined 32 which will be updated).

For each training iteration t_γ, a classical genetic algorithm through common *crossover* and *mutation* operations [26] applied to the binary mask B_v , selects the $v-th$ feature configurations to modify (among the defined 32 setups). Through the same *crossover* and *mutation* operations, the proposed algorithm changes the binary masks $A_1^{B^v}(i,j;k,l)$ and $A_2^{B^v}(i,j;k,l)$ of the selected $v-th$ setups, thus identifying the coefficient of the cloning templates $A_1(i,j;k,l)$, $A_2(i,j;k,l)$ which will be updated (together with the bias) by means of a random update (action a_t) generating a new setup of spatio-temporal (time t_γ) cloning templates $A_1^v(i,j;k,l,t_\gamma)$, $A_2^v(i,j;k,l,t_\gamma)$ and bias. The others templates remain unchanged with respect to initial configuration. Only the so generated adaptive setup which produces a decrease in the overall loss dynamic $L(t)$ will be accepted while the others will be discarded. At the end of the training phase, for each segmented CXR image, we have obtained a $32 \times 256 \times 256$ Volume of Interest (VOI) which optimizes the loss of the whole pipeline.

3.5. The 3D Non-Local DenseNet

The aim of this block is the classification of the generated 3D features by the previous RL Spatio-Temporal Features Generator block. Considering that these features require a specific classification capability, we have decided to use a densely connected deep classifiers embedding self-attention mechanisms based on Non-Local Blocks [37]. The designed network architecture consists of a sequence of 3D dense blocks. The first convolution layer processes the input volume (VOI) with a size of $32 \times 256 \times 256$ pixels using a kernel size of $3 \times 3 \times 3$ pixels. The output of this layer is processed by further dense blocks composed by $[6, 8, 8, 8, 8, 8, 6]$ 3D layers respectively, that also have a kernel of $3 \times 3 \times 3$ in size. The output is followed by a ReLU non-linear activation function. Moreover, each dense block is preceded by $[0, 1, 2, 3, 4, 5, 6]$ Embedded Gaussian Non-Local blocks [36], respectively. Finally, a transition-down layer with a $2 \times 2 \times 2$ max pooling completes the block. In short, the input VOI is processed by the described blocks (both dense and non-local) generating the feature maps which will gradually decrease (in dimension) until they become a one-dimensional vector having length of 864×1. The resulting feature map traverses three fully connected (FC) layers followed by ReLU. The final layer consists of a fully connected layer which outputs trained values to a softmax layer for a final classification. In Table 1, the details of the overall architecture are reported.

Table 1: The layers specification of the proposed Deep Architecture

Block	Output Size	Layer(s) Description	Layers Numbers
Convolution	$32 \times 16 \times 256 \times 256$	$3 \times 3 \times 3$ conv.	1
Dense Block	$128 \times 16 \times 256 \times 256$	Batch Normalization ReLU $3 \times 3 \times 3$ depth-wise conv. $1 \times 1 \times 1$ point-wise conv.	6
Transition Layer	$128 \times 8 \times 128 \times 128$	$1 \times 1 \times 1$ conv. $2 \times 2 \times 2$ maxpool	1
Dense Block	$256 \times 8 \times 64 \times 64$	[...]	8
Transition Layer	$256 \times 4 \times 32 \times 32$	$1 \times 1 \times 1$ conv. $2 \times 2 \times 2$ maxpool	1
Dense Block	$384 \times 4 \times 32 \times 32$	[...]	8
Transition Layer	$384 \times 2 \times 16 \times 16$	$1 \times 1 \times 1$ conv. $2 \times 2 \times 2$ maxpool	1
Dense Block	$512 \times 2 \times 16 \times 16$	[...]	8
Transition Layer	$512 \times 1 \times 8 \times 8$	$1 \times 1 \times 1$ conv. $2 \times 2 \times 2$ maxpool	1
Dense Block	$640 \times 1 \times 8 \times 8$	[...]	8
Transition Layer	$640 \times 1 \times 4 \times 4$	$1 \times 1 \times 1$ conv. $2 \times 2 \times 2$ maxpool	1
Dense Block	$736 \times 1 \times 4 \times 4$	[...]	8
Transition Layer	$768 \times 1 \times 2 \times 2$	$1 \times 1 \times 1$ conv. $2 \times 2 \times 2$ maxpool	1
Dense Block	$864 \times 1 \times 2 \times 2$	[...]	6
Transition Layer	$864 \times 1 \times 1 \times 1$	$1 \times 1 \times 1$ conv. $2 \times 2 \times 2$ maxpool	1
Fully Connected	350	FC, ReLU	1
Fully Connected	250	FC, ReLU	1
Fully Connected	250	FC, ReLU	1
Classification	2	FC, Softmax	1

The designed 3D densely connected classifier embeds separable convolution layers (both depth-wise and point-wise) [24]. In our pipeline, we adopted separable convolutions in order to yield effective results with fewer computational cost. As highlighted in Table 1, each dense block is followed by a transition down layer, aiming to reduce the dimension of the feature map by half. Finally, the output of dense blocks is passed to Non-Local Blocks. Non-Local Blocks have been recently introduced [37], as a very promising approach for capturing space-time long-range dependencies and correlation on feature maps, resulting in a sort of "self-attention" mechanism. Self-attention through Non-Local Blocks aims to enforce the model to extract correlation among feature maps by weighting the averaged sum of the features at all possible positions in the generated feature maps [37]. In our pipeline, Non-Local Blocks operate on almost each convolution layer to extract feature in dependencies at multiple hierarchical levels. Formally, given a generic deep network as well as a general Non-Local Block input data x (feature map), the employed non-local operation computes the corresponding response y_i (of the given Deep block) at location i in the input data as a weighted sum of the input data at all positions $j \neq i$:

$$y_i = \frac{1}{\psi(x)} \sum_{\forall j} \zeta(x_i,\ x_j)\beta(x_j) \tag{10}$$

where $\zeta(\cdot)$ denotes a pairwise potential which describes the affinity or relationship between data positions at index i and j, respectively. $\beta(\cdot)$ is a non-linear function modulating ζ according to input data. The sum is then normalized by a factor $\psi(x)$. The parameters of potentials $\zeta(\cdot)$ are learned during model's training as follows:

Table 2: Average performance metrics for different deep learning networks for three-class classification problem.

Schemes	Models	Acc.	Prec. (PPV)	Sens. (Recall)	F1 Scores	Spec.
W.out image augm.	SqueezeNet	95.19	95.27	95.19	95.23	97.59
	MobileNetv2	95.9	95.97	95.9	95.93	97.95
	ResNet18	95.75	95.8	95.75	95.78	97.88
	InceptionV3	94.96	94.98	94.95	94.96	97.49
	ResNet101	95.36	95.4	95.36	95.38	97.68
	CheXNet	**97.94**	**96.61**	**96.61**	**96.61**	**98.31**
	DenseNet201	95.19	95.06	95.9	95.04	97.87
	VGG19	95.04	95.06	95.03	95.04	97.51
With image augm.	SqueezeNet	95.10	95.18	95.10	95.14	97.17
	MobileNetv2	96.22	96.25	96.22	96.23	97.80
	ResNet18	96.44	96.48	96.44	96.46	97.91
	InceptionV3	96.20	97.00	96.40	96.60	97.50
	ResNet101	96.22	96.24	96.22	96.23	97.80
	CheXNet	96.94	96.43	96.42	96.42	97.29
	DenseNet201	97.94	**97.95**	97.94	97.94	98.80
	VGG19	96.00	96.50	96.25	96.38	97.52
	Proposed	**98.82**	97.67	**98.82**	**98.25**	**98.82**

Notes. Acc. = Accuracy; Prec. = Precision; Sens. = Sensitivity; Spec. = Specificity.

$$\zeta(x_i,\ x_j) = e^{\Theta'(x_i)^T\ \Phi(x_j)} \tag{11}$$

Where Θ' and Φ are two linear transformations of the input data x with learnable weights $W_{\Theta'}$ and W_{Φ}:

$$\Theta'(x_i) = W_{\Theta'}x_i;\quad \Phi(x_j) = W_{\Phi}x_j;\quad \beta(x_j) = W_{\beta}x_j \tag{12}$$

For the $\beta(\cdot)$ function, we defined a common linear embedding (classical $1 \times 1 \times 1$ convolution) with learnable weights W_{β}. The normalization function ψ is:

$$\psi(x) = \sum_{\forall j} \zeta(x_i,\ x_j) \tag{13}$$

In Eqs. (10)-(13) an Embedded Gaussian setup is reported [37]. We adopted the Embedded Gaussian as specifically recommended for 3D applications by the authors in [37]. The so processed features are fed into the final layer composed by a stack of fully connected layers (with 500, 350, and 250 neurons, respectively) with softmax for final classification, i.e. to perform a discrimination between a COVID-19 induced pneumonia with respect to no COVID-19 pathology or not COVID-19 pneumonia (viral pneumonia).

4. Experimental Results

The proposed pipeline has been tested and validated on the "COVID-19 RADIOGRAPHY DATABASE" reported in [14]. The updated version of the mentioned database includes 3616 COVID-19 positive cases, 10192 normal cases and 1345 viral pneumonia CXR images. Preliminary benchmarking results based on classical deep learning solutions have been reported in [5, 21]. We tested the proposed pipeline regarding the ability to discriminate patients with early COVID-19 pneumonia versus patients with viral no-COVID-19 pneumonia or patients with normal radiography. Therefore, it is a three-class discrimination that is practically more useful for physicians as it allows not only to early report the onset of COVID-19 pneumonia but also allows to discriminate cases of viral pneumonia not induced by the COVID-19, thus avoiding unnecessary treatments for these subjects with consequent optimization of healthcare costs. In order to provide a robust benchmarking of the proposed method, we proceeded to compare the approach herein described with the deep solutions proposed in [5] considering the same splitting of the input dataset (training, validation and testing) of the three classes to be discriminated (COVID-19, Normal, No COVID-19 Viral Pneumonia). Specifically, we considered a dataset of 423 images for each class divided as follows: 304 samples for the training, 34 for the validation set and 85 for the testing set. Next, we will extend the benchmarking results to include the full set of images available in the updated version of the dataset [14] and compare the results with the deep architectures that performed best in the previous comparison. In the following, we present more details about the used setup of our pipeline.

The CXR input image will be resized with a bi-cubic algorithmic to 512×512 pixels and fed as input to the semantic segmentation block based on the Criss-Cross model with ResNet-50 backbone and recurrence coefficient equal to 2. The aforementioned semantic segmentation module has been trained for 200 epochs using the SGD algorithm as optimizer, an initial learning rate equal to 0.001 and a dropout factor equal to 0.1. The so segmented image $I_M(x, y)$ will be resized to 256×256 and augmented by means of the RL Spatio-Temporal Features Generator block configured to generate —for each segmented CXR— a VOI having a size of $32 \times 256 \times 256$. The used Cellular Neural Network backbone consists of a 3×3 cloning templates $A_1(i, j; k, l)$, $A_2(i, j; k, l)$, $A_3(i, j; k, l)$, $D_1(i, j; k, l)$ and bias K_b. Each of the generated VOIs will therefore be fed to the downstream deep classifier. Our classification architecture is based on a densely connected backbone as reported in Table 1. To perform the related training, we defined a mini-batch size of 10, an initial learning rate of $3e-4$, a number of epochs equal to 900 and the stochastic gradient descent with momentum (SGDM) algorithm as learning optimizer. The collected experimental results are reported in Table 2 and Table 3 for each of the mentioned testing setups respectively. As benchmark indicators we used the same suggested in the compared survey manuscript [5]:

$$Accuracy_{class\_i} = \\ = \frac{TP_{class\_i} + TN_{class\_i}}{TP_{class\_i} + TN_{class\_i} + FP_{class\_i} + FN_{class\_i}} \tag{14}$$

$$Prec._{class\_i} = \frac{TP_{class\_i}}{TP_{class\_i} + FP_{class\_i}} \tag{15}$$

$$Sens._{class\_i} = \frac{TP_{class\_i}}{TP_{class\_i} + FN_{class\_i}} \tag{16}$$

Table 3: Average performance metrics comparision (current dataset dimension): DenseNet-201 vs Proposed.

Models	Accuracy	Precision (PPV)	Sensitivity (Recall)	F1 Scores	Specificity
DenseNet-201	97.72	97.51	95.61	96.55	**98.78**
Proposed	**98.05**	**97.54**	**96.59**	**97.06**	**98.78**

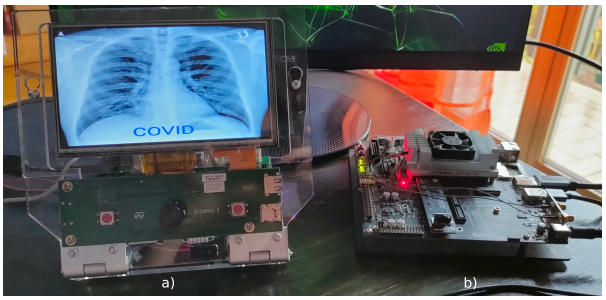

Figure 3: Innovative Point-of-Care system. The Figure shows the two embedded platforms connected to each other via IP socket, where a) represents the STA1295 Accordo5 [30] and b) represents the Jetson TX2.

$$F1\_score_{class\_i} = 2 \cdot \frac{Prec._{class\_i} \cdot Sens._{class\_i}}{Prec._{class\_i} + Sens._{class\_i}} \quad (17)$$

$$Spec._{class\_i} = \frac{TN_{class\_i}}{TN_{class\_i} + FP_{class\_i}} \quad (18)$$

where *class_i* is referred to one of the three classes i.e. COVID-19 induced pneumonia, Normal CXR or NO COVID-19 pneumonia. For each *class_i* the term TP means "True Positive", whicle FN means "False Negative", FP means "False Positive" and TN means "True Negative". For each deep architecture used as benchmarking comparison, the input CXR images have been resized according to their input size specification [5]. The following Table 2 and Table 3 report the collected comparison results (average results after 5-fold cross-validation) by considering the initial dataset dimension and the current updated dataset respectively. We remark that in Table 2 the dataset was composed by 423 images for each class splitted as 304 samples for the training, 34 for the validation set and 85 for the testing set. About the results reported in the Table 3, the dataset is composed by 3616 COVID-19 positive cases, 10192 normal cases and 1345 viral pneumonia CXR images splitted as follows: training 940, validation 200, testing 205 for each analyzed class. As confirmed from Table 2 and Table 3, our solutions outperform the compared deep solutions even though the pipeline based on DenseNet-201 (with augmentation) has better performance than ours in terms of precision having a substantially comparable specificity to our solution, which however shows a greater sensitivity and accuracy.

In Table 3 we have reported a comparison between our solution and the most performing pipeline implemented in the scientific literature, namely the DenseNet-201. This comparison —as remarked— is based on the current composition of the dataset. Even on a larger dataset, the proposed method shows significantly higher performance than the best platform proposed in the scientific literature, both in terms of accuracy and sensitivity, while showing substantially an overlapping specificity with respect to DenseNet-201.

5. Conclusions and Discussion

In this proposed scientific contribution, the authors exploited an innovative pipeline for performing COVID-19 induced pneumonia early prediction by examining a simple CXR of the subject. The CXR method allows to perform mass screening of the population due to the high sustainability of the method both in terms of costs and invasiveness (significant reduction of ionizing radiation compared to CT-scans). The proposed pipeline includes a block that performs an efficient semantic segmentation of the patient's CXR using a deep architecture embedding Criss-Cross self-attention mechanisms. The so segmented lung lobes are augmented by an innovative block that combines reinforcement learning and enhanced nonlinear cellular networks. The so generated 3D features are then classified by a deep network that embeds further self-attention layers based on Non-Local Blocks. The experimental results reported in Table 2 and Table 3 confirmed the promising performance of the proposed method also in comparison with other deep architectures. Both in the first version of the used dataset and in the recent version that includes a greater number of images, the proposed method showed superior performance compared to the state-of-the-art architectures. In addition to the advantage in terms of performance, the proposed pipeline allows a more efficient porting to embedded systems. Specifically, the authors have designed an innovative Point-of-Care consisting of two embedded platforms connected to each other via IP socket. The first platform consists of a Jetson TX2 equipped with GPU RTX 2080 with 8 Gbytes of video memory while the second is a platform with STA1295 Accordo5 Dual ARM A7 with GFX accelerator [30]. The so-designed system is shown in Figure 3. Both architectures contribute to the segmentation, augmentation, and classification of the CXR source. Specifically, the Jetson TX2 architecture hosts the segmentation and classification while the STA1295 Accordo5 platform hosts the augmentation part of the features and the correlated graphical rendering. The feed-forward inference process of the proposed pipeline takes on average 10 s for segmentation and about 20 s for classification and rendering, making the Point of Care practical in the medical field. Both platforms host a Linux-based operating system with OpenCV. Future developments are aimed at carrying out a large-scale clinical study that allows not only to assess signs of COVID-19 lung disease but also the related patient's prognosis.

References

[1] Asmaa Abbas, Mohammed M. Abdelsamea, and Mohamed Medhat Gaber. Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network. *Appl. Intell.*, 51(2):854–864, 2021. 2

[2] Parnian Afshar, Shahin Heidarian, Farnoosh Naderkhani, Anastasia Oikonomou, Konstantinos N. Plataniotis, and Arash Mohammadi. COVID-CAPS: A capsule network-based framework for identification of COVID-19 cases from x-ray images. *Pattern Recognition Letters*, 138:638–643, 2020. 2

[3] Hanan S. Alghamdi, Ghada Amoudi, Salma Elhag, Kawther Saeedi, and Jomanah Nasser. Deep learning approaches for detecting COVID-19 from chest X-ray images: A survey. *IEEE Access*, 9:20235–20254, 2021. 1, 2

[4] Joshua Bridge, Yanda Meng, Yitian Zhao, Yong Du, Mingfeng Zhao, Renrong Sun, and Yalin Zheng. Introducing the GEV activation function for highly unbalanced data to develop COVID-19 diagnostic models. *IEEE Journal of Biomedical and Health Informatics*, 24(10):2776–2786, 2020. 2

[5] Muhammad E. H. Chowdhury, Tawsifur Rahman, Amith Khandakar, Rashid Mazhar, Muhammad Abdul Kadir, Zaid Bin Mahbub, Khandakar Reajul Islam, Muhammad Salman Khan, Atif Iqbal, Nasser Al Emadi, Mamun Bin Ibne Reaz, and Mohammad Tariqul Islam. Can ai help in screening viral and COVID-19 pneumonia? *IEEE Access*, 8:132665–132676, 2020. 7, 8

[6] Leon O Chua and Lin Yang. Cellular neural networks: Theory. *IEEE Trans. on Circuits and Systems*, 35(10):1257–1272, 1988. 5

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2021. 1

[8] Elizabeth A Eisenhauer, Patrick Therasse, Jan Bogaerts, Lawrence H Schwartz, D Sargent, Robert Ford, Janet Dancey, S Arbuck, Steve Gwyther, Margaret Mooney, et al. New response evaluation criteria in solid tumours: revised recist guideline (version 1.1). *European journal of cancer*, 45(2):228–247, 2009. 5

[9] Adrian Florea and Valentin Fleaca. Implementing an embedded system to identify possible covid-19 suspects using thermovision cameras. In *Proc. of the 2020 24th Int. Conf. on System Theory, Control and Computing (ICSTCC)*, pages 322–327, 2020. 2

[10] Angelo Genovese, Mahdi S. Hosseini, Vincenzo Piuri, Konstantinos N. Plataniotis, and Fabio Scotti. Acute Lymphoblastic Leukemia detection based on adaptive unsharpening and Deep Learning. In *Proc. of the 2021 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1205–1209, 2021. 1

[11] Geoffrey E. Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with EM routing. In *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2018. 2

[12] Zilong Huang, Xinggang Wang, Yunchao Wei, Lichao Huang, Humphrey Shi, Wenyu Liu, and Thomas S. Huang. CCNet: Criss-cross attention for semantic segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2020. 3, 4

[13] Aras M. Ismael and Abdulkadir Şengür. Deep learning approaches for COVID-19 detection based on chest X-ray images. *Expert Systems with Applications*, 164:114054, 2021. 1, 2

[14] Kaggle. COVID-19 radiography database, 2018. 7

[15] Jingxiong Li, Yaqi Wang, Shuai Wang, Jun Wang, Jun Liu, Qun Jin, and Lingling Sun. Multiscale attention guided network for COVID-19 diagnosis using chest X-ray images. *IEEE Journal of Biomedical and Health Informatics*, 2021. 2

[16] Jinpeng Li, Gangming Zhao, Yaling Tao, Penghua Zhai, Hao Chen, Huiguang He, and Ting Cai. Multi-task contrastive learning for automatic CT and X-ray diagnosis of COVID-19. *Pattern Recognition*, 114:107848, 2021. 2

[17] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. of the 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 3

[18] Hassen Louati, Slim Bechikh, Ali Louati, Chih-Cheng Hung, and Lamjed Ben Said. Deep convolutional neural network architecture design as a bi-level optimization problem. *Neurocomputing*, 439:44–62, 2021. 2

[19] Yujin Oh, Sangjoon Park, and Jong Chul Ye. Deep learning COVID-19 features on CXR using limited training data sets. *IEEE Trans. on Medical Imaging*, 39(8):2688–2700, 2020. 2

[20] Elene Firmeza Ohata, Gabriel Maia Bezerra, Joao Victor Souza das Chagas, Aloisio Vieira Lira Neto, Adriano Bessa Albuquerque, Victor Hugo C. de Albuquerque, and Pedro Pedrosa Reboucas Filho. Automatic detection of COVID-19 infection using chest X-ray images through transfer learning. *IEEE/CAA Journal of Automatica Sinica*, 8(1):239–248, 2021. 2

[21] Tawsifur Rahman, Amith Khandakar, Yazan Qiblawey, Anas Tahir, Serkan Kiranyaz, Saad Bin Abul Kashem, Mohammad Tariqul Islam, Somaya Al Maadeed, Susu M. Zughaier, Muhammad Salman Khan, and Muhammad E.H. Chowdhury. Exploring the effect of image enhancement techniques on COVID-19 detection using chest x-ray images. *Computers in Biology and Medicine*, 132:104319, 2021. 7

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Proc. of the Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, 2015. 2

[23] Tamas Roska and Leon O Chua. Cellular neural networks with non-linear and delay-type template elements and non-uniform grids. *International Journal of Circuit Theory and Applications*, 20(5):469–481, 1992. 4, 5

[24] Francesco Rundo, Giuseppe Luigi Banna, Luca Prezzavento, Francesca Trenta, Sabrina Conoci, and Sebastiano

Battiato. 3d non-local neural network: A non-invasive biomarker for immunotherapy treatment outcome prediction. case-study: Metastatic urothelial carcinoma. *Journal of Imaging*, 6(12):133, 2020. 6

[25] Francesco Rundo, Giuseppe Luigi Banna, Francesca Trenta, Concetto Spampinato, Luc Bidaut, Xujiong Ye, Stefanos Kollias, and Sebastiano Battiato. Advanced non-linear generative model with a deep classifier for immunotherapy outcome prediction: A bladder cancer case study. In *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part I*, pages 227–242. Springer International Publishing, 2021. 5

[26] Adarsh Sehgal, Hung La, Sushil Louis, and Hai Nguyen. Deep reinforcement learning using genetic algorithm for parameter optimization. In *Proc. of the 2019 Third IEEE Int. Conf. on Robotic Computing (IRC)*, pages 596–601. IEEE, 2019. 6

[27] Afshar Shamsi, Hamzeh Asgharnezhad, Shirin Shamsi Jokandan, Abbas Khosravi, Parham M. Kebria, Darius Nahavandi, Saeid Nahavandi, and Dipti Srinivasan. An uncertainty-aware transfer learning-based framework for COVID-19 diagnosis. *IEEE Trans. on Neural Networks and Learning Systems*, 32(4):1408–1417, 2021. 2

[28] Feng Shi, Jun Wang, Jun Shi, Ziyan Wu, Qian Wang, Zhenyu Tang, Kelei He, Yinghuan Shi, and Dinggang Shen. Review of artificial intelligence techniques in imaging data acquisition, segmentation, and diagnosis for COVID-19. *IEEE Reviews in Biomedical Engineering*, 14:4—15, 2021. 1

[29] Mohammad Shorfuzzaman and M. Shamim Hossain. MetaCOVID: A siamese neural network framework with contrastive loss for n-shot diagnosis of COVID-19 patients. *Pattern Recognition*, 113:107700, 2021. 2

[30] STMicroelectronics. Automotive infotainment processors for display audio and cluster applications, 2018. 8

[31] S. Tabik, A. Gómez-Ríos, J. L. Martín-Rodríguez, I. Sevillano-García, M. Rey-Area, D. Charte, E. Guirado, J. L. Suárez, J. Luengo, M. A. Valero-González, P. García-Villanova, E. Olmedo-Sánchez, and F. Herrera. COVIDGR dataset and COVID-SDNet methodology for predicting COVID-19 based on chest X-Ray images. *IEEE Journal of Biomedical and Health Informatics*, 24(12):3595–3605, 2020. 2

[32] Shanjiang Tang, Chunjiang Wang, Jiangtian Nie, Neeraj Kumar, Yang Zhang, Zehui Xiong, and Ahmed Barnawi. EDL-COVID: Ensemble deep learning for COVID-19 cases detection from chest X-Ray images. *IEEE Trans. on Industrial Informatics*, pages 1–1, 2021. 2

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 1, 3

[34] Plácido L. Vidal, Joaquim de Moura, Jorge Novo, and Marcos Ortega. Multi-stage transfer learning for lung segmentation using portable x-ray devices for patients with COVID-19. *Expert Systems with Applications*, 173:114677, 2021. 2

[35] Linda Wang, Zhong Qiu Lin, and Alexander Wong. COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. *Scientific Reports*, 10(1):1–12, 2020. 2

[36] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 7794–7803, 2018. 5, 6

[37] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proc. of the 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 7794–7803, 2018. 6, 7

[38] Zheng Wang, Ying Xiao, Yong Li, Jie Zhang, Fanggen Lu, Muzhou Hou, and Xiaowei Liu. Automatically discriminating and localizing COVID-19 from community-acquired pneumonia on chest X-rays. *Pattern Recognition*, 110:107613, 2021. 2

[39] Yujia Xu, Hak-Keung Lam, and Guangyu Jia. MANet: A two-stage deep learning method for classification of COVID-19 from chest X-ray images. *Neurocomputing*, 443:96–105, 2021. 2

[40] Lipei Zhang, Aozhi Liu, Jing Xiao, and Paul Taylor. Dual encoder fusion u-net (DEFU-Net) for cross-manufacturer chest x-ray segmentation. In *Proc. of the 2020 25th Int. Conf. on Pattern Recognition (ICPR)*, 2021. 2

[41] Jieli Zhou, Baoyu Jing, Zeya Wang, Hongyi Xin, and Hanghang Tong. SODA: Detecting COVID-19 in chest X-rays with semi-supervised open set domain adaptation. *IEEE/ACM Trans. on Computational Biology and Bioinformatics*, 2021. 2