

ProgNet: Covid-19 prognosis using recurrent and convolutional neural networks

Mohamed Fakhfakh, Bassem Bouaziz, Faiez Gargouri and Lotfi Chaari, *Senior Member, IEEE*

Abstract—Humanity is facing nowadays a dramatic pandemic episode with the Coronavirus propagation over all continents. The Covid-19 disease is still not well characterized, and many research teams all over the world are working on either therapeutic or vaccination issues. Massive testing is one of the main recommendations. In addition to laboratory tests, imagery-based tools are being widely investigated. Artificial intelligence is therefore contributing to the efforts made to face this pandemic phase.

Regarding patients in hospitals, it is important to monitor the evolution of lung pathologies due to the virus. A prognosis is therefore of great interest for doctors to adapt their care strategy. In this paper, we propose a method for Covid-19 prognosis based on deep learning architectures. The proposed method is based on the combination of a convolutional and recurrent neural networks to classify multi-temporal chest X-ray images and predict the evolution of the observed lung pathology. When applied to radiological time-series, promising results are obtained with an accuracy rates higher than 92%.

I. INTRODUCTION

Coronaviruses are a large family of viruses that can cause a variety of diseases in humans, ranging from colds to Middle East Respiratory Syndrome (MERS) and Severe Acute Respiratory Syndrome (SARS). In recent months, the World Health Organization (WHO) has declared that a new coronavirus called COVID-19 has spread aggressively to several countries around the world [1]. COVID-19 can cause respiratory tract disease, fever and cough, and severe pneumonia in some extreme cases [2]. Pneumonia is an infection that causes inflammation mainly in the air sacs of the lungs responsible for oxygen exchanges [3].

The COVID-19 pandemic can be considered serious due to its high contagion and severity [4]. The impact on healthcare systems is also high due to the number of people requiring intensive care units (ICU) admission and mechanical ventilators for long periods [5]. Making evidence of COVID-19 infection is generally associated with specific lung pathologies. The most reliable way is to perform PCR (Polymerase Chain Reaction) [6] tests to assess the presence of Covid-19 RNA (Ribonucleic Acid) in the hosting individual. Because of low testing capacity especially at the very beginning of the pandemic spread, most countries performed tests generally to persons suffering from unclear and advanced symptoms, or individuals who evidently have been in contact with infected

ones.

In this sense, chest X-ray imaging is a technique that plays an important role in the diagnosis of COVID-19 disease. Radiological tests may be performed by analyzing chest X-ray images and identifying lung tissues potentially hosting infections. However, X-ray data analysis requires an expert in radiology, and may be time-consuming. Therefore, developing an automated analysis system may solve this problem and help saving precious time for radiologists. In this context, solutions based on artificial intelligence (AI) can provide an accurate and inexpensive diagnosis for COVID-19 and other types of pneumonia [7, 8]. Furthermore, it is important to assess the pathology evolution for infected people, especially those suffering from severe lung pathology who may need intensive care. This is mainly essential since no specific treatment is known up to date, which means that a treatment line adaptation may be recommended if no improvement is observed. Pathology prognosis is also useful for prioritization of patients to take in charge in case of limited numbers of ICUs.

In this paper, we propose a method for pathology prognosis based on analysing time series of chest X-ray images. The proposed method is based on both recurrent and convolutional neural networks, and allows to classify patients into two severity classes: positive or negative evolution. The main originality lies in the use of such a combination for COVID-19 prognosis.

The rest of this paper is organized as follows. After an introduction covering the context of this research and the problem to be handled, Section II is devoted to a state of the art linked to our research. In Section III, an overview of the proposed approach is presented and all the steps are detailed for the multi-temporal classification of X-ray images. An experimental validation is illustrated in Section IV. Finally, conclusions and future work are drawn in Section V.

II. RELATED WORK

Time series data refer to a consistent flow of data sets over a period of time. The analysis of these series has become a recent area of interest in artificial intelligence. Accurate forecasting is becoming more and more vital in all areas in order to make more informed and precise decisions. Time series analysis is mainly used for: *i*) descriptive analysis, i.e., identifying trends in the correlated data, *ii*) forecasting, to predict short-term events and *iii*) intervention analysis to see how the event can be evaluated during the time series.

M. Fakhfakh, B. Bouaziz and F. Gargouri are with the University of Sfax, MIRACL laboratory. E-mail: mohamed.fakhfakh.research@gmail.com, faiez.gargouri@isims.usf.tn. L. Chaari is with the University of Toulouse, IRIT - INP-ENSEEIH (UMR 5505), 2 rue Charles Camichel, BP 7122, Toulouse Cedex 7 France. E-mail: lotfi.chaari@toulouse-inp.fr

On the other hand, analysing time series generally presents the drawback related to the lack of annotated data sets. Annotations for time T_n generally have to account for times T_i with $i < n$. Having comparable data (same quality, size,...) is therefore an additional problem that leads to a difficulty in designing specific algorithms. In this sense, several applications have addressed this problem especially in clustering [9], change classification [10], change detection [11, 12], and forecasting [13].

In medical imaging, time series are often used for functional [14–16] and spectroscopic [17] imaging, as well as motion analysis [18]. In this sense, X-ray images, and specifically CT (Computerized Tomography) is widely used due to its high spatial and temporal resolutions. As regards lung diagnosis, X-ray images are usually used to track pneumonia or other specific diseases.

To analyse $3D+t$ ($3D$ +time) data, one usually need to design sophisticated algorithms that are automatic, user-friendly, fast and accurate. Today, Deep Learning (DL) has demonstrated its efficiency as stated hereabove in many fields of image analysis. Specifically, CNNs [19] are among the mostly used networks. The specificity of CNN lies in its capacity to automatically learn implicit features from images, in contrast to classical machine learning methods where the designer has to specify the features to be extracted and how to calculate them. Since 2012, several models of deep convolutional neural networks (DCNN) [20] have been proposed such as AlexNet [21], VGG [22], GoogleNet [23], Residual Net [24], DenseNet [25] and CapsuleNet [26].

As regards data related to Covid-19, very recent works have already started to design diagnosis aid tools based on DL in order to detect Covid-19 specificities in chest X-ray images. All the approaches used deep learning for image classification as Covid or non-Covid. In [27], a modified version of the ResNet-50 network has been proposed, on which a Feature Pyramid Network (FPN) is used to identify and automatically extract lesions from CT images. Using this module, the model can detect and classify CT images into three classes: healthy, COVID-19, and bacterial pneumonia. Likewise, chest radiographic images (CXR) have been used in [28]. The authors use a CNN based on various ImageNet¹ pre-trained models to extract the high-level features. Those features were fed into a Support Vector Machine (SVM) as a machine learning classifier, in order to classify the radiographic images affected by the corona effect coming.

In [29], the authors use a COVID-Net architecture with transfer learning to classify CXR images into four classes: normal bacterial infection, infection non-COVID, and COVID-19 virus. Another study [30] adopts a DeTraC deep CNN architecture [31] where the general idea is to add a class decomposition layer to the preformed models. The class decomposition layer aims to partition each class of the image dataset into several sub-classes. New labels are then assigned to the new set, where each subset is considered as an independent class, then these subsets are assembled to produce the final predictions.

Despite the efficiency and the performances obtained by

the various techniques described above for the detection of COVID-19, these approaches do not account for temporal correlations and only provide a decision about a single image. Prognosis remains therefore an open issue. Indeed, once a patient is declared infected, especially for those taken in charge in hospitals, it is important to monitor the evolution of the lung pathology. This monitoring can be done by checking blood oxygen level or other metrics. However, an accurate prognosis has mainly to be done by controlling lung images. In this paper, we propose a method to analyse time series of lung images related to infected patients in order to derive a prognosis whether the pathological state is getting better or not.

Recurrent neural networks (RNN) are a family of deep learning methods designed to manage temporal correlations between images in time series. These networks have recurrent connections in the sense that they keep information in memory: they can take into account at time T_n a number of past states T_i where $i < n$. These networks have been used in remote sensing to assess change detection in multi-spectral and hyper-spectral images [32]. RNNs are able to memorize information for a limited time and start to "forget" after a certain number of iterations, which makes training for many applications complicated. The algorithms used for updating the weight in RNNs are mainly based on the gradient with some well known practical problems such as gradient explosion.

To overcome this limitation, Long-Term Short-Term Memory (LSTM) have been proposed in [33–35] as a particular type of RNN. These models explicitly capture recursive temporal correlations and they have already proven their effectiveness in various fields, such as speech recognition [36], natural language processing [37] and image completion [38]. LSTMs have recently been used in medical imaging such in [39] where the authors propose a method with multi-modality and adjacency constraints for the segmentation of the cerebral image.

Finally, it is worth noting that LSTMs and CNNs have already been combined in a number of works like [40–42]

III. METHODOLOGY

A. General overview

The proposed methodology consists of applying a deep learning architecture for the multi-temporal classification of X-ray images in order to evaluate the COVID-19 evolution, and hence draw a vital prognosis for infected patients.

The proposed methodology is based on the combination of RNN and CNN architectures in order to assess the temporal evolution of images, and hence classify the time series into two main classes: positive and negative evolution. Such a classification can help doctors guess a vital prognosis for patients in critical situations.

Our architecture is capable of automatically learn temporal correlations of images provided as input. Fig. 1 gives a general overview of the proposed ProgNet architecture.

For an X-ray sequence $T = (T_1, T_2, \dots, T_n)$, each image T_i goes through a CNN in order to extract a characteristic vector. The obtained vector involves a set of information subsequently

¹<http://www.image-net.org/>

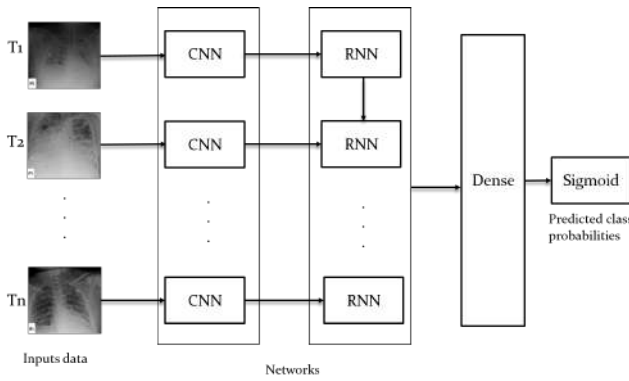


Fig. 1. Overview of the proposed ProgNet architecture.

generated by applying several convolutions and pooling layers. As a specific configuration, ResNet50 is one of the potential candidates to be used as a CNN which has shown its efficiency in different image classification applications.

An RNN is then applied to learn temporal correlations between the different characteristic vectors related to each image of a sequence T . LSTM is used due to its good performance demonstrated in the literature [33–35]. Four fully connected layer with a *sigmoid* decision function are then applied to perform binary classification.

A detailed description of the adopted networks is provided in Section III-B.

B. Used networks

1) *ResNet* : Residual networks (ResNet) is a classic CNN used as a backbone for many computer vision tasks. This model won the ImageNet challenge in 2015. In deep networks, low, medium, and high functionality and classifiers are extracted into a set of layers. ResNet mainly solves two key problems generally faced in the training deep neural networks: vanishing and exploding gradients. The core idea of ResNet is to introduce the “identity shortcut connection” which skips one or more layers. Fig. 2[left] illustrates plain layers where each convolution is connected to each other, which is the idea of all architectures before the appearance of ResNet, while Fig. 2[right] shows the residual network (Skip connection). Skipping layers allows avoiding gradient to vanish during back-propagation. Using such connections, one can train extremely deep networks that can take advantage of the power of depth, and hence allow capturing complex patterns in data.

As illustrated in Fig. 3, the ResNet50 [24] architecture used in this study is made up of 5 blocks, where each of them contains a set of convolution and max-pooling layers, followed by a skip connection.

2) *LSTM* : In some use cases, it is important to know what decisions have been made in the past in order to make an optimal decision at time t . LSTMs are adopted here since mixing long and short term dependencies in chest X-ray images is important and complementary with spatial dependencies analyzed by our CNN. LSTMs have sophisticated dynamics

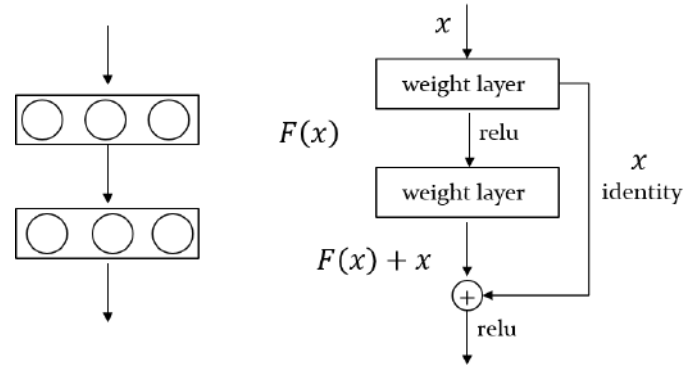


Fig. 2. Plain layers (left) and Residual blocks with skip connection (right) .

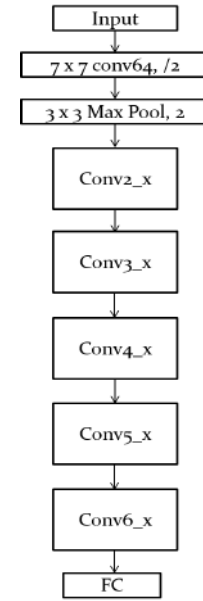


Fig. 3. Summarised overview of the ResNet50 architecture.

that allow it to easily “memorize” information for an extended number of time-steps. The “long term” memory is stored in a vector of memory cells $\{C_1, C_2, \dots, C_n\}$.

Although some differences exist in LSTM architectures, all of them have explicit memory cells for storing information for long periods of time. The LSTM can decide to overwrite the memory cell, retrieve it, or keep it for the next time step. A typical LSTM cell is illustrated in Fig. 4.

In Fig. 4, the cell input is denoted by x_t (input data), while h_{t-1} stands for the state hidden at the previous step, and C_{t-1} for the previous memory cell. The cell outputs are the next hidden state h_t and memory cell C_t .

Two main points distinguish an LSTM cell from a standard recurrent layer. First, the cell state is split into two parts, the long-term C_t and short-term h_t states. Second, three control gates along the state path (forget, input, and output gates) are added to regulate the cell states.

The first step of an LSTM cell is to decide what information will be thrown out of the cell state. The decision is made by a *sigmoid* layer called “forget gate layer (f_t)”. It evaluates

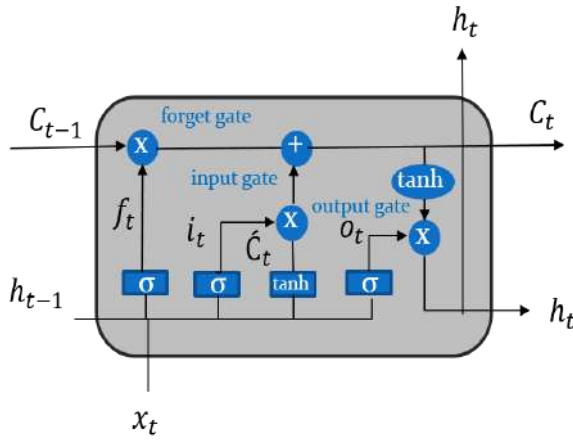


Fig. 4. Detailed LSTM cell architecture.

h_{t-1} and x_t to generate a number between 0 and 1 for each cell state C_{t-1} coefficient. The value 1 represents "completely keep this" while 0 represents "completely get rid of this". Eq.(1) shows how to control the information removal from the previous long-term state C_{t-1} :

$$f_t = \sigma(W_f[h_{t-1}, X_t] + b_f), \quad (1)$$

where σ is the *sigmoid* function, W_f and b_f corresponds to the weight matrix and bias.

The next step is to decide what new information has to be stored in the cell state. First, a *sigmoid* layer called the "input gate layer i_t " (Eq. (2)) decides which values to update. Next, a hyperbolic tangent (*tanh*) layer creates a vector of new candidate values \hat{C}_t (Eq. (3)), that will be added to the state. These elements are then combined to create a state update:

$$i_t = \sigma(W_i[h_{t-1}, X_t] + b_i), \quad (2)$$

$$\hat{C}_t = \tanh(W_c[h_{t-1}, X_t] + b_c). \quad (3)$$

As in Eq. (1), W_i and W_c stand for weight matrices, while b_i and b_c are the bias terms.

In an LSTM cell, the old cell state C_{t-1} has then to be updated into the new cell state C_t following Eq.(4):

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \hat{C}_t, \quad (4)$$

where " \cdot " is the point-wise matrix multiplication. The last step is to decide what information to produce. The output gate o_t calculation is based on the cell state following Eq. (5), while the hidden state is updated according to Eq. (6):

$$o_t = \sigma(W_o[h_{t-1}, X_t] + b_o) \quad (5)$$

$$h_t = o_t \tanh C_t \quad (6)$$

In (5), W_o and b_o correspond to the weight matrix and bias. The output of the block is recurrently connected back to the block input and all of the gates.

IV. EXPERIMENTAL VALIDATION

In this Section, we illustrate the performance of the proposed architecture for COVID-19 prognosis. Experiments are conducted on an open database of COVID-19 cases with chest radiographs². To the best of our knowledge, this is the only available dataset containing temporal acquisitions for a number of patients, with ground truth annotation as a therapeutic issue for each patient: death or survival. This database contains data for 42 patients, with up to 5 images for the same subject. For the validation of the proposed architecture, we used 34 sequences for training, each includes 3 X-ray images of the same patient, and 8 sequences for the test.

To implement our ProgNet architecture, we put four dense layers respectively, FC-1024, FC-500, FC-128, and FC-64. As regards coding, we used python programming language with Keras and Tensorflow libraries on a Intel(R) Core(TM) i7-3630QM CPU 2.40GHZ architecture with 8 Go memory.

In order to assess the performance of the proposed ProgNet architecture on the available data, we performed comparisons with three other possible configurations. Each configuration relies on a different CNN (see Fig. 1): AlexNnet, VGG16, VG19. These networks have already demonstrated an outstanding performance in the literature.

A. Loss and accuracy behavior

We first assess and compare the training and validation errors during the training procedure.

We used an ADAM optimization technique with a learning rate of 10^{-4} and binary cross-entropy loss. The minimum batch size was 32 and 60 epochs were considered. The weight decay was set to 10^{-4} to prevent over-fitting while training the model. The momentum value was set to 0.9. As regards the depth of the used LSTM, we defined 150 hidden units and added a dropout layer of 0.5.

Fig.5 displays the obtained train accuracy and loss curves for the proposed ProgNet (Fig.5[(d)]) as well as the other used configurations: AlexNnet (Fig.5[(a)]), VGG16 (Fig.5[(b)]), VG19 (Fig.5[(c)]). The displayed curves indicate similar convergence performance for the four architectures.

As regards the test set, Fig. 6 displays accuracy and loss curves for all architectures. When visually inspecting Fig. 6[(d)], we see that the loss and accuracy curves get better faster for the ProgNet architecture with a more stable behavior. Training with the ProgNet architecture is therefore faster and more efficient than the other configurations, partly due to the CNN depth.

It is worth noting that the observed peaks in the accuracy and loss curves may be caused by a lack of representatives of the training and test data, and hence a difficulty in learning some fine features. This deserves more investigation to assess the ability of the model to generalize [43, 44]. This is mainly due to the limited number of available time-series of X-ray chest images related to Covid-19.

²<https://github.com/ieee8023/covid-chestxray-dataset/>

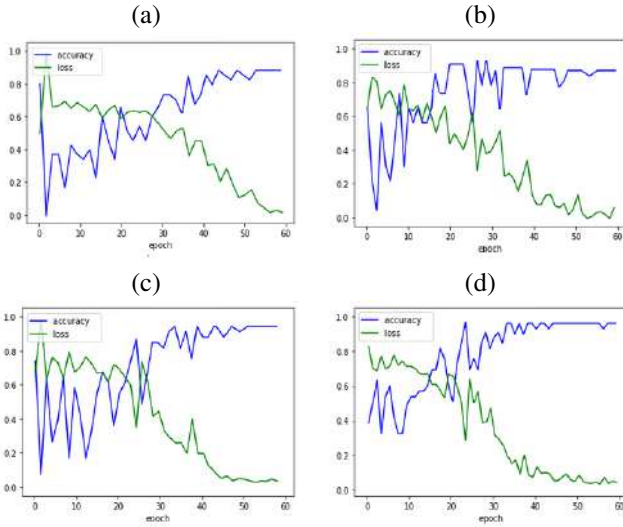


Fig. 5. Accuracy/Loss of the learning train obtained with (a) AlexNet, (b) VGG16, (c) VGG19 and (d) ProgNet.

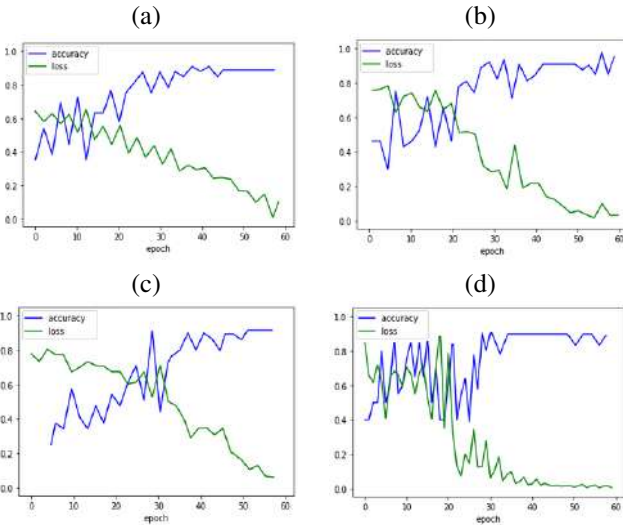


Fig. 6. Accuracy/Loss of the learning test curves obtained with (a) AlexNet, (b) VGG16, (c) VGG19 and (d) ProgNet.

B. Quantitative evaluation

To further assess the performance of the proposed method, Table I reports obtained values for the rates of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN). As reported in Table I, all the networks perform well with TP rates over 87%. However, the proposed ProgNet architecture slightly outperforms the others with the highest TP and TN rates, while the FN and FP rates are also the lowest. By enjoying the lowest FN, the proposed architecture appears very interesting since for such a prognosis, it is crucial to not miss negative evolution cases.

To further assess the quantitative performance, accuracy, precision and recall are also provided in Table II, in addition to the F_1 score [45]. The F_1 score takes into consideration both precision and recall in order to validate the accuracy. It

TABLE I
TP, FN, FP AND TN VALUES FOR THE PROPOSED PROGNET METHOD AND ALEXNET, VGG16, VGG19 ARCHITECTURES.

| | TP | FN | FP | TN |
|----------------|---------------|--------------|--------------|---------------|
| AlexNet | 88.37% | 11.96% | 11.63% | 88.04% |
| VGG16 | 90.14% | 8.79% | 9.86% | 91.21% |
| VGG19 | 91.90% | 7.89% | 8.1% | 92.11% |
| ProNet | 93.81% | 5.99% | 6.19% | 94.01% |

is the harmonic mean of both measures and can be calculated as

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \quad (7)$$

In Table II, the mean values over 10 runs are provided. For each run, a sub-set of training data is randomly chosen. Standard deviations over the 10 runs are also provided in the table.

TABLE II
MEAN ACCURACY, PRECISION, RECALL AND F1SCORE FOR THE PROPOSED PROGNET ARCHITECTURE AS WELL AS ALEXNET, VGG16, VGG19.

| | AlexNet | VGG16 | VGG19 | ProgNet |
|-------------------------------|---------------------|---------------------|---------------------|----------------------------|
| Accuracy | 0.888 ± 0.08 | 0.905 ± 0.03 | 0.923 ± 0.04 | 0.924 ± 0.03 |
| Precision | 0.883 ± 0.05 | 0.90 ± 0.02 | 0.91 ± 0.03 | 0.930 ± 0.02 |
| Recall | 0.880 ± 0.09 | 0.911 ± 0.05 | 0.920 ± 0.06 | 0.939 ± 0.04 |
| F_1 score | 0.881 ± 0.06 | 0.905 ± 0.03 | 0.914 ± 0.04 | 0.934 ± 0.02 |

Through the reported values, one can easily notice that the proposed ProgNet method outperforms the other competing architectures. Specifically, higher precision and recall values indicate that ProgNet is more efficient in retrieving ambiguous infection cases. Moreover, the reported low standard variation values show better stability for the proposed model, which indicates better generalization properties. A visual inspection of ambiguous time-series is provided in Section IV-C.

C. Qualitative Analysis

In this section, we illustrate some representative results obtained on a group of patients. These time series are made up of 3 images acquired at times T_1 , T_2 and T_3 . Fig. 7 displays four time series of Covid-19 patients, two of them (top) have survived, while the two others (bottom) are dead. For each patient, the obtained "survival score" (SC) is provided on the first column. This score is nothing but the probability predicted by the *sigmoid* function.

From a visual point of view, it is clear that time-series with death issue show more white areas over all images. Generally speaking, the spread of these areas increases over time. Reported SC values indicate how sure the ProgNet model is about the classification results for these time-series.

As reported in the previous section, good precision and recall values indicate the ability of our ProgNet architecture to

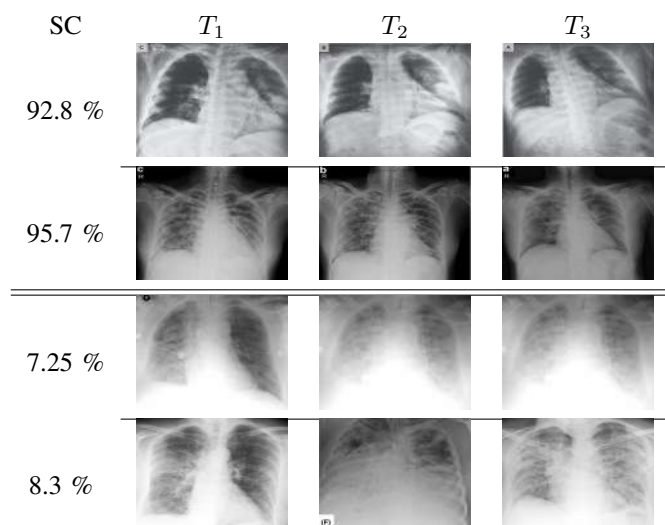


Fig. 7. Visualization of the COVID-19 classification using the proposed ProgNet architecture.

properly classify ambiguous cases. In this sense, Fig. 8 shows two examples of time-series corresponding to two patients with survival and death issues. These patients are well classified by the proposed ProgNet architecture, while the competing methods misclassify them. When visually inspecting the images for the death case, ambiguity comes from extended white areas in all the images with no visible improvement over time. The same remark holds for the survival case where white areas persist over time.

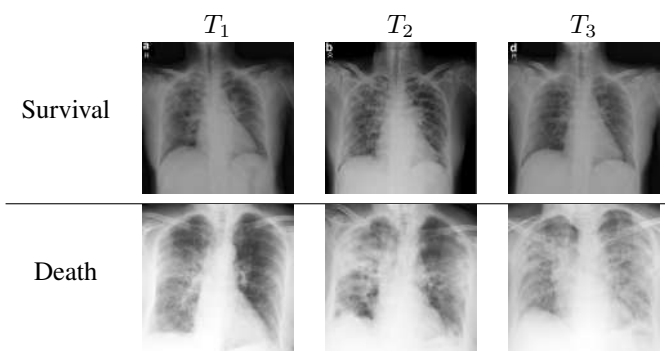


Fig. 8. Examples of time-series well classified by ProgNet for which the competing methods fail.

To investigate why the proposed ProgNet architecture fails to classify some cases, Fig. 9 shows images of a time-series linked to a patient mis-classified by ProgNet. Indeed, we notice in some cases an unexpected change in the situation of a patient during the last image of the sequence. This can lead to some problems in learning all features and managing the time correlation between them. Analysing longer sequences can certainly improve the model performance.

V. CONCLUSION

In this paper, we proposed an architecture for Covid-19 prognosis, based on a combination of a CNN and RNN

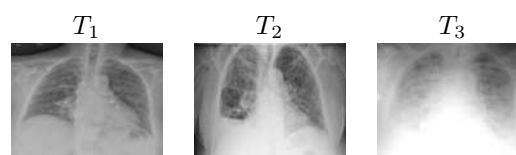


Fig. 9. Images of an ambiguous time-series mis-classified by ProgNet.

networks. The proposed architecture analyses both spatial and temporal dependencies in input time-series of chest X-ray images. Our method is segmentation-free, which makes it competitive with respect to other assessment methods relying on time-consuming lung segmentation algorithms. When applied on radiographic data, the proposed ProgNet architecture showed promising results with good classification performances, especially for ambiguous cases. Specifically, the reported low false positive rates are interesting for an accurate and personalised care workflow.

Future work will focus on applying the proposed architecture to CT data with longer time-series. However, data availability and homogeneity is a key issue. In this sense, the proposed architecture has to be adapted in order to handle data heterogeneity in time (time-series with different sizes).

REFERENCES

- [1] L. Chaari and O. Golubnitschaja, "Covid-19 pandemic by the "real-time" monitoring: the tunisian case and lessons for global epidemics in the context 3pm strategies," *EPMA Journal*, no. 10223, 2020, DOI:10.1007/s13167-020-00207-0.
- [2] C. Huang, Y. Wang, X. Li, L. Ren, J. Zhao, Y. Hu, L. Zhang, G. Fan, J. Xu, X. Gu, C. Zhenshun, Y. Ting, X. Jiaan, W. Yuan, W. Wenjuan, X. Xuelei, Y. Wen, L. Hui, L. Min, X. Yan, G. Hong, G. Li, X. Jungang, W. Guangfa, J. Rongmeng, G. Zhancheng, J. Qi, W. Jianwei, and C. Bin, "Clinical features of patients infected with 2019 novel coronavirus in wuhan, china," *The Lancet*, vol. 395, no. 10223, pp. 497–506, 2020.
- [3] P. Gautret, J.-C. Lagier, P. Parola, H. Van Thuan, L. Meddeb, J. Sevestre, M. Mailhe, B. Doudier, C. Aubry, S. Amrane, P. Seng, M. Hocquart, J. Finance, V. Esteves Vieira, H. Tissot Dupont, S. Honoré, A. Stein, M. Million, P. Colson, B. La Scola, V. Veit, A. Jacquier, J.-C. Deharoand M. Drancourt, P. Edouard Fournier, J.-M. Rolain, P. Brouqui, and D. Raoult, "Clinical and microbiological effect of a combination of hydroxychloroquine and azithromycin in 80 COVID-19 patients with at least a six-day follow up: an observational study," <https://www.mediterranee-infection.com/wp-content/uploads/2020/03/COVID-IHU-2-1.pdf>, pp. 1–28, 2020.
- [4] L. Chih-Cheng, S. Tzu-Ping, K. Wen-Chien, T. Hung-Jen, and H. Po-Ren, "Severe acute respiratory syndrome coronavirus 2 (sars-cov-2) and coronavirus disease-2019 (covid-19): The epidemic and the challenges," *International J. of Antimicrobial Agents*, vol. 55, no. 3, pp. 105924, 2020.
- [5] S. Jo S, S. Kim, D. H. Shin, and M. S. Kim, "Inhibition of sars-cov 3cl protease by flavonoids," *J. Enzyme Inhib. Med. Chem.*, vol. 35, no. 3, pp. 145–151, 2020.
- [6] V. M. Corman, O. Landt, M. Kaiser, R. Molenkamp, A. Meijer, D. K. W. Chu, T. Bleicker, S. Brünink, J. Schneider, M. L. Schmidt, D. G. J. C. Mulders, B. L. Haagmans, B. van der Veer, S. van den Brink, L. Wijsman, G. Goderski, J. L. Romette, J. Ellis, M. Zambon, M. Peiris, H. Goossens, C. Reusken, M. P. G. Koopmans, and C. Drosten, "Detection of 2019 novel coronavirus (2019-nCoV) by real-time RT-PCR," *Euro Surveill*, vol. 25, no. 3, pp. 1–8, 2020.
- [7] X. Xiaowei, J. Xiangao, M. Chunlian, D. Peng, L. Xukun, L. Shuangzhi, Y. Liang, C. Yanfei, S. Junwei, L. Guanqing, L. Yongtao, Z. Hong, X. Kaijin, R. Lingxiang, and W. Wei, "Deep learning system to screen coronavirus disease 2019 pneumonia," <https://arxiv.org/ftp/arxiv/papers/2002/2002.09334.pdf>, 2020.

- [8] F. Shi, J. Wang, J. Shi, Z. Wu, Q. Wang, Z. Tang, K. He, Y. Shi, and D. Shen, "Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for covid-19," *IEEE Reviews in Biomedical Engineering*, pp. 1–1, 2020.
- [9] B. Chandra, M. Gupta, and M. P. Gupta, "A multivariate time series clustering approach for crime trends prediction," in *2008 IEEE International Conference on Systems, Man and Cybernetics*, 2008, pp. 892–896.
- [10] E. Woźniak, W. Kofman, S. Aleksandrowicz, M. Rybicki, and S. Lewiński, "Multi-temporal indices derived from time series of sentinel-1 images as a phenological description of plants growing for crop classification," in *2019 10th International Workshop on the Analysis of Multitemporal Remote Sensing Images (MultiTemp)*, 2019, pp. 1–4.
- [11] Y. Lin, L. Zhang, and N. Wang, "A new time series change detection method for landsat land use and land cover change," in *10th International Workshop on the Analysis of Multitemporal Remote Sensing Images (MultiTemp)*, 2019, pp. 1–4.
- [12] W. Gharbi, L. Chaari, and A. Benazza-Benyahia, "Unsupervised bayesian change detection for remotely sensed images," *Signal, Image and Video Processing*, 2020, accepted.
- [13] R. Chuentawat and Y. Kan-ngan, "The comparison of PM2.5 forecasting methods in the form of multivariate and univariate time series based on support vector machine and genetic algorithm," in *2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2018, pp. 572–575.
- [14] L. Chaari, T. Vincent, F. Forbes, M. Dojat, and P. Ciuciu, "Fast joint detection-estimation of evoked brain activity in event-related fmri using a variational approach," *IEEE Transactions on Medical Imaging*, vol. 32, no. 5, pp. 821–837, May 2013.
- [15] M. Albughdadi, L. Chaari, J. Y. Tourneret, F. Forbes, and P. Ciuciu, "A bayesian non-parametric hidden markov random model for hemodynamic brain parcellation," *Signal processing*, vol. 135, no. 10223, pp. 132–146, 2017.
- [16] B. Bouaziz, L. Chaari, H. Batatia, and A. Quintero-Rincon., "Epileptic seizure detection using a convolutional neural network," in *In International Conference on Digital Health Technologies (ICDHT)*, Sfax, Tunisia, October, 15-16 2018.
- [17] A. Laruelo, L. Chaari, H. Batatia, S. Ken, B. Rowland, J.-Y. Tourneret, and A. Laprie, "Hybrid sparse regularization for magnetic resonance spectroscopy," in *IEEE International Conference of Engineering in Medicine and Biology Society (EMBC)*, Osaka, Japan, July, 3-7 2013, pp. 6768–6771.
- [18] M. Prummer, J. Horneegger, G. Lauritsch, L. Wigstrom, E. Girard-Hughes, and R. Fahrig, "Cardiac c-arm ct: A unified framework for motion estimation and dynamic CT," *IEEE Transactions on Medical Imaging*, vol. 28, no. 11, pp. 1836–1849, 2009.
- [19] L. Zewen, Y. Wenjie, P. Shouheng, and L. Fan, "A survey of convolutional neural networks: Analysis, applications, and prospects," *arXiv preprint arXiv:2004.02806*, 2020.
- [20] T. N. Sainath, M. Abdel-rahman, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 8614–8618.
- [21] K. Alex, S. Ilya, and H. Geoffrey E, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [22] S. Karen and Z. Andrew, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [23] T. Fang, "A novel computer-aided lung cancer detection method based on transfer learning from googlenet and median intensity projections," in *IEEE International Conference on Computer and Communication Engineering Technology (CCET)*, 2018, pp. 286–290.
- [24] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [25] Z. Wu, J. Hai, L. Zhang, J. Chen, G. Cheng, and B. Yan, "Cascaded fully convolutional densenet for automatic kidney segmentation in ultrasound images," in *2nd International Conference on Artificial Intelligence and Big Data (ICAIBD)*, 2019, pp. 384–388.
- [26] P. V. Arun, K. M. Buddhiraaju, and A. Porwal, "Analysis of capsulenets towards hyperspectral classification," in *9th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2018, pp. 1–5.
- [27] S. Ying, Z. Shuangjia, L. Liang, Z. Xiang, Z. Xiaodong, H. Ziwang, C. Jianwen, Z. Huiying, J. Yusheng, W. Ruixuan, et al., "Deep learning enables accurate diagnosis of novel coronavirus (covid-19) with ct images," *medRxiv*, 2020.
- [28] S. Prabira Kumar and B. Santi Kumari, "Detection of coronavirus disease (covid-19) based on deep features," *Preprints*, vol. 2020030300, pp. 2020, 2020.
- [29] W. Linda and W. Alexander, "Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest radiography images," *arXiv preprint arXiv:2003.09871*, 2020.
- [30] A. Asmaa, A. Mohammed M, and G. Mohamed Medhat, "Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network," *arXiv preprint arXiv:2003.13815*, 2020.
- [31] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, "Detrac: Transfer learning of class decomposed medical images in convolutional neural networks," *IEEE Access*, pp. 1–1, 2020.
- [32] M. Rußwurm and M. Körner, "Multi-temporal land cover classification with sequential recurrent encoders," *ISPRS International Journal of Geo-Information*, vol. 7, no. 4, pp. 129, 2018.
- [33] H. Sepp and S. Jürgen, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [34] S. Kamilya and J. Alex Pappachen, "A survey on lstm memristive neural network architectures and applications," *The European Physical Journal Special Topics*, vol. 228, no. 10, pp. 2313–2324, 2019.
- [35] "Lstm," .
- [36] K. Dutta and K. K. Sarma, "Multiple feature extraction for RNN-based assamese speech recognition for speech to text conversion application," in *International Conference on Communications, Devices and Intelligent Systems (CODIS)*, 2012, pp. 600–603.
- [37] W. Kahuttanaseth, A. Dressler, and C. Netramai, "Commanding mobile robot movement based on natural language processing with RNN encoderdecoder," in *5th International Conference on Business and Industrial Research (ICBIR)*, 2018, pp. 161–166.
- [38] Q. Wang, H. Fan, G. Sun, W. Ren, and Y. Tang, "Recurrent generative adversarial network for face completion," *IEEE Transactions on Multimedia*, pp. 1–1, 2020.
- [39] X. Kai and W. Ying, "Lstm-ma: A lstm method with multi-modality and adjacency constraint for brain image segmentation," in *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 240–244.
- [40] "Improvement in land cover and crop classification based on temporal features learning from sentinel-2 data using recurrent-convolutional neural network (r-cnn)," .
- [41] P. Hu, J. Tong, J. Wang, Y. Yang, and L. d. Oliveira Turci, "A hybrid model based on cnn and bi-lstm for urban water demand prediction," in *IEEE Congress on Evolutionary Computation (CEC)*, 2019, pp. 1088–1094.
- [42] Y. Heryadi and H. L. H. S. Warnars, "Learning temporal representation of transaction amount for fraudulent transaction recognition using CNN, stacked LSTM, and CNN-LSTM," in *IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, 2017, pp. 84–89.
- [43] J. Gareth, W. Daniela, H. Trevor, and T. Robert, *An introduction to statistical learning*, vol. 112, Springer, 2013.
- [44] G. Ian, B. Yoshua, and C. Aaron, *Deep learning*. MIT press, 2016.
- [45] Y. Sasak, "The truth of the F-measure," Tech. Rep., School of Computer Science, University of Manchester, 10 2007.