Classification and Detection of Pneumonia in X-Ray Images Using Deep Learning Techniques

Blida Montalico
UANCV
Ingeniería de Sistemas
Juliaca, Peru
alida216@gmail.com

Juan Carlos Herrera
UANCV
Ingeniería de Sistemas
Juliaca, Peru
29606930@uancv.edu.pe

2022 IEEE Sixth Ecuador Technical Chapters Meeting (ETCM) | 978-1-6654-8744-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/ETCM56276.2022.9935757

***Abstract*—Nowadays, artificial intelligence is applied in a great variety of fields. As we know there is an enormous quantity of available data that through artificial intelligence helps to make better and faster decisions furthermore efficiently. This research work is focusing in how to use Deep learning based on convolutional neural networks and propose a classification of pneumonia model through X-ray images. This research work is focusing on how to use deep learning based on convolutional neural networks and proposes a model that can classify pneumonia, through X-ray images. We are using our own model called XrayChestNet_v1 and complex models such as VGG16, ResNeXt50_32x4 y GoogLeNet applying transfer learning technique. In this research, we made two experiments, we tried different architectures in each of them. XrayChestNet_v1, has the task to help to classify and detect whether an X-ray shows changes or signs and classify them in two groups depending on results.**

***Keywords*—Pneumonia, Chest X-rays images, deep learning, convolutional neuronal network (CNN), computer vision**

I. Introduction

In 2019, pneumonia was one of the leading causes of mortality in children under five years of age worldwide. [1]. In 2019, there were 1,870,294 cases of acute respiratory infections and pneumonia in children under five years of age. Respiratory diseases and pneumonia were one of the leading causes of deaths in Peru. [2]

The early detection, adequate number of specialist, and hospitals are the best components to reduce the mortality rate againsting pneumonia. The purpose of this work is to develop a model that classifies and detects pneumonia through analising X-ray images using deep learning techniques. In order to have an early diagnosis. And propose a tool that supports specialists in the pneumonia diagnosis process.

The use of deep learning to image recognition, through layers, which learn their own representation from the input data that performs an optimal classification task. CNN It has become one of the most used methods for image classification and detection tasks. [4] CNNs were one of the first deep networks trained with back-propagation. They were successful unlike other deep networks that suffered from gradient problems due to their computational and statistical defficiencies [7]. We propose a model of 3 convolutional layers and 2 fully connected layers, dropout at 0.2, batch normalization and pooling (max-pooling) techniques were used for better performance. [5] ReLu was used as activation function. The development of a simple model that does not require a great computational capacity, will allow the training and testing process to be faster, easier and accurate.

II. Methodology

A. Pneumonia

Pneumonia is a pulmonary infection caused by the invasion of microorganisms into the distal airway and parenchyma. It generates an inflammation in one or both lungs, causing respiratory complications that can cause serious consequences, including death. [6]

Pneumonia can be classified by the type of contagion such as Community Acquired Pneumonia (CAP), with Streptococcus pneumonia being the main cause of this type of pneumonia. We also have Haemophilus influenza, Staphylococcus aureus, Legionella pneumophila, Mycoplasma pneumonia and many other viruses that generate the cap. Hospital Acquired Pneumonia (HAP) This infection is obtained during a hospital stay, the bacteria that causes this type of pneumonia are commonly resistant to first-line antibiotics, which can complicate treatment. Finally, Ventilator Acquired Pneumonia (VAP) that is acquired when the patient is mechanically ventilated. [6]

Pneumonia is mainly classified depending on the causative agent. One of the first is bronchopneumonia, it starts with an inflammation in the bronchi and followed by a complication in the lungs. On the second hand, we have lobal pneumonia, also known as non-segmental pneumonia, which affects the pulmonary lobes, thus affecting the rest of the pulmonary system. Finally, we have interstitial pneumonia, which causes scarring called pulmonary fibrosis, increasing the likelihood of viruses and germs that common bacteria can produce.

B. Transfer Learning

Transfer learning is popular in deep learning because of the enormous resources required to train deep learning models. Transfer learning only works in deep learning if the model features learned in the first task are general. (Czum, J. M. 2020).

C. Convolutional Neural Network (CNN)

High-dimensional image inputs are common, for example space-time images and multivariate images. When trying to model this type of data, some deep learning models become saturated with excess information. However, convolutional neural networks specialize in these problems, learning local patterns; in the case of images, patterns that are located in 2D windows of the inputs.

CNNs were one of the first deep networks trained with back-propagation. They were successful unlike other deep networks that suffered from gradient problems due to their computational and statistical efficiencies. [7]

D. VGG16 architecture

The VGG16 architecture [8], is a deep convolutional network specialized in classification tasks. The contribution of this architecture is the use of very small kernels (3x3), increasing the depth to 16 or 19 layers. For VGG training it requires RGB images with a fixed size of 224x224, a stride of 1 and a padding of 1, it also applies a max pooling of 2x2.

After the stack of convolutional layers, vgg uses three fully connected layers, the first two layers of 4096 and the third of 1000 neurons, one for each class, the final layer is softmax. The architecture was trained with Imagenet [12]. VGG has various configurations, for this work VGG16 was used, with 134 million parameters.

E. ResneXt50_32x4d architecture

ResneXt50_32x4d was proposed by Saining Xie et al [9], presents a modular architecture for image classification. The architecture is made up of blocks that add transformations with the same topology. It is based on the ResNet architecture, ResneXt's contribution is the use of cardinality, this is a simple and homogeneous design with multiple branches with a single set of hyperparameters. The architecture features 48 convolutional layers grouped into blocks of 3 convolutional layers, a fully connected layer, and a final softmax layer.

F. GoogLeNet Architecture

GoogLeNet [10], is a deep network proposed by Google, the contribution of this model is to increase the convolutional layers while maintaining computational resources. GoogLeNet implements 22 convolutional layers, it is used in image classification tasks. The architecture presents inception modules, implementing 9 of these modules. GoogLeNet is organized in three sections: The convolutional section, the inception modules and the output layer. The convolutional section is made up of two convolutional layers with max-pooling. In the second section, there are three blocks of two, five and two inception modules respectively. The last inception layer produces 1,024 maps with dimensions 7x7. The output layer is made up of an average-pooling layer, a dropout of 0.4, and a final softmax layer. ReLU is used in all convolutional layers. The loss function is cross entropy.

G. XrayChestNet_v1 architecture

The architecture designed and proposed is called XrayChestNet_v1, it is made up of five convolutional layers and 2 fully connected neural layers, 0.2 dropout, batch normalization and pooling techniques are used. ReLU was used as activation function. In the figure 1 XrayChestNet_v1 architecture implemented in Pytorch is shown.

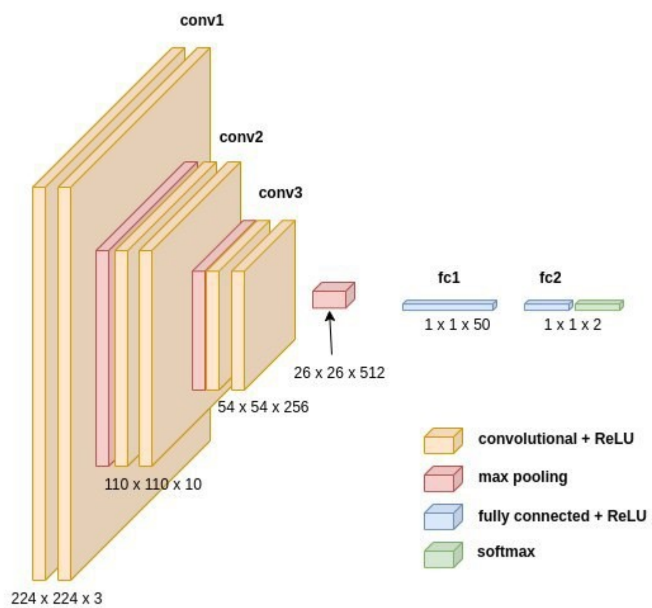

Fig. 1. Model developed in Pytorch called XrayChestNet_v1

H. Dataset

The dataset used for the experiments was publicly collected and labeled by Daniel S Kermany et al. [11]. The dataset consists of 5,232 frontal-view X-ray images, with two categories, images of lungs showing pneumonia (2,538 bacterial and 1,345 viral) and images of normal lungs (1,349). The images correspond to 5826 pediatric patients. An example of the normal lungs and pneumonia lung images is shown in figure 2.

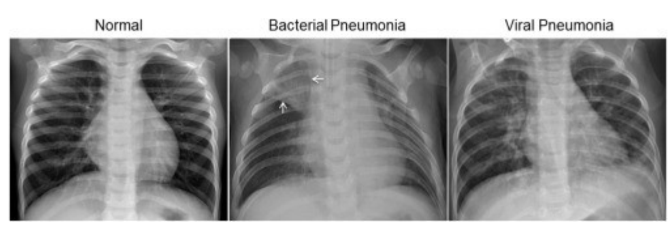

Fig. 2. X-ray images of healthy and pneumonia lungs - Dataset X-Ray Chest

For image processing, the dataset is divided into train, val and test files. This distribution of images was necessary to architectures training and test. The images for training are 5218, for validation they are 18 and for testing they are 626.

III. EXPERIMENTS

Regarding to development and training of the architectures, a Lenovo Intel i5 Laptop with 8Gb RAM and 4 Gb NVIDIA GPU, Windows 11 and Ubuntu 20.10 operating system was used. As software, python3.8, pytorch, numpy, matplotlib libraries were used. The work was developed in Jupyter Lab under Anaconda environments.

A. Experiment 1 - XrayChestNet_v1

On the first experiment we use the proposed architecture, designed for a specific problem to be solved, which is classification of chest X-ray images in pediatric patients. This model is compose of:

1) Convolutional layer 1: This first layer used a stride of 1, the Kernel (filter) with a dimension of 5x5, padding was not used. An image with 3 channels (RGB) and a dimension of 224x224 was used as input. In the convolutional layer, 10 filters were applied. Applying the formula, the dimension at the output of the first convolutional layer was obtained:

$$Dimension = \frac{DimensionIn - Kernel + 2(Padding)}{Stride} + 1 \tag{1}$$

$$Dimension = \frac{224 - 5 + 2(0)}{1} + 1 \tag{2}$$

A 2x2 max-pooling is applied to this output, obtaining a 110x110*10 feature map as output.

2) Convolutional layer 2: In this layer a stride of 1 was used, the Kernel (filter) with a dimension of 3x3 and padding was not implemented. An image with a 110x110x10 feature map was used as input. In this layer 256 filters were applied. The dimension at the output of the second convolutional layer will be:

$$Dimension = \frac{110 - 3 + 2(0)}{1} + 1 \tag{3}$$

Max pooling of 2x2 is applied to this output, obtaining as output a feature map of 54x54x256.

3) Convolutional layer 3: On the last convolutional layer, we used a stride of 1, a Kernel of 3. An image with 256 channels was used as input. 512 filters were applied to the layer. The dimension at the output of the third convolutional layer will be:

$$Dimension = \frac{54 - 3 + 2(0)}{1} + 1 \tag{4}$$

2x2 max-pooling is applied to this output, obtaining a 26x26x512 feature map as output.

4) Dense layer: After applying the 3 convolutional layers, a dimension of 26x26x512 is obtained that will enter the dense layers (fully connected layers). The first neural layer requires a total of 26x26x512 neurons as input and as output it will have 50 neurons. It uses ReLU as the activation function. The output layer has 50 neurons as input and two neurons as output, as it is a binary classification problem (PNEUMONIA/NORMAL). Softmax is applied as output of the dense layer.

For the training of experiment 1, 30 epochs were used, as a Cross Entropy loss function, momentum and a variable learning rate.

Three options were used for the optimizer: stochastic gradient descent (SGD), adaptive momentum estimation (Adam), and adaptive gradient algorithm (AdaGrad). SGD is the classical optimizer based on taking samples for faster calculation of gradient descent. AdaGrad, modifies SGD, by varying the learning rate based on calculations from previous iterations, unlike SGD, the learning rate will adjust automatically. Adam, has characteristics similar to momentum and AdaGrad, it takes the data from previous iterations (gradient averaging) and uses less memory for processing.

The results obtained for the 3 cases in the training and testing of the model are shown in the tableI.

TABLE I
EXPERIMENT 1

	Train Accuracy	Test Accuracy	Training Time
SGD	0.7429	0.6250	62m37s
ADAM	0.7429	0.6250	74m57s
Adagrad	**0.8562**	**0.7404**	71m49s

After training the model, the prediction was made with the test data of the dataset. In the 3 trainings, a better result was obtained in the test data when applying Adagrad with 0.7404 accuracy.

B. Experiment 2 - Transfer Learning

Transfer learning was used for the experiment. This allows us to use pre-trained models in the state of the art. These models require a lot of computational power to train them on datasets such as Imagenet . Our proposal is to use a pre-trained model with the VGG16, Resnext50_32x4 and GoogleNet architectures, which are specialized classification models.

Pytorch has these pre-trained models in its repository, this allows them to be applied to our dataset, with the same training routines as experiment 1.

Our work performs the classification of two categories (PNEUMONIA and NORMAL). The last layer of VGG16, Resnext50_32x4 and GoogleNet must be modified, because they were trained in imagenet with 1000 categories, it must be adapted to our problem with two categories, and perform the training in the last layer, keeping the weights of the previous layers.

For the training of experiment 2, 20 epochs were used, as a Cross Entropy loss function, a variable learning rate, SGD, Adam and AdaGrad were used as optimizer.

1) Training 1 Transfer Learning: The first training used the VGG, Resnext50_32x4 and Googlenet pretraining models, Cross Entropy was used as loss function and SGD as optimizer. After 20 training epochs, the following result was obtained:

TABLE II
RESULTS OF EXPERIMENT 1 WITH SGD

Architecture	Train Accuracy	Test Accuracy	Training Time
VGG	**0.9724**	0.8926	119m50s
Resnext50_32x4d	0.9722	**0.9503**	86m55s
GoogLeNet	0.9563	0.8926	65m15s

From the trainings, a better result was obtained with the test data when applying Resnext50_32x4d and SGD with 0.9503 accuracy.

In the figure 3 shows the images obtained in the test data for Resnext50_32x4. It is observed that only in Resnext50_32x4 it presents an error. Showing the improvement of the results when applying transfer learning on a small network.

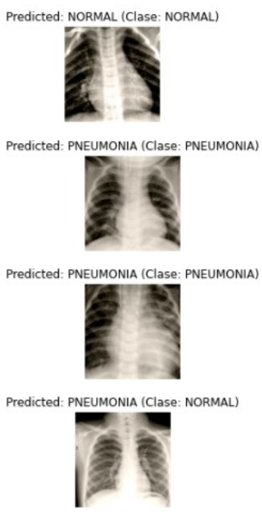

Fig. 3. Resnext50_32x4 model prediction with SGD that obtained the best accuracy

2) Training 2 Transfer Learning: The second training used the VGG, Resnext50_32x4 and Googlenet pretraining models, Cross Entropy was used as loss function and Adam optimizer. After 20 training epochs, the following result was obtained:

TABLE III
RESULTS OF EXPERIMENT 2 WITH ADAM

Architecture	Train Accuracy	Test Accuracy	Training Time
VGG	0.7429	0.6250	130m37s
Resnext50_32x4d	0.9074	0.8125	109m53s
GoogLeNet	**0.9369**	**0.8365**	72m44s

From the trainings, a better result was obtained with the test data when applying Googlenet and Adam with 0.8365 accuracy. One factor to consider when using Adam was the increase in training times, exceeding two hours, but with lower results than training 1 used by VGG.

In the figure 4 shows the images obtained in the test data for GoogLeNet with Adam. It is observed that GoogLeNet has only two correct predictions.

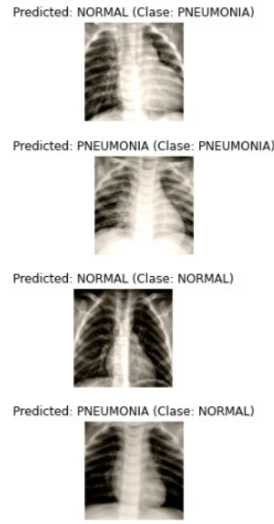

Fig. 4. Googlenet model prediction with Adam that obtained the best accuracy

3) Training 3 Transfer Learning: The training used the VGG, Resnext50_32x4 and Googlenet pretraining models, Cross Entropy was used as loss function and AdaGrad as optimizer. After 20 training periods, the results obtained were:

TABLE IV
RESULTS OF EXPERIMENT 3 WITH ADAGRAD

Architecture	Train Accuracy	Test Accuracy	Training Time
VGG	0.9688	0.8590	119m3s
Resnext50_32x4d	**0.9711**	**0.9551**	92m47s
GoogLeNet	0.9536	0.9375	76m49s

Of the trainings, the best result was obtained with the test data when applying Resnext50_32x4d and AdaGrad with 0.9551 accuracy.

In this training, the times were lower than training 2 and slightly higher than training 1, but the best result was obtained. In the figure 5, the images obtained in the test data for ResNeXt50_32x4 in the third training are shown.

IV. RESULTS

In experiment 1, an architecture based on 3 convolutional layers and 2 neural layers was implemented. In this experiment, three trainings were applied, having the optimizer as a differentiating factor. It was used as optimizers; Stochastic gradient descent (SGD), Adam and Adagrad. From the results obtained, only in the training data, Adagrad has an improvement in the metrics with an accuracy of 0.8562, in the test data also with AdaGrad the best accuracy of 0.7404 was obtained with a time of 71m49s.

Another important point in the results of experiment 1 is the time, and the epochs required to generalize and achieve the best results, while stochastic gradient descent (SGD) took 8 epochs, ADAM took only 4 training epochs, while AdaGrad took it up to epoch 27. This corresponds and is validated with

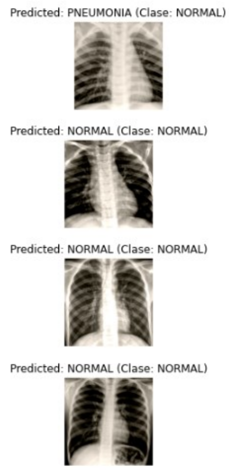

Fig. 5. ResNeXt50_32x4 model prediction with AdaGrad that obtained the best accuracy

the state of the art. In terms of time, being SGD, it has less complexity, therefore, it had the shortest time to complete the training, it took 62m37s to complete the 30 training epochs.

In experiment 2, transfer learning technique was used based on pre-trained models found in the Pytorch repositories. Three specialized classification architectures were selected, VGG16, Resnext50_32x4, and Googlenet. In terms of computational capacity, VGG16 required more resources, despite having fewer convolutional layers, but it has more dense layers compared to Resnext50_32x4 and Googlenet that add parameters to be calculated.

Similar to experiment 1, experiment 2 applied 3 optimizers, SGD, Adam and AdaGrad.

In the three architectures, with the training data VGG16 with SGD had the best result with 0.9754 accuracy, in the test data the best was Resnext50_32x4 with AdaGrad, with an accuracy of 0.9551 and a time of 92 min 47s which is higher compared to experiment 1.

Of the two experiments implemented in this research work, it can be validated that the use of transfer learning had better results in the accuracy metric compared to its own architecture with few layers. On the other hand, regarding the time of the two experiments, the XrayChestNet_v1 own model obtained better results, being 22.8% faster than transfer learning and using less computational capacity. This is validated with the state of the art.

V. CONCLUSIONS

In this research work, preprocessing and data augmentation were applied on dataset images. Deep learning was used to designing convolutional networks and knowledge transfer in chest X-ray image classification tasks in lung diseases, with very good results (accuracy of 0.9551 with Resnext50_32x4 and AdaGrad).

A model called XrayChestNet_v1, specialized in classifying chest X-ray images, was designed and implemented. This model had an accuracy of 0.7404 on acceptable test data for a light architecture, compared to models with 20 or more convolutional layers like Googlenet and Resnext50_32x4.

A convolutional model was implemented to classify and detect lung diseases, obtaining good results. Being an alternative method, suitable for the classification and detection of lung diseases on X-ray images.

REFERENCES

[1] Who.int. 2022. Pneumonia. [online] Available at: ¡https://www.who.int/news-room/fact-sheets/detail/pneumonia¿ [Accessed 6 July 2022].
[2] Perú, D. E. L. (2021). Boletín 2021.
[3] NCI Dictionary of Cancer Terms. [online] Available at: ¡https://www.cancer.gov/publications/dictionaries/cancer-terms/def/pulmonary-disease¿ [Accessed 6 July 2022].
[4] Learning Deep Learning: Theory and Practice of Neural Networks, Computer Vision, NLP, and Transformers using TensorFlow. 752. https://www.oreilly.com/library/view/learning-deep-learning/9780137470198/
[5] Du, C. J., Cheng, Q. (2014). Computer vision. In Food Engineering Series. https://doi.org/10.1007/978-1-4939-0311-57
[6] Lecciones de Anatomía Patologica. [online] Available at: ¡http://publicacionesmedicina.uc.cl/AnatomiaPatologica/Indice.html¿ [Accessed 6 July 2022].
[7] Understanding Deep Learning: Application in Rare Event Prediction. Journal of Chemical Information and Modeling, 1(1), 1–426. https://doi.org/10.13140/RG.2.2.34297.49765
[8] Very Deep Convolutional Networks for Large-Scale Image Recognition. http://arxiv.org/abs/1409.1556
[9] Xie, S., Girshick, R., Dollár, P., Tu, Z., ; He, K. (2016). Aggregated Residual Transformations for Deep Neural Networks. http://arxiv.org/abs/1611.05431
[10] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V.; Rabinovich, A. (2014). Going Deeper with Convolutions. http://arxiv.org/abs/1409.4842
[11] Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C. S., Liang, H., Baxter, S. L., McKeown, A., Yang, G., Wu, X., Yan, F., Dong, J., Prasadha, M. K., Pei, J., Ting, M., Zhu, J., Li, C., Hewett, S., Dong, J., Ziyar, I., … Zhang, K. (2018). Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning., 1122-1131.e9. https://doi.org/10.1016/j.cell.2018.02.010
[12] J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, ”ImageNet: A large-scale hierarchical image database,” 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.