Universidad Autónoma de Nuevo León

Facultad de Ciencias Físico Matemáticas

Detección de células infectadas con Malaria

Proyecto Final

Autor:
Lic. Leobardo García Reyes

Supervisado por:
Dr. José de Jesús Rocha Salazar

31 de marzo del 2022

Detección de células infectadas con Malaria

Introducción

La malaria, también conocida como paludismo, es causada por parásitos protozoarios del género Plasmodium. Se transmite generalmente entre humanos y otros animales mediante picaduras de mosquitos Anopheles, de los cuales 50 pueden infectar las células rojas de la sangre[4][5]. La malaria generalmente prevalece en los países menos desarrollados y zonas tropicales, en donde casi todos los bebés, niños y adultos tienen como resultado una concentración de hemoglobina reducida. Esto sucede especialmente durante las estaciones lluviosas, cuando la transmisión de la malaria es más alta[5].

Según la OMS (Organización Mundial de la Salud), la región de África mantiene un alto grado de casos confirmados de malaria del 95 % y del 96 % por defunciones, en donde el 80 % corresponde a niños menores de 5 años en 2020. También se calcula que en todo el mundo hubo 241 millones de casos de malaria y se estima que la mortalidad por malaria fue de 627,000[1].

Existen diferentes técnicas para el diagnóstico de la malaria como la microscopía, las pruebas de diagnóstico rápido (RDT), la reacción en cadena de la polimerasa (PCR), entre otras. Sin embargo, algunas de estas pruebas muestran un creciente de falsos negativos, pero siguen siendo utilizados debido a su facilidad, rapidez y rentabilidad. Por otra parte, pruebas no invasivas que no requieren de sangre, utilizando saliva u orina, son técnicas que están en vías de desarrollo que potencialmente pueden ayudar a prevenir y eliminar la malaria[6].

Entorno a la problemática que existe en la detección de la malaria y las técnicas costosas, este documento propone un modelo de aprendizaje profundo de una Red Neuronal Convolucional (CNN) que ayude a clasificar células entre "Parasitario" y "No Infectado" (Fig. 1) con ayuda de imágenes diapositivas de frotis de sangre delgada de la actividad de investigación Malaria Screener[2].

Datos

El conjunto de datos que se selecciono para el entrenamiento del algoritmo, se obtuvo a través del sitio web "Kaggle"[3], el cual referencia al sitio web "National Library of Medicine" (NIH) como el repositorio oficial de los datos[2]. El conjunto de datos contiene células segmentadas en imágenes de diapositivas de frotis de sangre delgada de la actividad de investigación Malaria Screener. Se recolectaron y fotografiaron frotis de sangre delgados con tinción de Giemsa de 150 pacientes infectados con malaria y 50 pacientes sanos en el Chittagong Medical College Hospital, Bangladesh. Los de NIH aplicaron un algoritmo basado en un conjunto de niveles para detectar y segmentar los glóbulos rojos[2]

En el sitio web "Kaggle", el conjunto de datos esta dividido de la siguiente forma:

- **Conjunto de entrenamiento:** 27,558 imágenes de células con instancias iguales de células parasitarias y no infectadas (Fig. 1).
- **Conjunto de prueba:** 15,832 imágenes de células con instancias iguales de células parasitarias y no infectadas (Fig. 1).
- **Conjunto de predicción:** 2 imágenes de células con instancias iguales de células parasitarias y no infectadas (Fig. 1).

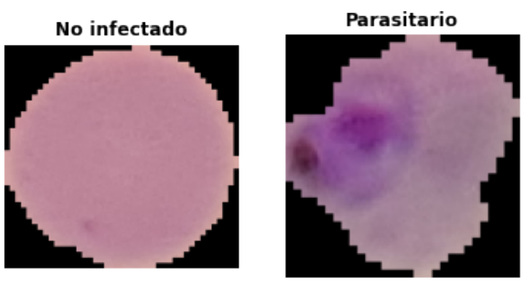


Figura 1: Ejemplo de una célula no infectada y parasitaria.

Metodología y Modelo

Pre-procesamiento

En esta sección se describirá la metodología implementada en el algoritmo CNN para obtener resultado favorables. El proceso comienza con una etapa de pre-procesamiento para normalizar las imágenes, seguido de la variación aleatoria de las imágenes para entrenar el modelo. La separación de los conjuntos de entrenamiento y validación, un algoritmo de CNN y, finalmente, la predicción del algoritmo en las imágenes de prueba.

En el apartado del pre-procesamiento de las imágenes, se realizó la normalización de las imágenes de entrada para las carpetas de entrenamiento y prueba. En este proceso, se escalan los valores de intensidad de los píxeles a valores entre 0 y 1. Una vez normalizada las imágenes, se incorporó el proceso de cambio aleatorio de imágenes. La incorporación de este proceso, aplica variaciones aleatorias a las imágenes de entrenamiento: giro horizontal aleatorio, giro vertical aleatorio y rotación aleatoria de 45°. También, como las imágenes tienen diferentes tamaños, se re-dimensiono cada una a 130x130.

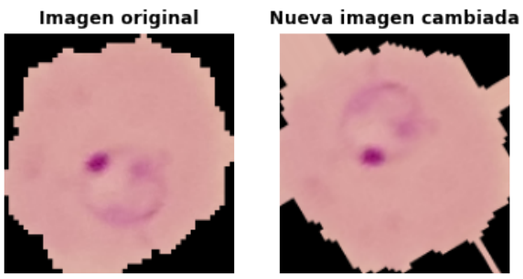


Figura 2: Nueva imagen generada.

Una vez aplicado la normalización de las imágenes, el cambio aleatorio y el re-dimensión de estas mismas; se realizó una división a la carpeta de entrenamiento para separarla como conjunto de entrenamiento (60 %) y conjunto de validación (40 %). Este conjunto de validación servirá de apoyo para el entrenamiento del algoritmo, confirmando si las decisiones que esta tomando se ajusta bien o peor a las imágenes reales. En el siguiente diagrama (Fig. 3) se puede observar el proceso descrito:

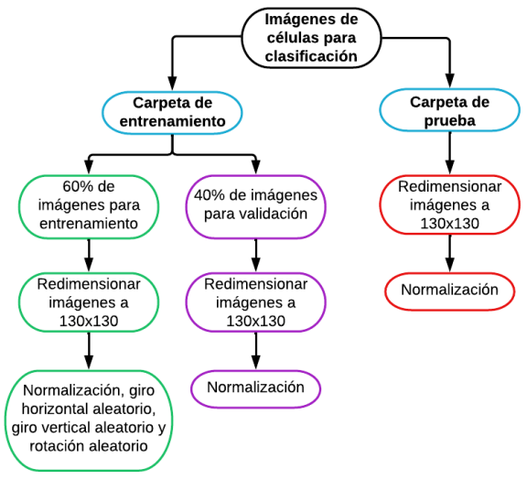


Figura 3: Diagrama del procesamiento de las imágenes.

Construcción de la Red Neuronal Convolucional

Como segundo paso, se implemento la CNN (Fig. 5), la cual se encargará de procesar y extraer características en imágenes 2D. El algoritmo se conforma de 2 partes: Extracción de características y Clasificador. En la extracción de características se reciben los datos de entrada de la capa anterior inmediata, pasando a la salida de la siguiente capa. En el arquitectura de la CNN, se emplean 4 tipos de capas:

- **Convolución:** Esta capa se utiliza para extraer características importantes de la imagen de entrada.
- **Pooling:** Esta capa se utiliza para reducir las dimensiones de los mapas de características. Este apartado existen diferente tipos de pooling: máximo, promedio, Suma, etc.
- **Flatten:** Una vez reducida la dimensión de la imagen en una matriz más pequeña, convierte esta matriz a un vector.
- **Clasificación:** Aquí es donde se decidirá si es una célula parasitarias o no infectadas.

En la parte de convolución se utilizó 4 capas de convolución, en donde cada una su propósito es extraer características más relevantes de cada imagen. Todas las capas de convolución se definió una matriz de Kernel de 3x3, en donde esta matriz se deslizará por la imagen de entrada (1, 1). Como función de activación se usó ReLU, que sirve para cambiar los números negativos a 0, ya que no es posible tener píxeles negativos. El número de filtros que se eligió para cada capa de convolución fue de 16, 32, 64 y 128.

Adicional a las capas de convolución, se les agrego un parámetro llamado padding, debido a que existe situaciones en la que, cuando se mueve nuestra matriz de filtro y llega al borde, no se ajusta a la matriz de entrada. Por este motivo, se rellenó la matriz de entrada con ceros y a esto se le llama "same padding".

Para elegir que pooling es el más adecuado entre el máximo y promedio, se represento en la Fig. 4. El pooling máximo selecciona los píxeles más brillantes de la imagen, es útil cuando el fondo de la imagen es oscuro y solo nos interesan los píxeles más claros de la imagen. Mientras el pooling promedio suaviza la imagen y, por lo tanto, es posible que no se identifiquen las características nítidas[7].

Entonces, se implemento 2 algoritmos de CNN, uno con pooling máximo y otro con el promedio para probar cual de los dos daba mejor rendimiento. Observando la Fig. 4, se puede decir que no hay gran diferencia uno con el otro, pero el promedio resalta un poco más la característica para determinar si la célula esta infectada o no.

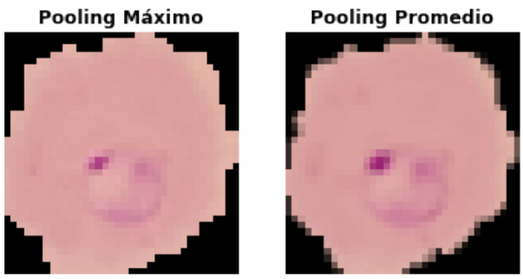


Figura 4: Comparación de pooling entre máximo y promedio.

Una vez extraídas las características en las capas de convolución y pooling se procede al paso de clasificación, se aplanan transformando la matriz en un vector para entrar a una capa de neuronas totalmente conectada. En esta capa se encuentran 512 neuronas y una función de activación ReLU. A esta misma capa se le aplica un dropout del 25 %, esto con la finalidad para que los datos no aprendan un solo camino, es decir, si una neurona se apaga otras neuronas tendrán que intervenir y manejar la representación requerida para hacer predicciones para las neuronas que faltan. En consecuencia, da como resultado una red que es capaz de una mejor generalización y es menos probable que sobre-ajuste los datos de entrenamiento.

Por ultimo, en la capa de salida se encuentra 1 neurona, porque solo se tiene 2 etiquetas en las cuales se puede clasificar: 0-No infectado y 1-Parasitario. Y como función de activación se aplicó la sigmoide, esta función nos regresará valores entre 0 y 1. Entonces, para determinar que etiqueta le corresponde, se define un umbral de clasificación de 0.5, donde lo valores mayores al umbral se le asigno la etiqueta de 1-Parasitario y menores o iguales al umbral se le asigno la etiqueta de 0-No infectado.

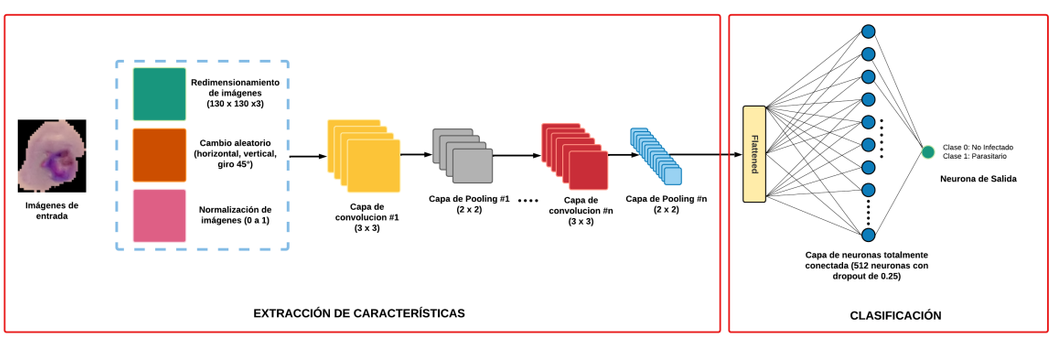


Figura 5: Arquitectura de la CNN.

Métricas, función de perdida y optimizador

Si bien la preparación de los datos y el entrenamiento del modelo es un punto clave, también lo es el medir el rendimiento de este modelo entrenado. Entonces, se utilizará las siguientes métricas[8]:

- **Matriz de confusión:** Es una tabla que describe el desempeño de un modelo de clasificación en un conjunto de datos de prueba cuyos valores verdaderos son conocidos. Esta matriz de confusión puede ser utilizada para estimar otras métricas como en la Fig. 6[9].

En donde, **Verdadero Negativo (TN):** El valor real es negativo y la prueba predijo que el resultado era negativo.
Verdadero Positivo (TP): El valor real es positivo y la prueba predijo que era positivo.
Falso Negativo (FN): El valor real es positivo, y la prueba predijo que el resultado es negativo.
Falso Positivo (FP): El valor real es negativo, y la prueba predijo que el resultado es positivo.

- **Accuracy (Exactitud):** Es la proporción de resultados verdaderos (Verdaderos Positivos como Verdaderos Negativos) dividido entre el número total de casos examinados (Verdaderos Positivos, Falsos Positivos, Verdaderos Negativos, Falsos Negativos). Sin embargo, una de sus desventajas es que no funciona bien con clases desequilibradas[9].

$$Acurracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision (Precisión):** Es la proporción de Verdaderos Positivos dividido entre los identificados como positivos (Verdaderos Positivos y Falsos Positivos). Sin embargo, una de sus desventajas es que no funciona bien con clases desequilibradas[9].

$$Precision = \frac{TP}{TP + FP}$$

- **Recall (Sensibilidad):** Es la proporción de Verdaderos Positivos que fueron correctamente identificados por el algoritmo entre el total de Verdaderos Positivos (Verdaderos Positivos y Falsos Negativos). Representa la fracción de Verdaderos Positivos[9].

$$Recall = \frac{TP}{TP + FN}$$

- **Specificidad (Especificidad):** Es la proporción de Verdaderos Negativos que fueron correctamente identificados por el algoritmo entre el total de Verdaderos Negativos (Verdaderos Negativos y Falsos Positivos). Representa la fracción de Verdaderos Negativos[9].

$$Specificidad = \frac{TN}{TN + FP}$$

- **F1 Score (Puntaje F1):** Asume que nos importa de igual forma la precisión y la exhaustividad[9].

$$F1 = 2\Big(\frac{Precision * recall}{Precision + recall}\Big)$$

- **AUC (Área Bajo la Curva):** Este puntaje nos da una buena idea de qué tan bien funciona el modelo.[9].

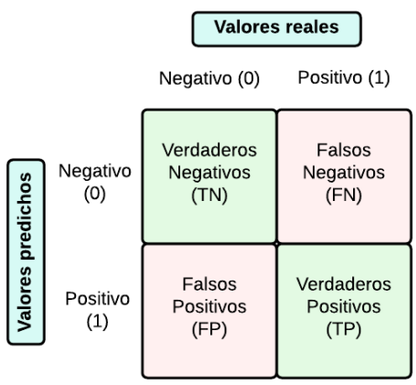


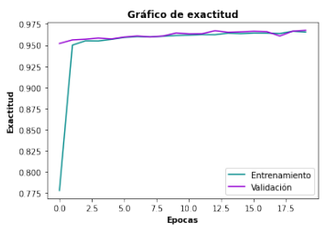
Figura 6: Ejemplo de representación de la matriz de confusión binaria.

Por otra parte, como función de perdida se utilizó la entropía cruzada binaria, ya que penaliza la probabilidad en función de qué tan lejos está del valor esperado real. La penalización es de naturaleza logarítmica, lo que genera una puntuación grande para las diferencias grandes cercanas a 1 y una puntuación pequeña para las diferencias pequeñas que tienden a 0[10]. Y por ultimo, como optimizador se utilizó el optimizador de Adam como alternativa para compararlo con el SGD y probar si obtenían buenos resultados.

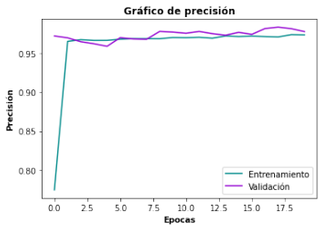
Resultados

En este apartado, se presentan los resultados del rendimiento obtenido al entrenar el modelo y validarlo. El conjunto de entrenamiento obtiene una exactitud del 96.55 %, una precisión del 97.34 % y una sensibilidad del 95.72 %. Y para el conjunto de validación 96.75 %, 97.76 % y 95.70 % respectivamente.

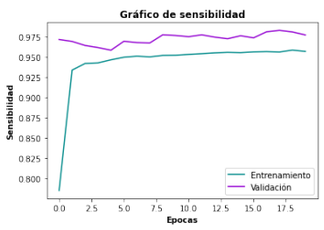
En cuento a la función de perdida es del 10.06 % y 8.91 %. En la siguiente Fig. 7 se muestra lo anterior descrito.



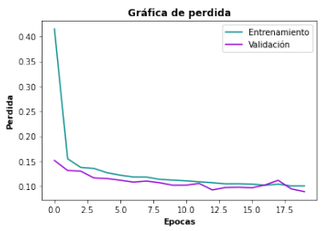
(a) Gráfica de exactitud.



(b) Gráfica de precisión.



(c) Gráfica de sensibilidad.



(d) Gráfica de perdida.

Figura 7: Gráficos de métricas utilizadas.

Una vez entrenado el modelo, se procese a evaluar el modelo con el conjunto de prueba, y como antes se había dicho, de acuerdo al umbral de clasificación se construye la matriz de confusión. Esta matriz se revelará que es lo que esta clasificando bien y mal entre "Parasitario" y "No Infectado" como se puede observar en la Fig. 8. A partir de esto se calculan las diferentes métricas.

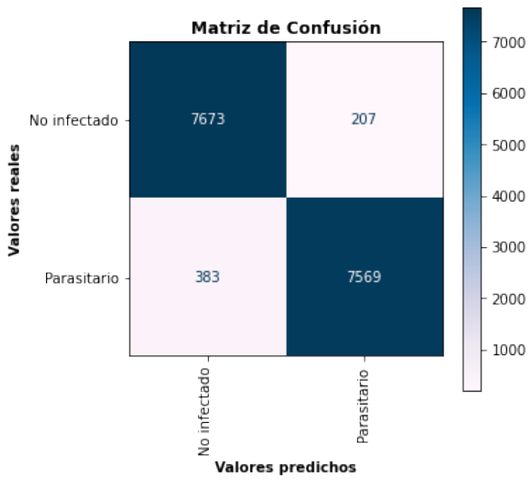


Figura 8: Matriz de confusión.

En la Tabla 1 se compara el modelo actual contra el modelo utilizando un pooling promedio y como se puede observar, no hay una gran diferencia. Entonces, los dos pooling, utilizando las mismas características de la CNN propuesta sirven para obtener buenos resultados.

Métrica	**Pooling Máximo**	**Pooling Promedio**
Exactitud	96.27 %	96.37 %
Precisión	97.34 %	97.01 %
Sensibilidad	95.18 %	95.74 %
Especificidad	97.37 %	97.01 %
Puntaje F1	96.25 %	96.37 %
Perdida	9.96 %	10.23 %

Cuadro 1: Comparación del pooling máximo y promedio.

Con ayuda de la sensibilidad y 1-especificidad se puede construir la curva ROC (Característica Operativa del Receptor), es un gráfico que muestra el rendimiento de un modelo de clasificación en todos los umbrales de clasificación. Y 1-especificidad es la tasa de falsos positivos. En la Fig. 9 se puede observar la curva ROC del modelo.

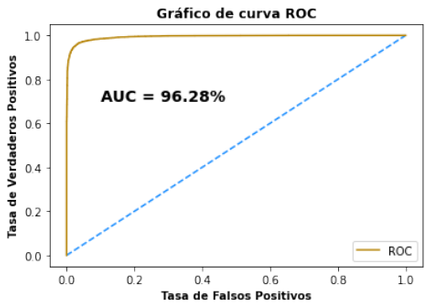


Figura 9: Gráfica de la curva ROC.

Por ultimo, se realizó una estimación del conjunto de predicción de la siguiente imagen (Fig. 10), la cual clasificó correctamente como Parasitario.

Figura 10: Predicción de una nueva imagen.

Conclusiones

La investigación y construcción de diferentes métodos nos da una vista más amplia sobre que metodología hay que elegir para obtener los mejores resultados. En este caso, al emplear una técnica para cambiar las imágenes en nuestro conjunto de datos tuvo un gran impacto en los resultados y en la mejora de la precisión del modelo.

Observamos que los filtros y etapas de pre-procesamiento aplicados, pueden ser de gran utilidad para resaltar características de interés en la imagen y/o corregir cuestiones propias del conjunto de datos original. En este apartado, se reafirma que para realizar pruebas, es importante ejecutar el código con una GPU de ser posible. Esto resulta fundamental para hacer varias pruebas modificando parámetros, etapas y características tanto del modelo como de los datos.

En conjunto, los puntos anteriores son aprendizajes fundamentales para buscar construir el mejor modelo posible que, de acuerdo con los resultados obtenidos, mejora en todos los aspectos con una versión nueva del modelo.

Por ultimo, es la importancia o resolución de problemas que puede abordar las redes neuronales, ya que que pueden reemplazar herramientas más costosas y hacer más eficaz el trabajo que se le esta solicitando. También, cada vez se generan más datos que contienen información valiosa que puede ser aprovechada. Y es aquí donde entra el deep learning para que las máquinas puedan detectar patrones, tendencias o relaciones entre los datos y adquirir información valiosa de lo analizado.

Referencias

[1] Paludismo. (2021, diciembre 6). Organización Mundial de la Salud. https://www.who.int/es/news-room/fact-sheets/detail/malaria.

[2] (S/f). National Library of Medicine. Recuperado el 21 de marzo de 2022, de https://lhncbc.nlm.nih.gov/LHC-downloads/downloads.html.

[3] Aniket, S. P. (s/f). Malaria Dataset [Data set]. En Malaria Dataset. Recuperado el 21 de marzo de 2022, de https://www.kaggle.com/datasets/miracle9to9/files1.

[4] Quan, Q., Wang, J., & Liu, L. (2020). An Effective Convolutional Neural Network for Classifying Red Blood Cells in Malaria Diseases. Interdisciplinary Sciences: Computational Life Sciences, 12(2), 217–225. https://doi.org/10.1007/s12539-020-00367-7.

[5] White, N. J. (2018). Anaemia and malaria. Malaria Journal, 17(1). https://doi.org/10.1186/s12936-018-2509-9.

[6] Mbanefo, A., & Kumar, N. (2020). Evaluation of Malaria Diagnostic Methods as a Key for Successful Control and Elimination Programs. Tropical Medicine and Infectious Disease, 5(2), 102. https://doi.org/10.3390/tropicalmed5020102.

[7] Singh, P., Raj, P., & Namboodiri, V. P. (2020). EDS pooling layer. Image and Vision Computing, 103923. https://doi.org/10.1016/j.imavis.2020.103923.

[8] Ricardo, B. R., Antonio, M. G., & Rodellar, J. (2020, abril 8). Estandarización de métricas de rendimiento para clasificadores Machine y Deep Learning. Revista Iberica de Sistemas e Tecnologias de Informacao, 184–196. https://www.researchgate.net/publication/342009715_Estandarizacion_de_metricas_de_rendimiento_para_clasificadores_Machine_y_Deep_Learning.

[9] Gonzalez, A. C. L. [AprendeIAconLigdiGonzalez]. (2019, mayo 24). MÉTRICAS DE EVALUACIÓN MODELOS DE CLASIFICACIÓN SCIKIT LEARN — #35 Curso Machine Learning con Python. Youtube. https://www.youtube.com/watch?v=K5PNrX694HQ.

[10] Función de pérdida de entropía cruzada. (2020, noviembre 26). ICHI.PRO. https://ichi.pro/es/funcion-de-perdida-de-entropia-cruzada-267783942726718.