

# AUTOMATIC DIAGNOSIS OF RHEUMATOID ARTHRITIS FROM HAND RADIOGRAPHS USING CONVOLUTIONAL NEURAL NETWORKS

## DIAGNÓSTICO AUTOMÁTICO DE ARTRITIS REUMATOIDE EN RADIOGRAFÍAS DE MANOS UTILIZANDO REDES NEURONALES CONVOLUCIONALES

M. BETANCOURT-HERNÁNDEZ<sup>a†</sup>, G. VIERA-LÓPEZ<sup>b</sup>, A. SERRANO-MUÑOZ<sup>b</sup>

a) Facultad de Telecomunicaciones y Electrónica, Universidad Tecnológica de La Habana, 19390 La Habana, Cuba; marcosraul94@gmail.com<sup>†</sup>

b) Grupo de Sistemas Complejos y Física Estadística, Facultad de Física, Universidad de La Habana, 10400 La Habana, Cuba

<sup>†</sup> corresponding author

Recibido 25/4/2018; Aceptado 15/6/2018

The traditional diagnosis method of Rheumatoid Arthritis (RA) consists in the evaluation of hands and feet radiographs. However, still for medical specialists it turns out to be a complex task because many times the correct diagnosis of the disease depends on the detection of very subtle changes for the human eye. In this work, we developed a system based on Artificial Intelligence (AI), using Convolutional Neural Networks (CNN) for the automatic detection of RA from hand radiographs. The model efficiency is measured with 15 cases achieving an accuracy of 100%. Results of the experiments conducted, showed a superior performance compared to similar state-of-the-art systems reported in the consulted bibliography. This model would be useful for Cuban medicine as a diagnosis tool.

El método tradicional de diagnóstico de la Artritis Reumatoide (RA) consiste en la evaluación de radiografías de manos y pies. Sin embargo, aún para los especialistas médicos resulta una tarea compleja pues, en muchas ocasiones, el correcto diagnóstico de la enfermedad depende de la detección de cambios muy sutiles para el ojo humano. En este trabajo, se desarrolla un sistema basado en inteligencia artificial (AI), empleando Redes Neuronales Convolucionales (CNN) para la detección automática de la RA en radiografías de manos. La eficiencia del modelo es medida con 15 casos obteniendo una precisión del 100%. Los resultados obtenidos por los experimentos realizados muestran un rendimiento superior a los sistemas similares reportados en la bibliografía consultada. Este modelo puede ser de utilidad para la medicina cubana como una herramienta de diagnóstico.

PACS: Medical applications (Aplicaciones médicas), 42.62.BE; neural networks (redes neuronales), 07.05.Mh; image processing (procesamiento de imágenes), 07.05.Pj

### I. INTRODUCTION

Rheumatoid arthritis is a relatively common disease, it is present in approximately 1% of the world population [1]. It is a chronic autoimmune systemic illness, characterized by a persistent inflammation of joints, producing atrophy and bone rarefaction that evolves through very painful outbreaks and typically affects the small joints, producing their progressive destruction and generating different degrees of deformity and functional disability [2]. Figure 1 shows a radiograph of a patient with an advanced stage of the disease.

Radiological examination is the fundamental reference method for the detection of RA. Methods such as Larsen [3] and Sharp [4] have been used for diagnosis, however, they are time-consuming processes, since they require the revision of all the joints of the affected areas and errors of appreciation are likely to occur.

Currently in Cuba, there are around 100 rheumatologists for 11 million inhabitants, distributed in such a way that in some provinces there may be up to one or two specialists, an insufficient number to cover the needs of the Cuban population. Therefore, the process of detecting this type of disease is extremely slow, forcing the patients to travel to very distant centers to receive an accurate diagnosis.

Generally, these patients are people with limitations and physical disabilities produced by the disease itself, and under these conditions, they have to go through long waiting lists, causing the progression of the illness and the exacerbation of symptoms.



Figure 1. Radiograph example of a patient with RA used in this study.

Advances in AI, especially CNN have opened new possibilities in the field of medicine, developing systems for detecting dermatological diseases [5], lung pathologies [6], fractures and bone damage [7], among others, with the aim of minimizing the margin of error in medical diagnosis and facilitating an early detection of illnesses.

## II. RELATED WORKS

In Langs *et al.* [8] a method for an automatic quantification of joint space narrowing and erosions in RA was developed. It can be used to detect the disease progression using an active shape model and Adaboost classifier.

In the study conducted by Murakami *et al.* [9], the diagnosis of Periarticular Osteoporosis (one of the symptoms of RA) was attempted. The system determines density characteristics of hand X-rays using histogram analysis, co-occurrence matrices, Fourier transformations and extraction of line components.

Chokkalingam & Komathy [10] proposed an automatic diagnosis system of RA from hand radiographs, using several digital image-processing algorithms for feature extraction and a neural network for its classification. Twenty-three images were used for training the model but no validation tests were performed, therefore its accuracy is unknown.

Murakami *et al.* [11] implemented a system for diagnosing RA, based on the detection of bone erosion. In this study, a segmentation algorithm is used to extract the area of the phalanges and a CNN for detecting the presence of the pathology. For training the model, 129 radiograph images were used. The validation was performed with 30 cases with RA, obtaining a true-positive rate of 80.5% and a false-positive rate of 0.84%, however the false-positive number of the segmentation algorithm was 3.3 per case.

Current implementations of systems for detecting RA from hand radiographs present limited results, with high error rates and low generalization. Their reliability must improve in order to be used as a diagnosis tool in the medical environment, which has no tolerance to errors due to the inherent importance of this task.

## III. METHODOLOGY

For the development of this work, we selected several CNN architectures and trained them end-to-end with hand radiograph images using only their pixel information. With these experiments we seek to evaluate the capacity of the models to learn RA visual features in a similar way a rheumatologist uses in a diagnosis.

### III.1. Image dataset

The images used in this study came from the department of Rheumatology in Hermanos Ameijeiras Hospital, located in Havana, Cuba. They consist in 92 gray-scale digital radiographs of both hands, used by medical specialist in their diagnosis of RA. In Table 1 the characteristics of this dataset are displayed.

#### III.1.1. Ethics

All medical images are anonymous, it is not possible to relate the patient's name to the radiograph, and they were only

used for the development of this work, following all rules of confidentiality.

Table 1. Image database characteristics.

Number of cases	92
With RA	37
Without RA	55
Image resolution	4280x3520 pixels
Image format	DICOM
Color depth	12 bits

### III.2. Software tools

For the implementation of neural networks, the open source library Keras was used. It is available in Python, and uses Google's Tensorflow as its backend. We chose it for its simplicity, speed and control during the design of complex networks. OpenCV and Numpy modules were used to handle and process the images.

### III.3. Preprocessing

In the first stage of data processing, the pixel matrix of the image was extracted, discarding all metadata information attached to the DICOM file header such as patient's name, sex, age, etc; ensuring the anonymity of each case in this research. Then, the resolution of the image was decreased from 4280x3520 to 256x204 pixels, keeping only 0.3466% of the original data. Finally, the values of the pixels were normalized in a range between 0 and 1.

#### III.3.1. Artificial augmentation of the dataset

Training a model with small amounts of data is a challenging task. To face this problem the machine learning community has been using different procedures to artificially extend datasets.

Transforming the original radiographs with a random combination of rotation, zooming, stretching and flipping allow us to improve the classification results of the CNN architectures due to the increment of the training examples. It is important to empathize that the transformations applied do not affect the visual features in small regions of the original images, and instead they allow the networks to learn these features in different angles and sizes.

The models were trained with different copies in each epoch. Then, for example, after 1000 epochs each model have learned from 77 000 different instances. In Figure 2 some examples of transformed images are shown.

#### III.3.2. Dataset division

The dataset was divided in two, one for system training and other to be used as a reference point for validation. The images selected for validation were chosen randomly,

obtaining 5 images of patients with RA and 10 without RA. This division appears in Table 2.

Table 2. Image database characteristics.

Cases	Training	Validation
With RA	32	5
Without RA	45	10
Total	77	15

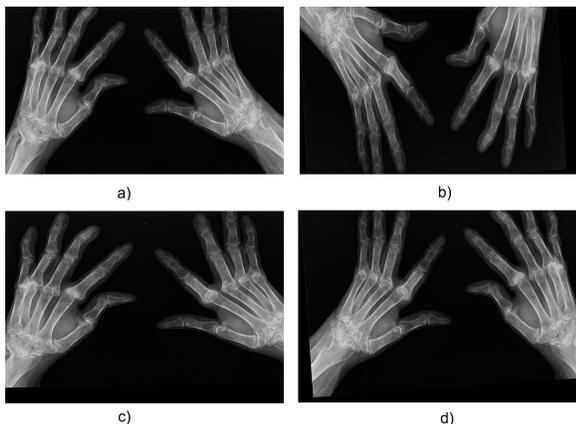


Figure 2. Examples of generated images with transformations. Image (a) is the original. (b) was obtained with horizontal flipping and height distortion, zooming and width distortion was used to produce (c) and (d) was generated with vertical flipping, zooming and rotation.

### III.4. Architecture selection

Different CNN models were initially considered like InceptionV3 [12] and VGG16 [13], which obtained notable results in the ILSVRC competitions, but presented a high degree of overfitting in all tests performed, mainly due to the large number of parameters they had in relation to the dataset size used for their training. As a result of the above, architectures with a smaller number of parameters and significantly less operations were chosen.

#### III.4.1. LeNet

The first architecture implemented was a modified version of LeNet [14]. To achieve better results, changes were made in relation to the original paper, such as increasing the number of feature maps, using 3x3 kernels instead of 5x5, modifying the activation functions to ReLU, changing the output function to Softmax and adding more neurons in full connection layers. We used dropout after max pooling layers and before output layer as a regularization technique. All model specifications can be found in Table 3.

#### III.4.2. Network in Network

A minimalist variant of Network in Network [15] was used, in order to reduce the computational cost of the original architecture. Max pooling layers were added and fewer convolutional filters and layers were implemented. Dropout

was used after max pooling layers. The proposed model is described in Table 4.

Table 3. Modified version of LeNet architecture

Layer	Parameters
Input	image shape = 256x204 pixels
Convolution #1	kernel size = (3,3), strides = (2,2), filters = 64, activation function = ReLU
Max pooling	pool size = (3,3), strides = (2,2)
Dropout	probability = 0.1
Convolution #2	kernel size = (3,3), strides = (1,1), filters = 128, activation function = ReLU
Max pooling	pool size = (3,3), strides = (2,2)
Dropout	probability = 0.1
Full Connection #1	number of neurons = 256, activation function = ReLU
Full Connection #2	number of neurons = 256, activation function = ReLU
Dropout	probability = 0.5
Full Connection #3	number of neurons = 2, activation function = Softmax

#### III.4.3. SqueezeNet

The last architecture implemented was SqueezeNet [16], no changes were made in relation to the original paper except adding dropout after max pooling layers. These model characteristics are displayed in Tables 5 and 6.

Table 4. Reduced version of Network in Network.

Layer	Parameters
Input	image shape = 256x204 pixels
Convolution #1	kernel size = (5,5); strides = (1,1), filters = 32; activation function = ReLU
Max pooling	pool size = (2,2); strides = (2,2)
Dropout	probability = 0.1
Convolution #2	kernel size = (3,3); strides = (1,1), filters = 32; activation function = ReLU
Max pooling	pool size = (2,2); strides = (2,2)
Dropout	probability = 0.1
Convolution #3	kernel size = (3,3) strides = (1,1); filters = 32; activation function = ReLU
Max pooling	pool size = (2,2); strides = (2,2)
Dropout	probability = 0.1
Convolution #4	kernel size = (3,3); strides = (1,1), filters = 64; activation function = ReLU
Convolution #5	kernel size = (3,3); strides = (1,1), filters = 64; activation function = ReLU
Convolution #6	kernel size = (3,3); strides = (1,1), filters = 64; activation function = ReLU
Max pooling	pool size = (2,2); strides = (2,2)
Dropout	probability = 0.5
Convolution #7	kernel size = (1,1); strides = (1,1), filters = 2; activation function = ReLU
Global Average Pooling	activation function = Softmax

### III.5. Hyperparameters

Due to the complexity and high dimensionality of the system, some parameters were selected by default to present this

preliminary results. Default parameters can be seen in the original papers cited on each case.

Table 5. SqueezeNet architecture with dropout.

Layer	Parameters
Input	image shape = 256x204 pixels
Convolution #1	kernel size = (3,3); strides = (2,2) filters = 64; activation function = ReLU
Max pooling	pool size = (3,3); strides = (2,2)
Dropout	probability = 0.1
Fire module #1	squeeze = 16; expand = 64
Fire module #2	squeeze = 16; expand = 64
Max pooling	pool size = (3,3); strides = (2,2)
Dropout	probability = 0.1
Fire module #3	squeeze = 32; expand = 128
Fire module #4	squeeze = 32; expand = 128
Max pooling	pool size = (3,3); strides = (2,2)
Dropout	probability = 0.1
Fire module #5	squeeze = 48; expand = 192
Fire module #6	squeeze = 48; expand = 192
Fire module #7	squeeze = 48; expand = 256
Fire module #8	squeeze = 48; expand = 256
Convolution #2	kernel size = (1,1); strides = (1,1) filters = 2; activation function = ReLU
Global Average Pooling	activation function = Softmax

### III.5.1. Loss function

Being a classification problem of two classes, binary cross entropy loss function was used (1)

$$E = y \log(p) + (1 - y) \log(1 - p), \quad (1)$$

where  $y$  is the correct output (0 or 1) and  $p$  is the predicted probability.

Table 6. Fire module.

Layer	Specifications
Convolution #1	kernel size = (1,1), strides = (1,1), filters = squeeze, activation function = ReLU
Convolution #2	kernel size = (1,1), strides = (1,1), filters = expand, activation function = ReLU
Convolution #3	kernel size = (3,3), strides = (1,1), filters = expand, activation function = ReLU
Concatenation	Convolution #2 & Convolution #3

### III.5.2. Optimization algorithm

For the selection of the optimization algorithm SGD, Adadelta [17], Adam [18] and Adamax [18] were considered, comparing their results after 100 epochs of training. These tests were repeated 3 times with all models (averaging their outputs) and the best results are shown in Figure 3. For Adam and Adamax the learning rate was set at 0.001 and 0.1 for SGD and Adadelta. Despite using SGD and Adadelta a learning rate 100 times higher, still presented a lower performance. Adam was chosen for having the fastest convergence result.

### III.5.3. Learning rate

A variable learning rate was used, starting with 0.001 (default parameter for Adam), then it is reduced with a step function. This process is detailed in Figure 4.

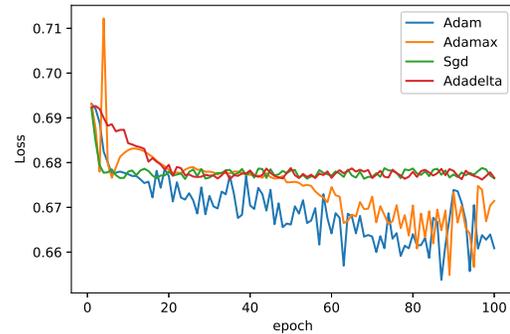


Figure 3. Learning rate decay is visualized over training epochs. A step function decay is used and the learning rate is divided by 2 after 400 epochs of training, then by 5 after 700 and by 10 after 900. Beyond 1000 epochs, the learning rate was fixed at 0.00001.

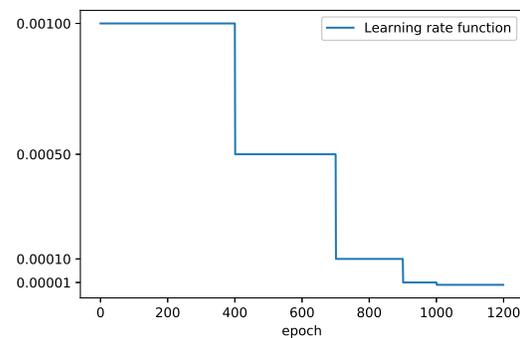


Figure 4. Learning rate decay is visualized over training epochs. A step function decay is used and the learning rate is divided by 2 after 400 epochs of training, then by 5 after 700 and by 10 after 900. Beyond 1000 epochs, the learning rate was fixed at 0.00001.

## IV. RESULTS

All models were trained 15 times with the same dataset, in Figure 5 best five training iterations are shown. Table 7 displays best validation results, including the number of the epoch in which they were obtained and loss error, beyond that point the models began to show signs of overfitting. LeNet and Network in Network reached their maximum point for a accuracy of 93% and SqueezeNet, correctly classified all images scoring 100%.

## V. DISCUSSION AND CONCLUSIONS

In this study we trained and evaluated several CNN architectures with our own Cuban dataset. We propose a system that detects RA from hand radiographs without extensive preprocessing or handcrafted features, only using raw pixel values and achieving better accuracies than similar

models of the state of the art. These preliminary results show the potential of this system, and in future works we will seek to extend the dataset in order to present a reliable diagnosis tool for Cuban medicine.

Table 7. Top validation results.

Model	Accuracy	Loss	Epoch number
LeNet	93 %	0.50	891
Network in Network	93 %	0.22	1248
SqueezeNet	100 %	0.088	1761

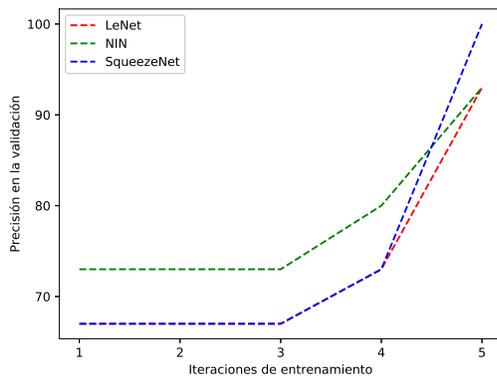


Figure 5. Best five training iterations of LeNet, Network In Network (NIN) and SqueezeNet, showing their validation accuracy.

## VI. ACKNOWLEDGEMENTS

The authors would like to express their gratitude to Dr. Arbelio Pentón Madrigal for supporting this research and the Department of Rheumatology in Hermanos Ameijeiras Hospital, specially to Dr. Chico and Dr. Selva for supporting this research.

## REFERENCES

[1] D. S. C. Mathers and B. Pflieger, *Criterion 1* (2006).  
 [2] D. Aletaha, T. Neogi, A. J. Silman, J. Funovits, D. T. Felson, C. O. Bingham, N. S. Birnbaum, G. R. Burmester,

V. P. Bykerk, M. D. Cohen et al., *Arthritis Rheum.* **62**, 2569 (2010).  
 [3] A. Larsen, K. Dale and M. Eek, *Acta Radiologica. Diagnosis* **18**, 481 (1977).  
 [4] J. T. Sharp, G. B. Bluhm, A. Brook, A. C. Brower, M. Corbett, J. L. Decker, H. K. Genant, J. P. Gofton, N. Goodman, A. Larsen et al., *Arthritis Rheum.* **28**, 16 (1985).  
 [5] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau and S. Thrun, *Nature* **542**, 115 (2017).  
 [6] M. Anthimopoulos, S. Christodoulidis, L. Ebner, A. Christe and S. Mougiakakou, *IEEE Trans. Med. Imaging* **35**, 1207 (2016).  
 [7] J. Olczak, N. Fahlberg, A. Maki, A. S. Razavian, A. Jilert, A. Stark, O. Sköldenberg and M. Gordon, *Acta Orthop.* **88**, 581 (2017).  
 [8] G. Langs, P. Peloschek, H. Bischof and F. Kainberger, *IEEE Trans. Med. Imaging* **28**, 151 (2009).  
 [9] S. Murakami, H. Kim, J. Tan, S. Ishikawa and T. Aoki, *Technological Advancements in Biomedicine for Healthcare Applications*, (Medical Information Science Reference, 2013).  
 [10] S. Chokkalingam and K. Komathy, *World Academy of Science, Engineering and Technology International Journal of Computer Information Systems and Control Engineering* **8**, 834 (2014).  
 [11] S. Murakami, K. Hatano, J. Tan, H. Kim and T. Aoki, *Multimed. Tools Appl.* (2017).  
 [12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, *Proc. IEEE*, 2818 (2016).  
 [13] K. Simonyan and A. Zisserman, *arXiv preprint arXiv:1409.1556* (2014).  
 [14] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, *Proc. IEEE* **86**, 2278 (1998).  
 [15] M. Lin, Q. Chen and S. Yan, *arXiv preprint arXiv:1312.4400* (2013).  
 [16] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally and K. Keutzer, *arXiv preprint arXiv:1602.07360* (2016).  
 [17] M. D. Zeiler, *arXiv preprint arXiv:1212.5701* (2012).  
 [18] D. P. Kingma and J. Ba, *arXiv preprint arXiv:1412.6980* (2014).

This work is licensed under the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0, <http://creativecommons.org/licenses/by-nc/4.0>) license.

