

Red Neuronal Convolutiva Basada En El Procesamiento De Imágenes Para La Detección De Enfermedades Dermatológicas

JUNIOR

Ing. Paola Andrea García Meneses
Carrera de Ingeniería de Sistemas, Escuela Militar de Ingeniería
La Paz, Bolivia
paogar.meneses@gmail.com



Convolutional Neural Network Based On Image Processing For The Detection Of Dermatological Diseases

Resumen— El presente trabajo tiene como objetivo detectar imágenes según el tipo de enfermedad a la que pertenezcan. Mediante las redes neuronales, que son modelos para simular como procesa información el cerebro humano, el modelo aprenderá los patrones característicos de cada enfermedad a analizar. Al alojar una imagen para su procesamiento se brindará un resultado final a que enfermedad pertenece una imagen, brindando un resultado con la precisión esperada.

Palabras Claves— Redes neuronales convolucionales, procesamiento, diagnóstico, machine learning, resolución, matriz.

Abstract— The present work aims to detect images according to the type of disease to which they belong. Using neural networks, which are models to simulate how the human brain processes information, the model will learn the characteristic patterns of each disease to be analyzed. Hosting an image for processing will provide a final result that the disease belongs to an image, providing a result with the expected precision.

Keywords— Convolutional neural networks, processing, diagnosis, machine learning, resolution, matrix.

1. INTRODUCCIÓN

Una de las ramas de inteligencia artificial es el Deep learning que se basa en las redes neuronales artificiales, su objetivo es buscar modelos que solucionen problemas difíciles mediante técnicas algorítmicas, a través de un conjunto de datos para su aprendizaje. Se han desarrollado varias herramientas aplicando distintos tipos de redes neuronales para identificar enfermedades de especialidades médicas como la diabetes, cáncer en la piel, conjuntivitis entre otras.

El hospital Arco Iris brinda servicios de salud a la población en general, es un hospital de tercer nivel, se puede acceder a una cita médica, mediante una reserva previa donde todas las especialidades médicas están disponibles. Se caracteriza por ser un hospital que ofrece servicios a bajo costo.

En este hospital se detectó que los médicos dermatólogos usan instrumentos mecánicos para la detección de enfermedades, además que cuentan con pocos sistemas para la automatización. Los turnos de citas médicas se llenan con bastante facilidad, por este motivo los doctores deben asegurarse de brindar consultas con la calidad en un tiempo reducido.

El presente proyecto de grado brindará un diagnóstico de enfermedades dermatológicas, por medio de la detección de imágenes. El médico tomará al paciente una foto de su padecimiento en la consulta médica, esta imagen será la entrada de la red neuronal convolutiva donde en base a sus patrones característicos, se clasificará la foto con la enfermedad del paciente.

Para el proyecto de grado se utilizarán las redes neuronales artificiales, que es un paradigma de aprendizaje y los procesamiento automáticos que están inspirados en el funcionamiento del sistema nervioso humano, el cual consiste en un sistema de interconexión de neuronas en una red que colabora entre ellas para crear una respuesta de salida. Una subclasificación de las redes neuronales son las redes neuronales convolucionales, esta red aprende a reconocer una gran cantidad de patrones dentro de imágenes, pero para ello

necesitan entrenarse previamente con una cantidad importante de datos. En este proyecto el data set será un banco de imágenes clasificadas por enfermedades dermatológicas.

II. MATERIALES Y METODOS

Se cuenta con tres data set (acné, rosácea, vitiligo), en el cual se quiso ampliar y aumentar los datos, por lo que se utilizó “Data Augmentation”, que es el aumento de datos y se puede utilizar para expandir artificialmente el tamaño de un conjunto de datos de entrenamiento mediante la creación de versiones modificadas de imágenes en el conjunto de datos.

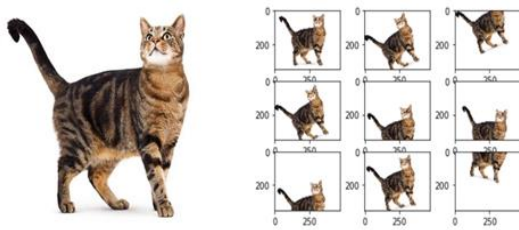


Figura 1. Técnica Data Augmentation

MobileNet es una arquitectura de redes neuronales convolucionales eficiente y portátil, comparado a otras arquitecturas de RNC MobileNet es ligero, estable y de libre uso. Una vez implementado no consume recursos y es rápido al sacar un resultado. La arquitectura de MobileNet clasifica hasta 1000 clases, esto quiere decir, si se hace el reconocimiento de imágenes sobre razas de perro esta arquitectura, puede identificar 1000 razas (teniendo los data set correspondientes), para este proyecto solo se utilizará tres clases o salidas debido a que solo se clasificarán 3 enfermedades dermatológicas es por ello que esta cantidad debería tener la salida de la red neuronal convolucional.

Se desarrolló la RNC con la ayuda del entorno de programación Colabority o también llamado Colab, el cual permite ejecutar y programar en Python desde el navegador, además tiene las siguientes ventajas:

- No requiere configuración
- Es de libre uso
- Da acceso gratuito a GPUs llamado "Colab"
- Tiene instalado todas las librerías de Python

Los datos que fueron utilizados para que la red neuronal aprenda fueron los siguientes:



Figura 2. Carpetas de almacenamiento para el proceso de aprendizaje de la red neuronal convolucional

Cada carpeta almacena internamente las subcarpetas 0, 1, 2 las cuales contienen las enfermedades de acné, rosácea y vitiligo respectivamente. Las imágenes están recortadas de tal modo que muestra solo la zona afectada es decir que no tienen ruido y tampoco muestran otros elementos (manos, ojos, nariz, fondo de captura, etc.). Cabe resaltar que las carpetas train y valid sirven para que la RNC aprenda, mientras que la carpeta de test sirve para comprobar que tan bien aprendió.

A continuación, se indicará la función de las carpetas

- Carpeta train: La carpeta train ayuda a que la red neuronal aprenda mediante imágenes los patrones característicos de cada enfermedad, en esta carpeta se aloja la mayor cantidad de imágenes a comparación de las otras carpetas, esto se debe a que mientras más datos la red neuronal tenga, mayor será su aprendizaje. Un 68% de las imágenes están en esta carpeta.
- Carpeta valid: La carpeta valid funciona como un examen diagnóstico para la RNC, tiene como finalidad ver los resultados de como aprendió la red esta carpeta sirve en la etapa de entrenamiento. Un 16% de imágenes está dentro de esta carpeta.
- Carpeta test: Tiene la funcionalidad de evaluar. Contiene imágenes que desconoce la RNC, de esta manera se sabrá como predice el modelo. Esta carpeta contiene 16% del total de imágenes.

Con la librería `image_data_generator` obtenemos los datos separados (train, test, valid) y lo preparamos con los “requisitos” de entrada a la arquitectura MobileNet. Añadido a eso le ajustamos el size y lo preparamos en lotes.

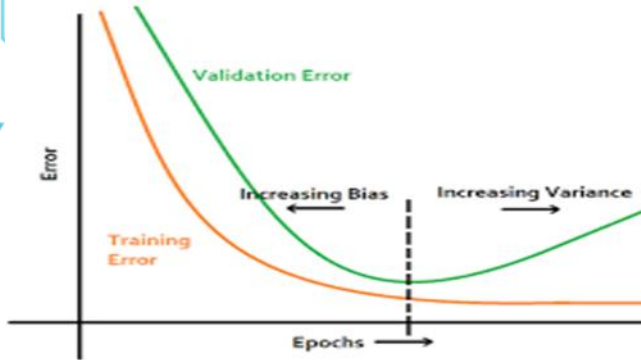


Figura 3. . Gráfica de los errores vs. las épocas en la etapa de entrenamiento de la red neuronal convolutacional

Mediante la figura 3 se grafica los errores que llega a tener una red neuronal convolutacional, mientras se inicia el proceso de entrenamiento la red neuronal tendrá mayor número de errores, a medida que pasan las épocas el modelo tiende a presentar menos errores, pero esto no significa que con un número alto de épocas se halla la cantidad mínima de error, en algún punto el error llega a ser mínimo pero conforme pasan las épocas va a seguir siendo el mismo, es decir que se mantiene constante el error como se muestra en la línea naranja

En teoría todos los modelos de aprendizaje deberían tener similitud a la gráfica naranja, donde poco a poco baja en nivel, pero la mayoría de modelo en la realidad tienden a la línea verde. Asimismo, la línea verde representa que en algún punto llega a ser optimo el error, pero por algún motivo (cantidad de datos, arquitectura de RNC, entre otros) en las siguientes épocas incrementa el número de errores.

Es por ello que se vio necesario utilizar la librería EarlyStopping el cual congela los pesos de la neurona, para tener el mínimo error y guardar el más óptimo.

La exactitud o precisión que maneja la red neuronal se representa mediante una formula:

$$Accuracy = \frac{N^{\circ} \text{ de imagenes bien clasificadas}}{Total \text{ de imagenes}}$$

El mejor resultado es 1 donde la red neuronal predijo a la perfección todos los datos (imágenes de enfermedades dermatológicas). El peor resultado es 0 donde la red neuronal no pudo predecir ninguna enfermedad. El rango de exactitud tomará valores entre 0 y 1 y trabaja como una probabilidad.

III. RESULTADOS

El funcionamiento de la red neuronal convolutacional, la imagen de entrada se pasa a una matriz de pixeles, luego se pasa por filtro (una matriz de menor tamaño que la imagen de entrada), este filtro buscará patrones significativos de la imagen para que en una posterior etapa sea reducida de tamaño mediante la función de activación este proceso es iterativo hasta que logre terminar las capas convolucionales de la arquitectura y al final brinda una respuesta final y es saber a qué clase pertenece la imagen. El resultado tanto de la detección de imágenes como la evaluación del médico dermatólogo será la patología que presenta el paciente.

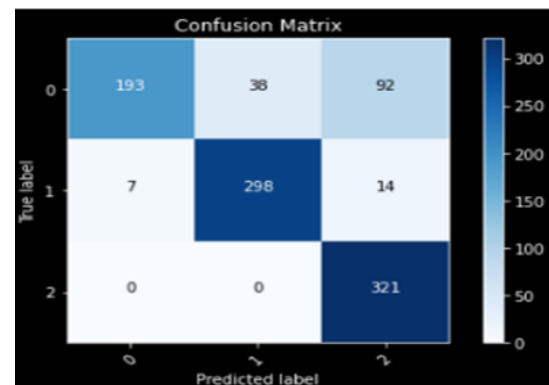


Figura 4. Matriz de confusión con resultado de la predicción de la red neuronal convolutacional

La matriz de confusión es una herramienta muy útil para valorar cómo de bueno es un modelo clasificación basado en aprendizaje automático, los resultados de la etapa de predicción son:

- Acertó 193 imágenes que pertenecían a acné
- Acertó 298 imágenes que pertenecían a rosácea.
- Acertó 321 imágenes que pertenecían a vitiligo.

Las imágenes que pertenecían a vitiligo, la RNC no tuvo ninguna equivocación. Mientras que la red no pudo acertar las siguientes imágenes:

- Las imágenes que pertenecían a acné, la RNC predijo que 38 imágenes eran la enfermedad de rosácea.
- Las imágenes que pertenecían a acné, la RNC predijo que 92 imágenes eran la enfermedad de vitiligo.
- Las imágenes que pertenecían a rosácea, la RNC predijo que 7 imágenes eran la enfermedad de acné.
- Las imágenes que pertenecían a rosácea, la RNC predijo que 14 imágenes eran la enfermedad de vitiligo.

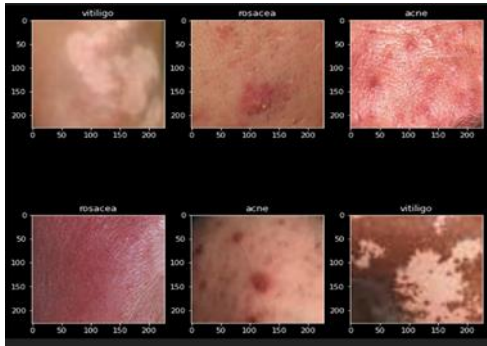


Figura 5. Predicciones de la red neuronal convolucional

IV. CONCLUSIONES

Mediante las fases de la metodología visión artificial, el ordenador pudo comprender, distinguir y clasificar imágenes con padecimientos de las tres enfermedades analizadas, también se logró hacer una buena estructura en cuanto al diseño de la red neuronal convolucional en base a las capas convolucionales, capas max-pooling y las funciones de activación. Este diseño sirvió para que la detección de enfermedades sea más precisa al momento de dar un resultado final.

Se ha demostrado que se puede trabajar con dos metodologías distintas haciendo un híbrido entre la fusión de ellas, se trabajó en la parte de ingeniería de software con la metodología XP que sirvió para establecer un marco de trabajo, en el cual se pudo analizar, planificar y controlar el proceso de desarrollo del software web, asimismo se trabajó con la metodología visión artificial para procesar imágenes de enfermedades dermatológicas con el fin de producir información que pueda ser entendida por la computadora. La combinación de las dos metodologías resultó ser adecuada para el desarrollo del proyecto porque se obtuvo las respuestas esperadas, esto se verificó en las pruebas que se hicieron.

Se logró recopilar imágenes de enfermedades dermatológicas de buena calidad para el entrenamiento de la red neuronal convolucional, con la particularidad de que solo tengan los patrones característicos del padecimiento, además se hizo una selección de muestras restringiendo el ruido en la imagen y teniendo una alta resolución para su futuro procesamiento

V. APENDICES

Entrenar una red neuronal consiste en ajustar cada uno de los pesos de las entradas de todas las neuronas que forman parte de la red neuronal, para que las respuestas de la capa de salida se ajusten lo más posible a los datos del aprendizaje.

```
Epoch 1/20
427/427 - 1624s - loss: 0.5621 - accuracy: 0.8848 - val_loss: 7.7824 - val_accuracy: 0.566
Epoch 2/20
427/427 - 30s - loss: 0.1936 - accuracy: 0.9365 - val_loss: 1.5041 - val_accuracy: 0.7141
Epoch 3/20
427/427 - 30s - loss: 0.1558 - accuracy: 0.9506 - val_loss: 2.5947 - val_accuracy: 0.6712
Epoch 4/20
427/427 - 29s - loss: 0.1452 - accuracy: 0.9522 - val_loss: 3.5846 - val_accuracy: 0.4398
Epoch 5/20
427/427 - 30s - loss: 0.1104 - accuracy: 0.9637 - val_loss: 3.5188 - val_accuracy: 0.6607
Epoch 6/20
427/427 - 28s - loss: 0.0895 - accuracy: 0.9707 - val_loss: 1.9773 - val_accuracy: 0.6105
Epoch 7/20
427/427 - 30s - loss: 0.1028 - accuracy: 0.9700 - val_loss: 2.5005 - val_accuracy: 0.6398
Epoch 8/20
427/427 - 28s - loss: 0.0998 - accuracy: 0.9712 - val_loss: 2.1502 - val_accuracy: 0.6335
Epoch 9/20
427/427 - 29s - loss: 0.0854 - accuracy: 0.9738 - val_loss: 3.9855 - val_accuracy: 0.4241
Epoch 10/20
427/427 - 29s - loss: 0.0654 - accuracy: 0.9799 - val_loss: 1.5815 - val_accuracy: 0.7005
Epoch 11/20
427/427 - 28s - loss: 0.0576 - accuracy: 0.9827 - val_loss: 11.6255 - val_accuracy: 0.3455
Epoch 12/20
427/427 - 29s - loss: 0.0560 - accuracy: 0.9824 - val_loss: 1.9910 - val_accuracy: 0.6094
Epoch 13/20
427/427 - 28s - loss: 0.0595 - accuracy: 0.9792 - val_loss: 3.2493 - val_accuracy: 0.6419
Epoch 14/20
427/427 - 30s - loss: 0.0634 - accuracy: 0.9799 - val_loss: 1.9048 - val_accuracy: 0.6503
Epoch 15/20
427/427 - 29s - loss: 0.0477 - accuracy: 0.9855 - val_loss: 2.8044 - val_accuracy: 0.5990
Epoch 16/20
427/427 - 28s - loss: 0.0409 - accuracy: 0.9871 - val_loss: 2.7424 - val_accuracy: 0.7005
Epoch 17/20
427/427 - 29s - loss: 0.0648 - accuracy: 0.9824 - val_loss: 2.2803 - val_accuracy: 0.6209
Epoch 18/20
427/427 - 29s - loss: 0.0519 - accuracy: 0.9820 - val_loss: 5.4945 - val_accuracy: 0.6513
Epoch 19/20
427/427 - 28s - loss: 0.0535 - accuracy: 0.9813 - val_loss: 1.2680 - val_accuracy: 0.6974
Epoch 20/20
427/427 - 29s - loss: 0.0327 - accuracy: 0.9874 - val_loss: 1.9070 - val_accuracy: 0.6293
<tensorflow.python.keras.callbacks.History at 0x7fc761802850>
```

Referencias

- [1] Moret Bonillo, V. (2000.). Fundamentos de inteligencia artificial. Santiago de Compostela: Universidad de Coruña.
- [2] Ortiz, P. S. (2009). Vision Artificial para el control en industrias. Mexico.
- [3] Pino Díez, R., Gómez, A. G., & Martínez, y. N. (2001). Introducción a la inteligencia artificial sistemas expertos, redes neuronales artificiales y computación evolutiva. España Madrid: Universidad de Oviedo.
- [4] Roger S. Pressman, P. (2002). Ingeniería del software U N ENFOQUE PRÁCTICO. Madrid: MCGRAW-HILL INTERAMERICANA 5a. ed.
- [5] Rouhiainen, L. (2018). Inteligencia artificial: 101 cosas que debes saber hoy sobre nuestro futuro. España: Alienta Editorial.

- [6] Russell, S. J. (2004). Inteligencia artificial: un enfoque moderno. Madrid/México: Pearson/Prentice Hall.
- [7] Sampieri, R. H. (2003). Metodologías para la investigación. McGra-Hill.
- [8] Severance, C. (2020). Python para todos. España.
- [9] Vilet, J. R. (2005). Apuntes de Procesamiento Digital de Imágenes. San Luis Potosí: Universidad Autónoma de San Luis Potosí.
- [10] Viñals, J. T. (2015). Introducción al aprendizaje por refuerzo profundo | Teoría y práctica en Python. WATCH THIS SPACE.
- [11] Welling, L. (2001). Desarrollo Web con PHP y MySQL. Caracas: Dialnet
- [12] Ackerman, A. B. (2002). Atlas clínico de las 101 enfermedades más comunes de la piel . Hospital Manual Plus : Editor Marbán.
- [13] Bauer, F. (1968). NATO Software Engineering Conference . Alemania.
- [14] Benk, K. (1999). Programación extrema explicado. Boston: Addison Wesley.
- [15] Cadavid, F. M. (2013). Revisión de metodologías ágiles para el desarrollo de software. Artículo.
- [16] Debrauwer, F. V. (2016). UML 2.5: iniciación, ejemplos y ejercicios corregidos. Alemania: Ediciones ENI.
- [17] Enrique Alegre, G. P. (2016). Metodos y conceptos en vision por computador. España: Grupo de la vision del comite de automática (CEA).
- [18] Henry, K. (1995). Fundamentos de base de datos. Madrid: McGraw-Hill.
- [19] Heurtel, O. (2016). PHP 7: Desarrollar un sitio web dinámico e interactivo. ENI.
- [20] Kaplan, J. (2017). Inteligencia Artificial: Lo que todo el mundo debe saber. Madrid España: TEELL EDITORIAL.
- [21] Kendall, K. &. (2011). Analisis y diseño de sistemas. México: Pearson.
- [22] Larman, C. (2002). UML y Patrones. Mexico: ALHAMBRA; 2nd edición.
- [23] Letelier, T. (s.f.). Metodologías Ágiles para el desarrollo de Software eXtreme Programming XP. Valencia: Departamento de sistemas informaticos.
- [24] Manian, D. (2012). HTML5 Boilerplate web Development. Packt Publishing.
- [25] Maravall, D. (1993). Reconocimiento de formas y vision Artificial. RA-MA.
- [26] Martin F. Kendall, S. (1999). UML Gota a Gota. Santiago.
- [27] Mendoza, G. d. (2008). Análisis Coste-Beneficio. Editorial Ariel.
- [28] Montero, B. M. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. Espirales, 10.
- [29] Moret Bonillo, V. (2000.). Fundamentos de inteligencia artificial. Santiago de Compostela: Universidad de Coruña.
- [30] Ortiz, P. S. (2009). Vision Artificial para el control en industrias. Mexico

Fecha de Envío del Artículo: 5/07/2021
Fecha de Aceptación de artículo: 13/07/2021