Design of a tool for the classification of skin cancer images using Deep Neural Networks (DNN)

(Diseño de una herramienta para la clasificación de imágenes de cáncer de piel utilizando Redes Neuronales Profundas (DNN))

Diana Paola Merchán Vargas,[1] Helis Navarro Báez,[2] Jaime Guillermo Barrero Pérez[3] y Jeyson Arley Castillo Bohórquez[4]

Campo temático: Machine Learning.

Resumen

El cáncer de piel es una de las enfermedades más comunes en la población mundial. Habitualmente, el diagnóstico requiere la adquisición de imágenes dermatoscópicas. Tanto la biopsia como la histopatología se han utilizado en estadios avanzados. Su detección precoz es muy importante para aumentar la calidad y la esperanza de vida del paciente. En Colombia, la falta de profesionales calificados e instrumental médico dificulta esta tarea. La clasificación automática es un gran desafío, debido a la amplia variedad y morfología de las lesiones cutáneas. Hoy en día, Deep Learning alcanza niveles elevados de precisión en las tareas de clasificación de imágenes y está destinado a convertirse en una solución confiable para la clasificación de imágenes médicas. En esta investigación, utilizó estas ventajas de DNN para construir una red neuronal convolucional (CNN) entrenada con bases de datos de código abierto para la clasificación de lesiones cutáneas benignas y malignas. Después del proceso de entrenamiento, desarrollamos un sistema embebido con raspberry Pi 3 B + con una cámara genérica e implementamos la CNN descrita en Python basada en codificación. Para la clasificación benigna y maligna, el prototipo alcanzó un nivel de precisión del 91,06% en la puntuación F1 y una sensibilidad del 91,98%.

Palabras Clave: cáncer de piel, redes neuronales profundas (DNN), dermatólogos.

[1] Universidad Industrial de Santander. ingdianamerchan@gmail.com
[2] Universidad Industrial de Santander. helis.navarro@correo.uis.edu.co
[3] Universidad Industrial de Santander. jbarrero@uis.edu.co
[4] Universidad Industrial de Santander. jeison.castillo@correo.uis.edu.co

Abstract

Skin cancer is one of the most common diseases in the world population. Usually, the diagnosis requires the acquisition of dermatoscopic images. Both biopsy and histopathology have been used in advanced stages. Its early detection is very important to increase patient life quality and life expectancy. In Colombia, the lack of qualified professionals and medical instruments difficulties this task. The automatic classification is a huge challenge, due to ample variety and morphology in skin lesions. Nowadays, Deep Learning reaches elevated accuracy levels in image classification tasks and is set to become a reliable solution for medical image classification. In this research, used these DNN advantages to build a convolutional neural network (CNN) trained with open source databases to the classification of skin lesions benign and malignant. After the training process, we develop an embedded system with raspberry Pi 3 B+ with a generic camera and implemented the CNN described in Python coded-based. For the benign and malignant classification, the prototype reached an accuracy level of 91.06% in the F1 score and a recall of 91.98%.

Key Words: Skin cancer, Deep Neural Networks (DNN), Dermatologists.

1. Introduction

Skin cancer is one of the most common dermal pathology and in some countries, it ranks first in frequency (Gamerosa & Tellez, 2016). Its incidence has increased in recent decades due to climatic factors and high exposure to solar radiation UV. In Colombia the incidence of cancer is even higher than in other countries because geographically is located near the equatorial parallel; it means that the indices of exposure to ultraviolet solar radiation are among the highest in the world. In addition to this, the population is poorly informed about how to protect itself from UV radiation, which further increases the risk factors and incidence of the disease (Instituto de Hidrología, Meteorología y Estudios Ambientales, 2014).

The purpose of this research is to provide an introduction to the approaches of AI in medical applications through the development of an embedded system with a CNN model for the detection of the benign and malignant classification. For this goal, we use public databases such as ISIC and HAM10000 to obtain the skin lesion images datasets. The result of this project it's a embedded system with two models implemented on-board. For the model description we use Python and TensorFlow backend.

The final goal is to build a prototype device that has an acceptable level of precision to establish a precedent for the development of a useful tool for the diagnosis of some types of cancer and that serves dermatologists as support in their procedures.

2. Data Base

We worked compiling two databases, regarding the problem of the benign and malign classification. To do this, we used two public databases and merge them, these had already classified the training and test data. The Databases used are:

- ISIC (The International Skin Imaging Collaboration): This set consists of 2357 images of 226x226 px of malignant and benign oncological diseases (Kaggle, s.f.).
- HAM10000 ("Human Against Machine with 10000 training images") dataset: The dataset consists of 10015 dermatoscopic images for academic machine learning purposes and is publicly available through the ISIC archive (Tschandl, 2018). Table 1 show the distribution of the final new datasets.

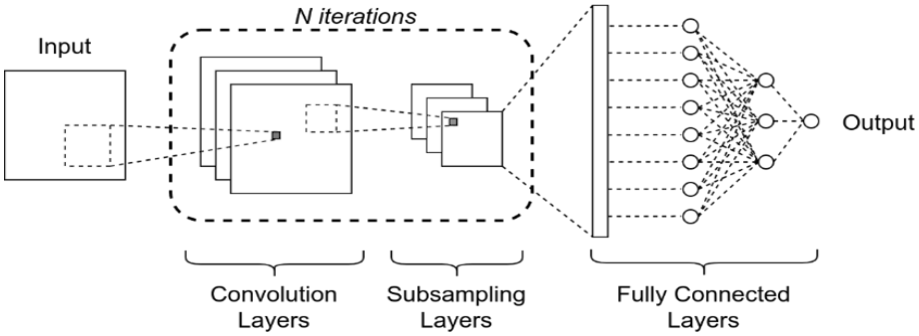
Table 1. Distribution of datasets for Malignant and Benign.

Class	Train val	Test
benign	2200	264
malignant	2200	263

3. Convolutional Neural Networks (CNN)

There are different architectures in the DNN's field, each one with specific features and capabilities. One of the most commonly used architectures for image classification is CNN. Nowadays, the CNN's have achieved impressive performance on many computer vision related tasks, such as object detection and image recognition (Qin, Yu, Liu, & Chen, 2018) , breast cancer (Chanampe, y otros, 2019) and classification of 1D signals for arrhythmia's detection (Castillo, Granados, & Fajardo, 2020). In Figure 1, shows the basic structure of a CNN.

Figure 1. Basic configuration in a CNN



Source: Self made.

4. Transfer Learning and Training Strategies

A. Transfer learning

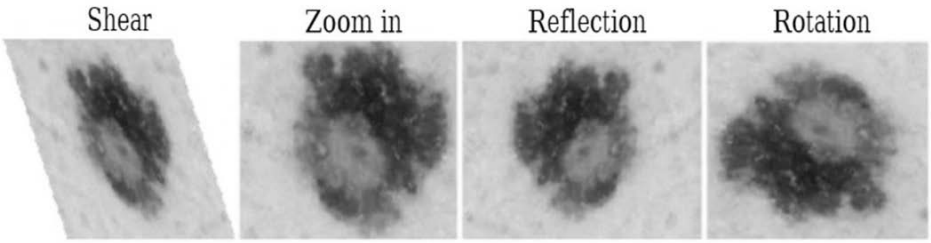
DNN's are generally trained under a supervised learning framework where a model learns a single task using labeled data (Mao, 2020). Usually, a network created from scratch requires full training of its parameters and filter. Likewise, the training process will require time depending on both the depth of the network and the number of parameters. One of the disadvantages of this method is that the filters learn particular features from the examples; it increases the risk of overfitting. Transfer learning appears as a useful tool for solving many challenging problems with the development of DNN's by sharing general features from extensive databases with many classes (Shin, y otros, 2016).

B. Data augmentation

Data augmentation encompasses a set of techniques that improve the size and quality of training data sets, so that better deep learning models can be built with them. Unfortunately, many application domains do not have access to big data, such as medical image analysis (Shorten & Khoshgoftaar, 2019).

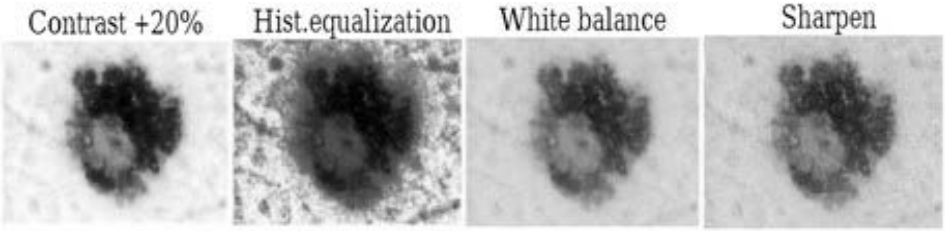
In Figure 2 and Figure 3, is a clear example of Data augmentation in which different types of transformations provided by Mikolajczyk and Grochowski in the domain of melanoma classification are applied to the same image (Mikołajczyk & Grochowski, 2018).

Figure 2. Basic configuration in a CNN



Source: Self made

Figure 3. Same image after different types of color transformations



Source: Adapted from (Mikołajczyk & Grochowski, 2018)

C. Learning Rate

For the adjustment of the hyperparameters, different tests and evaluation of the epoch, the batch size and the learning rate will be carried out, in such a way that fixed values considered optimal for subsequent training with the selected networks will be established.

One strategy to improve the losses adjusts the learning rate when the optimizer

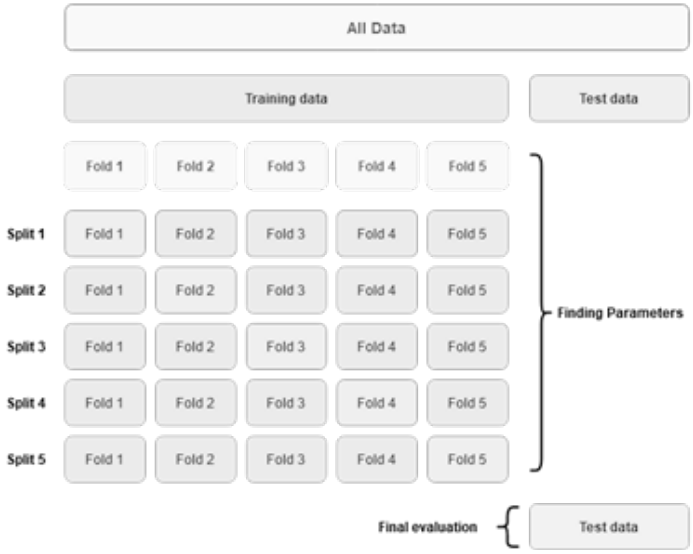
cannot improve the results over a number of epochs. This strategy helps the optimizer when reaching some plateau, to move down the error surface towards the new minima, by modifying the learning rate (Ng, Besounda Mourris, & Karantoforoosh, 2021). The number of epochs to wait is called "patience".

D. K-fold validation

Once we are done with training our model, we simply cannot assume that it will perform well with data that you have not seen before, it means that we cannot be sure that the model has the desired precision and variation in the production environment (Sanjay, 2018). The process of deciding whether numerical results that quantify hypothetical relationships between variables are acceptable as descriptions of the data are known as validation.

Figure 4 shows an example of the performance measure by k-fold cross-validation.

Figure 4. The performance measure by k-fold cross-validation



Source: Self made

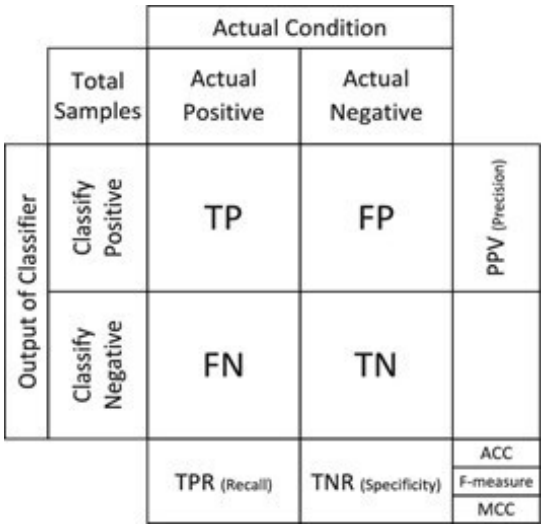
E. F1 score

Nowadays, the researchers think that the most reasonable performance metric is the relationship between the number of samples classified correctly and the total number of samples (Chicco, 2020), which we know as precision that works correctly when the labels are more than two classes.

However, when the data set is unbalanced, accuracy can no longer be considered a reliable measure because it provides an overly optimistic estimate of the capacity of the classifier in the majority class (Sokolova, Japkowicz, & Szpakowicz, 2006).

The F1 score is used to combine the accuracy and recall measurements into a single value. This is practical because it makes it easier to compare the combined performance of accuracy and recall between various solutions (Chicco, 2020). F1 score is calculated by taking the harmonic mean between accuracy and recall (Powers, 2010). First extract the confusion matrix from the predicted targets compared with the real targets (Xia, Zhang, & Li, 2015).

Figure 5. The confusion matrix and relevant evaluation index



Source: Adapted from (Xia, Zhang, & Li, 2015)

Figure 5 show the configuration for a confusion matrix for binary classification. Notice that the multiclass confusion matrix methodology is not different. From the confusion matrix we can obtain the following type 1 error measurements:

- True Positive (TP): The number of residues classified as interacting correctly.
- False Positive (FP): The number of residues classified as interacting correctly incorrectly.
- False Negative (FN): The number of residues classified as non-interacting incorrectly.
- True Negative (TN): The number of residues classified as non-interacting correctly.

5. Selecting the architecture

The criteria used for the selection of the architecture are the following:

- Less complexity in terms of the number of layers and number of parameters.
- Small size (taking into account the capacity of the selected microprocessor).
- An architecture pretrained for image processing and classification.

Taking into account the previous indicators, we have selected the CNN as the optimal architecture, such as the most used architectures for image classification. In order to apply the transfer learning process, we take into account the open access repository of pretrained models provided by Keras (Team, s.f.). The selected models were loaded, trained and tested with the following parameters:

- epochs = 20
- learning rate = 0.001
- batch size = 64
- Optimizer = Adam
- Dataset = Malignant Vs benign

Table 2 Summarizes the results obtained in the default train process. Note that the models with less parameters obtained better results. It is due the application of residual layer into the architecture description. Compared to sequential CNN architectures, the Residual CNN's shows better generalization, meaning the features can be utilized in transfer learning with better efficiency (Glorot & Bengio, 2010).

Table 2. Train process for the select pretrained models, using the malignant and benign Database

First Stage			
Network	**Time/epochs [s]**	**Params**	**Accuracy**
MobileNet V1	392	3.23M	0.9034
MobileNet V2	430	2.38M	0.8390
VGG 19	4929	20.07M	0.5644
VGG 16	3004	14.71M	0.5492
Inception V3	810	21.91M	0.8371
Resnet50	1177	23.79M	0.4621

Considering the results of the Table 2 MobileNet V1 and Inception V3 have been selected as the models to be developed because:

- They are computationally small.
- They are the most used in the state of the art (Morocho Jiménez, 2019).
- The training time is less than others.

6. Training Methodology

For the training process in the two selected models, we developed a factorial experimentation methodology. By applying scaled strategies in training, we can observe the improvement of each one. Therefore, we can determine an appropriate training process. Table 3 shows the different version (Diana, 2021) to apply in the development of training. Note that each experiment is developing with the better parameters and features in the previous version.

Table 3. Description of each of the versions implemented

Version	Description
v1	Training the pretrained model - no freeze
v2	Training the pretrained model - freezing all layers
v2b	Training the pretrained model - freezing of 20 layers
v3	Add Image Data Generation, Add 1 hidden layer layer - 1024 neurons For Regularization: Early Stopping and Model Checkpoint)
v4	For Validation use k fold validation strategy
v5	For Regularization use Reduce Learning Rate task

7. Implementation

For the implementation of the embedded system, we use the best results obtained in the experiment. The embedded use the Tensorflow Lite version, a specific version for portable devices that compress the models in size and memory. Therefore, after saving the models in the Keras format (.h5) is necessary to transform the file into .tflite format. The Figure 6 shows a portion of code required for the conversion .h5 format to .tflite format.

Figure 6. Convert .h5 format to .tflite format

```
import tensorflow as tf
from tensorflow import lite
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dense, Conv2D, Flatten, Dropout, MaxPooling2D
from tensorflow.python.keras.preprocessing.image import ImageDataGenerator
from tensorflow.python.keras.models import load_model
from tensorflow.python.keras import layers
import numpy as np
import matplotlib.pyplot as plt



model = load_model("modelo_inceptionv3_ampliada_imagedatagenerator.h5", compile=False)
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()
open("modelo_inceptionv3_ampliada_imagedatagenerator.tflite", "wb").write(tflite_model)
```

Source: Self made

The device selected for the implementation, is the raspberry pi 3 model B+ (Static Raspberrypi, s.f.). For the image acquisition the device use a USB type raspberry webcam camera.

After loading the models and running them on the card, they had the following response times, the memory occupied by each model is summarized the Table 4.

Table 4. Computational consumption of the model.

Network	Memory [MB]
MobileNet V1	17
Inception V3	95.5

Taking into account the results obtained in the Table 4, we finally worked with the MobileNet V1 model since it occupies less computational space and the execution time is less. To visualize the results, a local web page was developed Using Flask which is a "micro" framework that allows us to create web applications with Python in a very simple way, it can be very convenient for certain applications that do not need many extensions. This does not need an infrastructure with a web server to test the applications but in a simple way, you can run a web server to see the results that are obtained. The card was configured as a server in which there is a 5000 port where the camera can be observed in real-time, there the photo can be taken and evaluated. To obtain these results it can be executed from a PC, phone or tablet.

8. Results

In the first stage, we made the training process for the two selected models. Table 5 shows the training results for version v1, v2, v2b. Note that version v2 obtained better validation accuracy than other versions for MobileNet V1. In the other hand, Inception V3 shows better results with version v2b.

Table 5. Summary of the two selected models for their performance according to each version

First Stage				
Model	**Version**	**Time/epochs[s]**	**Acc (Val)**	**Loss (Val)**
MobileNet V1	v1	392	0.8807	0.7560
MobileNet V1	v2	12	0.8883	0.5709
MobileNet V1	v2b	91	0.8788	0.6869
Inception V3	v1	810	0.8390	0.6038
Inception V3	v2	16	0.8252	0.4193
Inception V3	v2b	175	0.8845	0.4320

For the second stage, made the training of models applying the strategies described in section 6-D. For the Image Data Generator, we used a horizontal and vertical flip. We ignored both shear and rotation transformations. These transformations generate black corners and, it does not match with the usual dermatoscopic images. We also ignored the zoom transformation because the size of the skin lesion is relevant to diagnosis. Table 6 shows the results for these changes.

Table 6. Results obtained for version v3

Network	**Version**	**Acc (val)**	**loss (val)**
MobileNet V1	v3	0.8614	0.4768
Inception V3	v3	0.8561	0.3826

In the next stage, the changes made in version v3 were taken into account, k fold validation is added to the train process. Table 7 summarizes the results for these changes. Note that both validation accuracy and loss it is the K-mean value. Notice that the accuracy improve instead the v3 version.

Table 7. Results obtained for version v4

Network	Version	Acc (val)	loss(val)
MobileNet V1	v4	0.8990	0.4795
Inception V3	v4	0.8516	0.3279

In this last stage we apply the reduce learning rate callback. Table 8 summarizes the results for these changes. Note the improvement on both validation accuracy and loss. Also notice that MobileNet V1 shows a better performance than Inception v3.

Table 8. Results obtained for version v5

Network	Version	Acc (val)	loss(val)
MobileNet V1	v5	0.9295	0.2074
Inception V3	v5	0.9064	0.3312

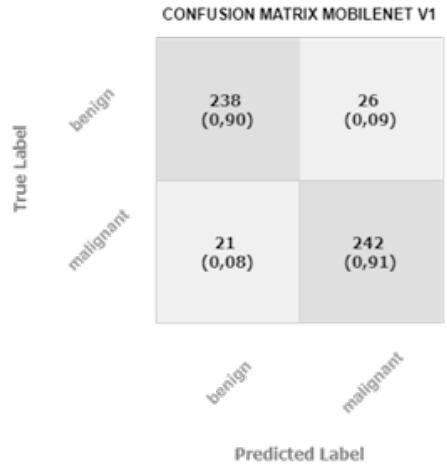
Regarding the results observed in the training process, we use the MobileNet V1 and Inception V3 best model (version v5) for testing on the device. For testing, we decided to use the F1 score, as validation measurement. Table 9 shows the results obtained for F1 score values for both each class and general, for benign and malignant classifications. Note that the performance reached by MobileNet V1 is better than Inception V3, and is according the results obtained in Table 8. The test results indicate that MobileNet V1 shows better performance than Inception V3.

Table 9. F1 score for benign and malignant classification

	F1 score		
	Benign	Malignant	Total
MobileNet V1	0.9109	0.9123	0.9104
Inception V3	0.8308	0.8276	0.8301

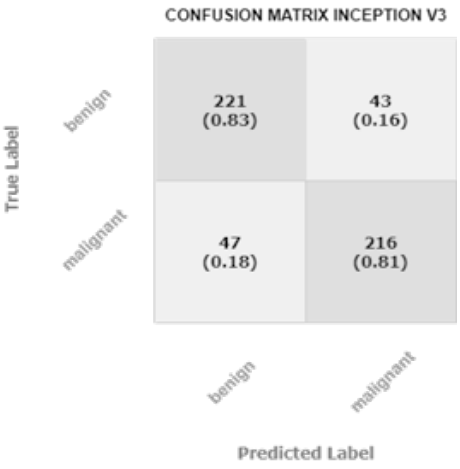
Figure 7 and Figure 8 shows the confusion matrix for the MobileNet V1 best model and the Inception V3 best model respectively. In this matrix is showed: both the number and the percentage of both benign and malignant images well and wrongly classified. Note that MobileNet V1 reach better accuracy levels than Inception V3.

Figure 7. Confusion matrix for MobileNet V1



Source: Self made

Figure 8. Confusion matrix for Inception V3



Source: Self made

Taking into account the matrix results, we evaluate the significant measures for medical classification. The FP indicator is one of the most important indicators. The measurement FP indicates the portion of patients without lesion but classified such an ill patient. It means that a healthy person will be treated unnecessarily, and it represent increased costs in health systems by adding overhead. Therefore, the goal is to keep

the FP near 0 % Other main measurement is the Recall. It measurement indicates the portion of patient well classified. It indicator is important because measure the portion of correct diagnosis. Thus, the goal is reach accuracy levels near 100 %. Table 10 shows the indicators obtained for malignant and benign class, using the MobileNet V1 best model. Notice that MobileNet V1 shows a better performance than Inception V3.

Table 10. Evaluation measurements for Malignant and benign classification

	Acc (Test)	Recall	FP
MobileNet V1	0.9106	0.9198	0.0965
Inception V3	0.8303	0.8197	0.1612

9. Conclusions

We have developed a prototype for malignant and benign lesions. For this task, we used the ISIC public databases with 2357 images of benign and malignant oncological diseases and the HAM10000 public database with 10015 images for academic machine learning purposes. From these databases we select the images of interest. For the training process we apply Transfer learning to Two pretrained CNN models, using diverse training strategies, tuning the hyperparameters by step-by-step methodology. The implemented models running in a Raspberry Pi3 B+ using the TensorFlow backend and a Python coded-based Interface.We tested the best versions of the training for two different architectures and our results suggest that MobileNet V1 achieves the best performance. Our results indicate that the use of data augmentation reduces the losses in the validation accuracy. Besides, the results suggest that the learning rate reduction task improve the validation accuracy and reduce the losses. The final result is a MobileNet V1 trained model that reaches an F1-score of 91.06%, a recall of 91.98% and an FP rate of 9.65% for the benign and malignant classification problem.

Our research indicates that transfer learning reduces considerably the required time for the training process and reduce the final losses for the same given accuracy. The results of this research contribute to the state of art regarding Deep learning applied to the development of medical applications. Also, this prototype is the first stage in future work, the results of this project will represent a helpful tool for dermatologists and future research of medical image classification in embedded systems.

10. Future Work

Taking into account the increase in skin cancer in Colombia, an alternative to implement in future work is to seek medical support from dermatologists,

dermatologists oncologists; to search or create skin cancer data banks in Colombia with a considerable size of images in order to offer a practical and simple low-cost tool with better performance for the classification of this type of diseases.

Bibliographic References

Castillo, J. A., Granados, Y. C., & Fajardo, C. A. (2020). Patient-specific detection of atrial fibrillation in segments of ecg signals using deep neural networks. *Ciencia E Ingenieria Neogranadina, vol. 30, no. 1*, 45–58.

Chanampe, H., Aciar, S., Vega, M. d., Molinari Sotomayor, J. L., Carrascosa, G., & Lorefice, A. (2019). Modelo de redes neuronales convolucionales profundas para la clasificacion de lesiones en ecograf´ ´ıas mamarias. *in XXI Workshop de Investigadores en Ciencias de la Computacion (WICC´ 2019, Universidad Nacional de San Juan).*

Chicco, D. J. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics, vol. 21, no. 1*, 6.

Diana, H. (2021). *Tesisredes-dianahellis-2021/skincancerclassifier.* Obtenido de https://github.com/TesisRedes-DianaHellis-2021?tab=repositorie

Gamerosa, P. C., & Tellez, J. E. (2016). El cancer de piel, un problema actual. *Revista de la Facultad de Medicina UNAM, vol. 59, no. 2*, 6–14.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 249–256.

Instituto de Hidrología, Meteorología y Estudios Ambientales. (2014). *"Indice ultravioleta (iuv) - ideam.* Obtenido de http://www.ideam.gov.co/web/tiempo-y-clima/indice-ultravioleta-iuv-

Kaggle. (s.f.). *Skin Cancer ISIC.* Obtenido de https://www.kaggle.com/nodoubttome/skin-cancer9-classesisic

Mao, H. H. (2020). A survey on self-supervised pre-training for sequential transfer learning in neural networks. *arXiv preprint arXiv:2007.00800*.

Mikołajczyk, A., & Grochowski, M. (2018). Data augmentation for improving deep learning in image classification problem. *2018 international interdisciplinary PhD workshop (IIPhDW). IEEE*, 117–122.

Morocho Jiménez, J. I. (2019). *Detección de tumores cutáneos malignos y*

benignos usando una red neuronal convolucional. Quito: B.S. thesis.

Ng, A., Besounda Mourris, Y., & Karantoforoosh, K. (2021). *Deep learning.* Obtenido de https://www.coursera.org/specializations/deep-learning

Pérez Lorenzo, C., & et al. (2019). *Detección precoz de cáncer de piel en imágenes basado en redes convolucionales,.* B.S. thesis.

Powers, D. M. (2010). Evaluation: from precesion, recall and f-measure to roc, informedness, makedness and correlation. *arXiv preprint a arxiv*, 16061.

Qin, Z., Yu, F., Liu, C., & Chen, X. (2018). How convolutional neural network see the world-a survey of convolutional neural network visualization methods. *arXiv preprint arXiv:1804.11191*.

Sanjay, M. (2018). *Why and how to Cross Validate a Model?* Obtenido de https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f

Shin, H.-C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., . . . Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging, vol. 35, no. 5*, 1285–1298.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data, vol. 6, no. 1,*, 60.

Sokolova, M., Japkowicz, N., & Szpakowicz, S. (2006). Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. *Australasian joint conference on artificial intelligence. Springer*, 1015–1021.

Static Raspberrypi. (s.f.). *Raspberry pi modelbplus product brief.* Obtenido de https://static.raspberrypi.org/files/product-briefs/raspberry-pi-modelbplus-product-brief.pdf

Team, K. (s.f.). *Keras documentation: Keras applications.* Obtenido de https://keras.io/api/applications

Tschandl, P. (2018). *The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions.* Obtenido de https://doi.org/10.7910/DVN/DBW86T

Xia, B., Zhang, H., & Li, Q. L. (2015). Pets: A stable and accurate redictor of protein-protein interacting sites based on xtremely-randomized ree. *IEEE Transactions on NanoBioscience, vol. 14*, 1–1, 11 .