UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Implementación de inteligencia artificial explicable en un modelo de detección de Leishmaniasis cutánea

Trabajo de graduación presentado por Michele Benvenuto Caffaro para optar al grado académico de Licenciado en Ingeniería en Ciencias de la Computación y Tecnologías de la Información

Guatemala,

2022

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Implementación de inteligencia artificial explicable en un modelo de detección de Leishmaniasis cutánea

Trabajo de graduación presentado por Michele Benvenuto Caffaro para optar al grado académico de Licenciado en Ingeniería en Ciencias de la Computación y Tecnologías de la Información

Guatemala,

2022

Vo.B0.:

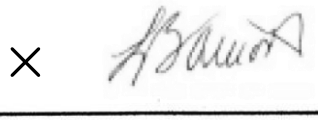
(f) 

Ing. Oscar Iván Robles

Tribunal Examinador:

(f) 

Ing. Oscar Iván Robles

(f) 

Ing. Eddy Omar Castro

(f) 

MSc y MBA Luis Alberto Suriano

Fecha de aprobación: Guatemala, 9 de diciembre de 2022.

Índice

Lista de figuras

Lista de cuadros

Resumen

Desde la concepción de las Redes Neuronales Convolucionales en 1980 para la identificación de caracteres escritos a mano este tipo de inteligencia artificial ha evolucionado hasta el presente en el cual estos modelos son la base para las aplicaciones de detección de imágenes, reconocimiento de imágenes, clasificación de imágenes, análisis de imágenes y vídeo y procesamiento de lenguajes naturales (Bhatt y col., 2021). A medida que el desarrollo de las redes neuronales convoluciones avanza y estas son aplicadas a una mayor cantidad de ámbitos una problemática que surge es la naturaleza de caja negra de este tipo de redes. Caja negra es un termino que se utiliza para un proceso que dado una entrada proporciona una salida sin mostrar retroalimentación sobre el proceso por el cuál se obtuvo este resultado. En el ámbito de la medicina, en el cual se requiere explicación con respecto a las decisiones tomadas, este comportamiento es un impedimento que estas tecnologías se apliquen en una mayor cantidad de problemas a pesar de los beneficios potenciales de su uso. Este proyecto aborda esta problemática en un modelo construido para la identificación de la enfermedad Leishmaniasis cutánea en imágenes desarrollado por estudiantes de la Universidad del Valle de Guatemala.

Abstract

Since the conception of Convolutional Neural Networks in 1980 for the identification of handwritten characters, this type of artificial intelligence has evolved to the present in which these models are the basis for applications of image detection, image recognition, classification of imaging, image and video analysis, and natural language processing (Bhatt y col., 2021). As the development of convolutional neural networks progresses and these are applied to a greater number of situations, a problem that arises is the black box nature of this type of network. Black box is a term used for a process that, given an input, provides an output without showing feedback on the process by which this result was obtained. In the field of medicine, in which explanation is required regarding the decisions made, this behavior is an impediment for these technologies to be applied in a greater number of problems despite the potential benefits of their use. This project addresses this problem in a model built for the identification of cutaneous Leishmaniasis in photographs, this model was developed by students at the Universidad del Valle de Guatemala.

CAPÍTULO 1

Introducción

El presente trabajo tiene como objetivo la introducción de los tres pilares de la explicabilidad de inteligencias artificiales, transparencia, interpretabilidad y explicabilidad a un modelo de detección inteligente de Leishmaniasis cutánea. Esto con la finalidad que este modelo se adapte de mejor manera al ambiente de la medicina en el cual se estará desempeñando. Este proyecto se enfoca en la creación de herramientas que permitan integrar los tres conceptos anteriormente mencionados al modelo de inteligencia artificial con el objetivo de que estos puedan ser posteriormente utilizados por otros equipos de trabajo que deseen trabajar con este modelo en futuros proyectos.

Los procedimientos empleados para introducir los conceptos anteriormente mencionados se dividieron en tres etapas, la primera de estas siendo la recolección de tanto las imágenes utilizadas durante el desarrollo del modelo y el modelo en si con el objetivo de replicar con la mayor precisión posible el comportamiento del modelo. La segunda etapa consistió en la elaboración de los métodos de explicación y por último la validación de estos métodos de explicación por medio de la explicación de imágenes nuevas y evaluando los resultados.

Luego de la implementación y validación de los métodos de explicación se pudo afirmar que los tres pilares de la explicabilidad de Inteligencias Artificiales se pudieron introducir al modelo de manera exitosa proporcionando así resultados con su debida explicación. Además se pudo observar un comportamiento inusual del modelo y se analizan posibles causas y se recomiendan posibles acciones a tomar para mejorar este comportamiento.

CAPÍTULO 2

Antecedentes

2.1. Redes Neuronales Convolucionales en el campo de la medicina

2.1.1. Detección de retinopatía diabética

El método principal para estudiar enfermedades del fondo del ojo como la retinopatía diabética y el glaucoma utiliza Redes Neuronales Convolucionales (Cai y col., 2020). La siguiente tabla muestra las aplicaciones de redes neuronales desde el año 2017 hasta el año 2020, en lo general estas aplicaciones utilizan modelos preentrenados por organizaciones como CaffeNet, GoogleNet y VGG19.

Tarea	Estructura de Red	Resultados
Segmentación de vasos	CNN profunda	ROC $>$ 0.99; precisión de clasificación $>$ 0.97
Segmentación de vasos	CNNs completas	Segmentación de vasos de alto desempeño
Segmentación de vasos	FCN	precisión de 95.33 %; AUC: 0.974
Segmentación de vasos	CNN + CRF multiescalar	Competitividad en la sensibilidad asegurando precisión

Cuadro 1: Resumen de CNNs implementados en la detección de enfermedades del fondo del ojo

(Cai y col., 2020)

2.1.2. Deep Learning based Automatic Detection

En otoño del año 2018 investigadores del Hospital Universitario y Escuela Nacional de Medicina de Seúl desarrollaron un algoritmo para el análisis de radiografías del tórax y detección de posibles crecimientos anormales de células como posibles canceres. El algoritmo nombrado DLAD (Deep Learning based Automatic Detection) fue comparado con la habilidad de diagnóstico de distintos médicos y obtuvo mejores resultados que 17 de los 18 médicos comparados (Wilson y Greenfield, 2019).

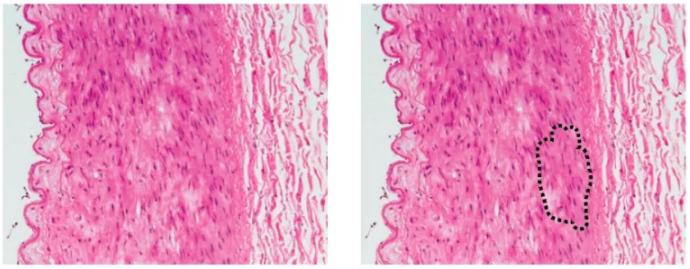


Figura 1: Resultados de diagnóstico del modelo DLAD

(Wilson y Greenfield, 2019)

2.1.3. LYNA

Desarrollado en otoño del año 2018 por investigadores de Google AI Healtcare LYNA (Lymph Node Assistant) tiene como meta identificar tumores metastásicos de cáncer de mama en biopsia de linfas. Este algoritmo logro identificar regiones sospechosas no distinguibles para el ojo humano, al ser probado con dos conjuntos de datos distintos LYNA clasifico correctamente 99 % de las imágenes probadas y al momento que este algoritmo fue utilizado junto con personal médico se alcanzo una reducción del 50 % en el tiempo utilizado para analizar las muestras de tejidos afectados (Wilson y Greenfield, 2019).

2.2. Aplicaciones de xAI

2.2.1. HistoMapr

HistoMapr es un sistema de xAI que se enfoca en satisfacer las necesidades de eficacia y precisión en los diagnósticos patológicos; un objetivo adicional de este sistema es la integración de la patología computacional. Este sistema fue diseñado para reflejar la jerarquía natural y organización espacial encontrada en tejido del pecho y otros órganos (Tosun y col., 2020). HistoMapr se utiliza en el ámbito médico para analizar imágenes médicas, analizar áreas de interés para capturar características relevantes para un diagnóstico, cuantificar estas áreas de interés y luego clasificarlas como potencialmente peligrosas.

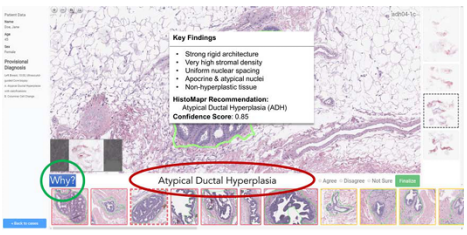


Figura 2: Ejemplo de información proporcionada por HistoMapr

(Tosun y col., 2020)

2.2.2. xAI en análisis forense

Los expertos de seguridad informática manejan una gran cantidad de información que se debe analizar por analistas forenses. Varios autores han planteado técnicas de xAI para reducir la cantidad de trabajo enfocado al manejo de datos y poder enfocar el trabajo en eventos de mayor importancia. Wang et al. en su trabajo (Wang y col., 2020) propone el uso de SHAP para facilitar el reconocimiento de características únicas de distintos ataques de intrusión.

Nadeem et al. (Nadeem y col., 2021) propone la utilización de un paradigma basado en grafos para enseñar estrategias utilizadas por atacantes aprendidas durante ataques de intrusión. Este paradigma permite que los analistas puedan decidir que alertas investigar seleccionando el grafo deseado. Estas explicaciones utilizan un modelo de autómata basado en sufijos para distinguir entre alertas similares pero en contextos distintos

CAPÍTULO 3

Justificación

La Leishmaniasis cutánea es una enfermedad endémica en regiones tropicales y subtropicales de 98 países, lo cual la clasifica como la sexta enfermedad tropical con mayor importancia. Según el Centro de Estudios de Salud de la Universidad de Guatemala, la leishmaniasis cutánea es la forma clínica más frecuente en Guatemala, observándose con mayor frecuencia en los departamentos de El Petén, Alta Verapaz, Quiché, Izabal, Baja Verapaz y Huehuetenango. El Ministerio de Salud Pública y Asistencia Social (MISPAS) indica que la identificación temprana de los casos al igual que el diagnóstico y tratamiento es vital para el tratamiento de esta enfermedad, lastimosamente las condiciones de los departamentos mencionados anteriormente dificultan la instalación de clínicas permanentes y por lo tanto las clínicas móviles tienden a ser los centros de diagnóstico por elección del personal de salud. Una de las dificultades que presentan las clínicas móviles es el acceso limitado de equipo de diagnóstico.

La implementación de herramientas de inteligencia artificial tiene gran potencial en el ámbito de la medicina; Thomas Davenport y Ravi Kalakota en su trabajo "The potential of artificial intelligence in healthcare" mencionan aplicaciones de modelos de inteligencia artificial en el tema del diagnóstico de enfermedades. Una de estas aplicaciones es el modelo Watson creado por IBM que ha mostrado potencial en el área de medicina de precisión y el diagnóstico y tratamiento de cáncer(Davenport y Kalakota, 2019). Un reto que presenta la utilización de AI en el campo de la medicina es la naturaleza de caja negra que estos presentan en un ambiente que requiere que las decisiones a tomar puedan ser justificadas y explicadas de manera clara. Wojciech Samek y Klaus-Robert Müller en su libro "Towards Explainable Artificial Intelligence" menciona que esta naturaleza de caja negra de los modelos de Deep Learning y Machine Learning se debe a la estructura anidada y no lineal de estos modelos. El no proveer información sobre como exactamente se realizan las predicciones que muestra el modelo los hace inaceptables para posibles áreas de aplicación como es la medicina.

Este proyecto tiene como meta principal introducir los conceptos de inteligencia artificial explicable (xAI) en un agente inteligente de detección de Leishmaniasis construido por

miembros de la Universidad del Valle de Guatemala , con el objetivo de proveer la explicabilidad necesaria para que el modelo se pueda aplicar de manera apropiada en el ambiente de la medicina.

CAPÍTULO 4

Objetivos

4.1. Objetivo general

Aplicar los tres principios de Inteligencia Artificial Explicable a un modelo de detección inteligente de Leishmaniasis para que este se adapte de mejor manera al ambiente de la medicina, por medio de la implementación de herramientas que permitan la explicación de las distintas etapas con las cuales el modelo obtiene resultados y como el modelo toma decisiones.

4.2. Objetivos específicos

- Introducir el principio de transparencia de Inteligencia Artificial Explicable en la implementación de un agente inteligente de detección de Leishmaniasis
- Introducir el principio de explicabilidad de Inteligencia Artificial Explicable en la implementación de un agente inteligente de detección de Leishmaniasis. Siguiendo los conceptos de la explicabilidad requerida en el ámbito de la medicina.
- Introducir el principio de interpretabilidad de Inteligencias Artificiales Explicables en la implementación de un agente inteligente de detección de Leishmaniasis.
- Identificar comportamientos del modelo que puedan llegar a ser anómalos o no deseados. De encontrase comportamientos indeseados, recomendar posibles acciones de ciencias de datos y procesamiento de datos que puedan combatir estos comportamientos.

CAPÍTULO 5

Alcance

El alcance de este proyecto es la generación de distintas herramientas que proporcionen del proceso de toma de decisiones del modelo con el objetivo de clarificar el proceso de toma de decisiones por parte del modelo. A la fecha de elaboración del proyecto los posibles usos futuros del modelo son inciertos, por lo tanto las herramientas realizadas se enfocan en ser lo más generales posibles explicando las dos fases de un modelo, el entrenamiento y la toma de decisiones post entrenamiento.

Este proyecto no se enfoca en la mejora del comportamiento del modelo sino en la explicación del mismo, por lo tanto, si se diera el caso que durante la generación de las explicaciones del proceso de toma de decisiones del modelo se identifica cualquier tipo de comportamiento anómalo en la toma de decisiones del modelo, este se documentara y en la sección de recomendaciones se plantearan posibles razones de este comportamiento y posibles maneras de atacar este comportamiento anómalo.

CAPÍTULO 6

Marco teórico

6.1. Leishmaniasis cutánea

La enfermedad Leishmaniasis cutánea es la variante más común de Leishmaniasis y se caracteriza por lesiones y/o ulceras en las áreas afectadas (Organización Mundial de la Salud, 2022). Esta enfermedad se contagia por medio de parásitos de Leishmania que son transmitidos por medio de picaduras de flemotomos femeninos infectados por los parásitos(Organización Mundial de la Salud, 2022).

Según estudios realizados por la Organización Mundial de la Salud existen los siguientes factores de riesgo:

- Condiciones socioeconómica: Condiciones domesticas y sanitarias pobres pueden llegar a aumentan los sitios de apareo y reposo de flemotomos y su acceso a personas. Los flemotodos son atraídos por grandes grupos de personas en espacios reducidos.
- Malnutrición: Dietas bajas en proteínas, hierro, vitamina A y zinc incrementan el riesgo a que una infección progrese a enfermedad completa.
- Movilidad de la población: La migración y movimiento de población no inmune a áreas con ciclos de transición existentes.
- Cambios en el medio ambiente: La incidencia de Leishmaniasis puede ser afectada por cambios en urbanización e incursiones humanas en áreas forestales.
- Cambios climáticos: La Leishmaniasis es sensible al clima ya que afecta su epidelomogía de las siguientes maneras: cambios en temperaturas, lluvias y húmeda tienen efecto en los huéspedes de los parásitos alterando su distribución, pequeños cambios en temperatura pueden tener efecto en el ciclo de desarrollo de parásitos en los flemotomos permitiendo la transmisión de los parásitos en regiones no endémicas.

6.1.1. Leishmaniasis cutánea en Guatemala

En Guatemala las formas clínicas de leishmaniasis más comunes son la cutánea y la visceral siendo la forma cutánea la que tiene presenta la mayor cantidad de casos (581) según estudios del Ministerio de Salud Pública y Asistencia Social (Chavez, 2016). Estos estudios identificaron que esta enfermedad es endémica en 5 departamentos del país que son: El Petén, Alta Verapáz, Izabal, El Quiché y Huehuetenango.

6.2. Machine Learning

Machine Learning (ML) es una categoría de Inteligencia Artificial que se enfoca en el uso de datos y algoritmos para imitar el proceso de aprendizaje humano. Durante los últimos años aplicaciones de ML se han implementado en problemas reales de alta complejidad y han tenido altos rendimientos en distintos problemas en áreas como: filtrado de spam, detección de fraude en redes sociales, compraventa de acciones, reconocimiento facial y de formas, diagnóstico médico entre otros (Alzubi y col., 2018). Dependiendo de la categoría del problema a resolver se pueden aplicar distintos enfoques de machine learning. Las categorías de problemas que se pueden resolver con un acercamiento de machine learning son:

- Problemas de clasificación: En este tipo de problemas la salida puede ser un valor dentro de un conjunto de clases predeterminada, por ejemplo si o no. Dependiendo de la cantidad de clases puede ser un problema de clasificación binario o multi-clase
- Problemas de detección de anomalías: Estos problemas consisten en analizar un patrón y detectar cambios o anomalías en este patrón. Por ejemplo, las compañías de tarjetas de crédito utilizan ML para detección de anomalías para encontrar y advertir de posibles transacciones fraudulentas.
- Problemas de regresión: Problemas que manejan información numérica continua, en estos problemas se hacen preguntan cosas como "¿Cuánto?"
- Problemas de clusters: Este tipo de problemas consiste en analizar los datos proporcionados para poder realizar clases o clusters. Luego de entrenar el algoritmo este puede clasificar nuevas observaciones en uno de los clusters
- Problemas de refuerzo: Algoritmos reforzados son generalmente utilizados cuando el sistema debe tomar decisiones basado en experiencia previa. El algoritmo aprende el comportamiento esperado por medio de prueba y error y dependiendo y usando los conceptos de recompensas y sanciones el agente logra aprender cómo realizar una tarea sin especificar cómo se debe realizar.

ML se utiliza para resolver problemas de distintas naturalezas que el algoritmo debe aprender, a pesar de esto los modelos de machine learning se pueden definir con 6 tareas principales que se mencionan a continuación:

- Recolección y preparación de los datos: Esta tarea consiste en reunir y preparar los datos a un formato que permita ser pasado como ingreso al algoritmo de ML.

- Selección de características: Los datos obtenidos de la tarea anterior pueden llegar a tener varias características algunas de estas no relevantes para el proceso de aprendizaje. En esta tarea las características innecesarias deben ser removidas
- Elección de algoritmos: No todos los algoritmos son aptos para todas las tareas. Dependiendo de la naturaleza del problema existen algoritmos de ML más adecuados.
- Selección de modelos y parámetros: Consiste en la configuración inicial de parámetros y ajustes que requieren los modelos
- Entrenamiento: Luego de la etapa anterior el modelo debe ser entrenado con parte de la data como data de entrenamiento
- Evaluación de resultados: Antes de poder implementar el modelo en situaciones reales se debe probar el modelo con data que no ha visto para analizar su comportamiento.

Dependiendo de cómo se entrene el algoritmo o de la naturaleza de la información que se tiene a disposición durante el entrenamiento los algoritmos de ML se pueden clasificar en una de las siguientes categorías:

- Aprendizaje supervisado
- Aprendizaje sin supervisar
- Aprendizaje reforzado
- Aprendizaje evolutivo
- Aprendizaje semi-supervisado
- Aprendizaje de conjunto
- Redes neuronales artificiales
- Aprendizaje basado en instancias
- Algoritmos de reducción de dimensionalidad
- Aprendizaje híbrido

6.3. Redes Neuronales

Las Redes Neuronales son un subconjunto del aprendizaje de máquina. Las redes neuronales están conformadas por capas de neuronas. Estas capas pueden ser, una capa de inputs, capas escondidas y una capa de outputs. Cada neurona está conectada a otra neurona y tiene un respectivo peso y límite. Si la salida de una neurona es mayor a límite de la misma ésta envía información a la siguiente capa de la red, de lo contrario no se envía información (IBM Cloud Education, 2021b).

6.4. Tipos de Redes Neuronales

Las Redes Neuronales se pueden clasificar en varios tipos dependiendo del propósito de la misma, las tres Redes Neuronales comúnmente utilizadas se presentan a continuación

- FeedFoward Neural Networks: Este tipo de Redes Neuronales se caracteriza por la utilización de neuronas sigmoideas como los nodos de sus capas, este tipo de redes son la base para temas como, visión de computadoras, procesamiento de lenguajes naturales y son la base para otros tipos de redes neuronales (IBM Cloud Education, 2021b)
- Redes Neuronales Convolucionales: Este tipo de Redes Neuronales se caracteriza por su utilización en problemas de reconocimiento de imágenes, reconocimiento de patrones y visión de computadoras. Debido a que son el punto de interés de este proyecto, estas se explicaran a mayor profundidad en la siguiente sección de este trabajo.
- Redes Neuronales Recurrentes: Este tipo de Redes Neuronales se utilizan con series de tiempo y se caracterizan por ser utilizadas en problemas de predicciones de eventos futuros. En su estructura este tipo de redes se caracteriza por sus ciclos de retroalimentación(IBM Cloud Education, 2021b).

6.5. Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales (CNN) se distinguen de otras redes neuronales tanto por el tipo de capas que conforman la red como por su mejor desempeño en tareas relacionadas con la identificación de imágenes y sonidos(O'Shea y Nash, 2015). El diseño y la arquitectura de una CNN se basa en el supuesto que los inputs a esta serán imágenes y por lo tanto los mismos se adaptan a manejar este tipo especifico de datos. Una de las mayores distinciones de las CNNs al momento de compararlas con otro tipo de redes neuronales es que las neuronas que componen las capas se organizan en tres dimensiones dos de estas relacionadas con las características físicas del input (alto y ancho de la imagen) y una tercera dimensión, la profundidad, que refiere a las posibles clases objetivo (O'Shea y Nash, 2015) .Existen tres tipos de capas principales que componen una Red Neural Convolucional estas son:

- Capa Convolucional
- Capa de Pooling
- Capa Fully-Connected

6.5.1. Capa Convolucional

Las capas convolucionales son el bloque principal de las CNNs. Estas capas requieren los siguientes elementos: datos de ingreso, un kernel o filtro y un mapa de características.

Los kernels son matrices de pequeñas dimensionales que contienen pesos para activar este kernel, los pesos de un kernel son modificados al momento de realizar el entrenamiento de un modelo (O'Shea y Nash, 2015) el proceso de entrenamiento de un modelo se explica a mayor profundidad en la sección 6.5.4. El kernel recorre los campos receptores del input revisando si existe o no una característica, a este proceso se le conoce como convolución.

Las imágenes que se utilizan como input consisten en una matriz de píxeles en tres dimensiones cada dimensión representando los valores RGB del píxel. El kernel se le aplica a una sección de la imagen por medio de un producto punto, este resultado es guardado en un arreglo que será la salida de este proceso (O'Shea y Nash, 2015). Luego el filtro es movido por una cantidad de píxeles llamada el "paso" para volver a aplicar el kernel a otra sección de la imagen. El arreglo resultante de este proceso es conocido como el mapa de características, mapa de activación o característica convolucionada. A continuación se puede observar un diagrama del proceso:

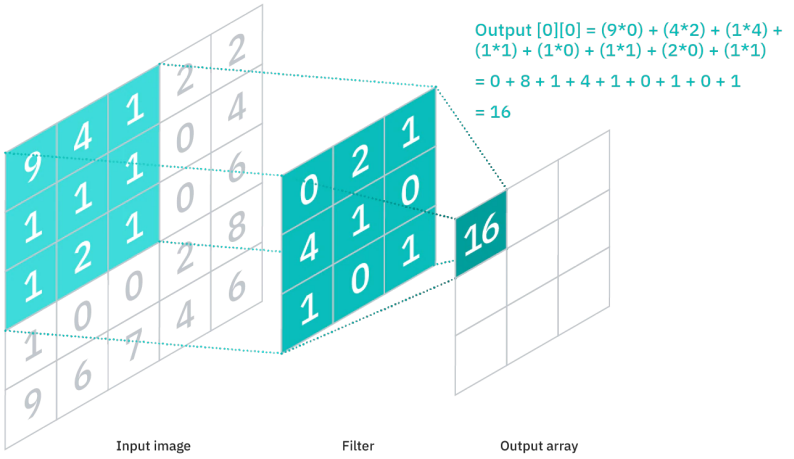


Figura 3: Proceso de aplicación de un filtro en una capa convolucional

(IBM Cloud Education, 2021a)

Como se puede observar en el diagrama los pesos del kernel se mantienen a lo largo de la imagen. Existen tres parámetros de la red (conocidos como hiperparametros) que se deben definir antes de entrenar la red, estos son:

- Número de filtros: El número de filtros que se aplican afecta la profundidad del resultado, N filtros generarán N mapas de características distintos también conocido como una profundidad de N
- El paso: El número de píxeles que el kernel se mueve sobre la matriz
- Zero-padding: normalmente utilizado cuando el filtro no se ajusta al input. Este parámetro asigna todos los valores fuera del input a zero, existen tres tipos de padding:
    - Valid Padding: En este padding la última convolución se elimina si las dimensiones no se alinean

- Same Padding: Este tipo de padding se asegura que la capa de output tenga el mismo tamaño que la capa de input
- Full Padding: Este tipo de padding incrementa el tamaño del output agregando ceros en los bordes del input.

Luego de cada operación convolucional, la CNN aplica una Unidad Linear Rectificada (ReLU por sus siglas en inglés) al mapa de características, con el objetivo de introducir no linealidad al modelo (IBM Cloud Education, 2021a). Una capa convolucional puede ser seguida de otra capa convolucional en la red, esto incorpora jerarquía en la CNN haciendo que entre más profunda se encuentre la capa convolucional mayor detalle revise en la imagen.

6.5.2. Pooling layer

Las capas pooling tienen como objetivo reducir la dimensionalidad del input que reciben. Similar a las capas convolucionales las capas de pooling realizan una operación a todos el input, estas se diferencian en que el kernel aplica una función de agregación a los valores de los campos receptivos y así se genera el arreglo resultante (O'Shea y Nash, 2015). Existen dos tipos de pooling:

- Max Pooling: A medida que el filtro se mueve a través del input, se escoge el valor mayor para enviar al arreglo de salida
- Average Pooling; A medida que el filtro se mueve a través del input calcula el promedio de los valores del campo receptivo para guardar en el arreglo de salida

6.5.3. Fully-Connected Layer

En este tipo de capas cada neurona en la capa de salida está conectada directamente a una neurona en la capa anterior. Esta capa realiza la clasificación del input basado en las características extraídas por las capas y filtros anteriores. Las capas fully connected utilizan una función de activación softmax para clasificar los inputs de manera apropiada en una escala del 0 al 1(Norvig y Russell, 2022).

6.5.4. Entrenando una Red Neural Convolucional

El proceso de entrenar una CNN consiste en la modificación de los parámetros y pesos de una red neuronal con el objetivo de minimizar la función de perdida sobre un conjunto de datos de entrenamiento (Norvig y Russell, 2022). El algoritmo mayormente utilizado para optimizar el proceso de entrenamiento es el descenso de gradiente estocástico (SGD por sus siglas en inglés).

El algoritmo de descenso de gradiente consiste en, empezando en un punto arbitrario en un espacio de pesos (en el caso de una Red Neuronal un kernel) se calcula una estimación del gradiente y se procede a movernos una pequeña dirección hacia la dirección de mayor cambio

negativo(Norvig y Russell, 2022). Este proceso se repite hasta que llegamos a un punto de conversión en el espacio de pesos, contextualizando este proceso a las redes neuronales este punto de conversión son los pesos del kernel finales que el modelo utilizara para realizar predicciones.

$$w_i \leftarrow w_i - \alpha \frac{\partial}{\partial w_i}$$

Figura 4: Ecuación para el cálculo de nuevos pesos con el método de descenso de gradiente

(Norvig y Russell, 2022)

En la Figura 4 el termino α refiere al rito me aprendizaje que tendrá el proceso de descenso de gradiente al momento de entrenar el modelo.

El descenso de gradiente estocástico es una variante del descenso de gradiente que, al momento de analizar la data de entrenamiento, en cada iteración del proceso de entrenamiento selecciona aleatoriamente pequeños lotes de la información total, esto con el objetivo de facilitar la cantidad de cómputos necesarios en cada etapa del proceso de entrenamiento(Norvig y Russell, 2022).

6.6. Inteligencia Artificial Explicable

Inteligencia Artificial Explicable es un conjunto de procesos y métodos que permiten a los usuarios comprender y confiar en los resultados obtenidos y presentados por un modelo de machine learning. Para realizar esta tarea los sistemas de XAI se basan en los siguientes principios: El sistema debe ser capaz de explicar sus capacidades, sus conocimientos, las acciones que ha realizado, que está realizando y que acciones va a realizar (Gunning y col., 2019).

Las explicaciones proporcionadas por el sistema se encuentran ligadas a un contexto que está definido por las tareas, habilidades y expectativas del usuario del sistema de AI (Gunning y col., 2019). Es por esto que los conceptos de interpretabilidad y explicabilidad están ligados al dominio en el que se trabaja y no pueden ser definidos sin este.

Existen dos tipos de explicaciones que pueden ser proporcionadas por un sistema de XAI, completas o parciales. Los modelos que son completamente interpretables proporcionan explicaciones completas y transparentes, mientras que los modelos parcialmente interpretables revelan información importante del proceso de razonamiento del modelo (Gunning y col., 2019). Todo modelo interpretable se rige por las restricciones de interpretabilidad de su dominio, mientras que cajas negras o modelos sin restricciones no las obedecen.

Durante la definición de modelos de XAI se asume que la explicación se proporcionará a un usuario que depende de la decisión tomada por el modelo, este usuario puede tener distinta naturaleza o necesidad. Cada posible grupo que interactúe con nuestro modelo puede tener una preferencia a que procesos del modelo requieren de explicación, una explicación efectiva toma en cuenta el usuario objetivo del sistema(Gunning y col., 2019).

6.7. Métodos de xAI

6.7.1. Explicaciones utilizando sustitutos

Este tipo de explicaciones consiste en aproximar las predicciones realizadas por modelos utilizando una función sustituto que sea más interpretable que el modelo original. Una de las técnicas más populares de este método de explicación son las Explicaciones Locales Agnósticas del Modelo (LIME por sus siglas en ingles). El método LIME utiliza muestras de la entrada de interés, evalúa el modelo de inteligencia artificial en estas muestras y ajusta la función sustituto para que esta se aproxime de mejor manera la función de interés. Una ventaja de este tipo de explicaciones es que debido a que son agnósticas al modelo se pueden aplicar a cualquier tipo de modelo sin importar su arquitectura (Samek y Müller, 2019). Una desventaja de la aplicación de este tipo de explicaciones es que en modelos con arquitecturas muy complicadas el computo para las aproximaciones puede llegar a tomar una cantidad considerada de tiempo para computar las explicaciones de una predicción.

6.7.2. Explicaciones utilizando perturbaciones locales

Este tipo de explicaciones se enfocan en la reacción que tiene el modelo a cambios locales, este tipo de explicaciones incluye la utilización de los gradientes del modelo tanto como los acercamientos que basados en perturbaciones de la entrada de los modelos (Samek y Müller, 2019).

Las explicaciones basadas en los gradientes obtenidos de las funciones utilizadas por el modelo han sido utilizados a través de la historia de los modelos de aprendizaje de maquina. El análisis de sensitivas es un ejemplo de este tipo de explicaciones, esta explicación se enfoca en la explicación en los posibles cambios de la predicción en lugar de la predicción en si, analizando como cambia la predicción dependiendo de pequeños cambios a la entrada que se le proporciona al modelo.

Las explicaciones basadas en perturbaciones prueban la reacción del modelo a perturbaciones locales más generales. El método de sensibilidad de oclusión permite medir la importancia de distintas secciones del input por medio de ocultar secciones del mismo. El análisis de diferencias de predicciones utiliza muestreo condicional con los píxeles cercanos a un mapa de activación con el objetivo de remover información a la entrada del modelo y medir los cambios a las predicciones (Samek y Müller, 2019). Una problemática que presentan este tipo de explicaciones es que el modelo y las explicaciones deben medir todos los impactos de las perturbaciones en el modelo haciendo que estas explicaciones sean demandantes en recursos computacionales.

6.7.3. Explicaciones basadas en propagaciones

Las explicaciones basadas en propagaciones se caracterizan por la utilización de la estructura interna de los modelos para integrar al proceso de las predicciones las explicaciones necesarias (Samek y Müller, 2019). Los métodos mayormente utilizados son la Deconvolución

y la Retropropagación guiada, estos métodos se enfocan en identificar patrones dentro de la entrada del modelo para identificar patrones que tienen efecto positivos en las predicciones del modelo.

6.8. Objetivos de xAI

La literatura no define concretamente el propósito principal de un modelo de inteligencia artificial explicable, pero si definen distintos objetivos a aspirar dependiendo del contexto en el cual se esta aplicando el modelo. Estos objetivos son:

Objetivo de explicabilidad	Audiencia objetivo
Integridad	Expertos del dominio, usuarios del modelo que se ven afectados por la decisiones tomadas
Casualidad	Expertos del dominio, gerentes y miembros de la junta ejecutiva, entidades reguladoras
Transferibilidad	Expertos del dominio, cientificos de datos
Informatividad	Todos
Confianza	Expertos del dominio, desarrolladores, gerentes, entidades reguladoras
Rectitud	Usuarios del modelos que se ven afectado por las decisiones tomadas, entidades reguladoras
Accesibilidad	Product Owners, gerentes , usuarios del modelo que se ven afectados por las decisiones tomadas
Interactividad	Expertos del dominio, usuarios del modelo que se ven afectados por las decisiones tomadas
Conciencia de la privacidad	Usuarios del modelo que se ven afectados por las decisiones tomadas, entidades reguladoras

Cuadro 2: Objetivos de explicabilidad y usuario al cual se enfoca

(Barredo Arrieta y col., 2020)

- Integridad: Varios autores coinciden con afirmar que este es el objetivo principal de un modelo de AI explicable (Ribeiro y col., 2016). La integridad de un modelo refiere a la confianza que se tiene que un modelo actuara de la manera esperada al momento de enfrentar un problema dado (Barredo Arrieta y col., 2020).

- Casualidad: Refiere a la habilidad de un modelo de encontrar la casualidad dentro de variables. Varios autores argumentan que los modelos explicables tienen la capacidad de facilitar la búsqueda de relaciones que se puedan probar con mayor fuerza parar encontrar enlaces de mayor fuerza entre variables. Los modelos de aprendizaje de maquina encuentran correlaciones entre la data utilizada para su entrenamiento y por lo tanto puede llegar a no ser suficiente evidencia para mostrar relaciones causa-efecto (Rani y col., 2006). A pesar de esto, causalidad involucra la correlación, así que

modelos explicables pueden llegar a validar los resultados de técnicas para la inferencia de casualidad o probar intuición de posibles relaciones de causalidad.

- Transferibilidad: La explicabilidad de modelos defiende la transeferibilidad, facilitando la tarea de aclara las barreras de las características que pueden afectar al modelo, esto permite una mejor comprensión y implementación del modelo. Además, comprender la funcionalidad de un modelo permite la reutilización del conocimiento obtenido por este modelo para problemas similares.

- Informatividad: El propósito principal de los modelos de aprendizaje de maquina es su utilización para el apoyo en toma de decisiones. Es importante aclarar que el problema resuelto por un modelo no es necesariamente igual al problema que debe resolver un humano. Por lo tanto los modelos deben proporcionar la mayor cantidad de información posible para que la toma de decisión no se vea afectada por ideas equivocadas (Caruana y col., 2015). Con este objetivo en mente, los modelos de aprendizaje de maquina explicables deben dar información sobre el problema.

- Confianza: La confianza de un modelo siempre debe ser analizada en ambientes que requieran fiabilidad. Un modelo explicable debe tener información sobre la confianza de sus predicciones (Barredo Arrieta y col., 2020).

- Rectitud: Desde un punto de vista social, la explicabilidad se puede definir como la capacidad que tiene un modelo de obtener y garantizar rectitud. Un modelo explicable debe mostrar las relaciones que afectan su resultado, permitiendo el análisis ético del modelo (Goodman y Flaxman, 2017). Igualmente, un modelo de XAI debe presentar posibles parcialidades en la información que se le ha mostrado para su entrenamiento (Hendricks y col., 2018).

- Accesibilidad: Algunos autores sugieren que la explicabilidad permite que el usuario final del modelo de aprendizaje de maquina pueda tener efecto en el proceso de mejorar y desarrollar dicho modelo (Chander y col., 2017). Los modelos explicables facilitan el lidiar con algoritmos de aprendizaje de maquina a personas no especializadas en temas de inteligencia artificial.

- Interactividad: Este punto se enfoca en la facilidad con la cual el usuario final que utiliza el modelo puede interactuar con dicho modelo para asegurar mejores resultados (Langley y col., 2017).

- Conciencia de la Privacidad: Uno de los resultados de la explicabilidad de algoritmos de Aprendizaje de Maquina es la habilidad de la revisión de privacidad. No poder comprender los patrones capturados por un modelo puede llegar a representar una violación a la privacidad (Barredo Arrieta y col., 2020). La habilidad de explicar el funcionamiento interno de un modelo permite analizar que un modelo no viole la privacidad requerida del ámbito en el cual este es utilizado.

CAPÍTULO 7

Metodología

7.1. Recolección de imágenes

Las imágenes utilizadas para la elaboración de este proyecto se obtuvieron de dos fuentes principales, las bases de datos del Centro de estudios en Salud de la Universidad del Valle de Guatemala y las imágenes públicas recolectadas por el grupo que desarrollo el modelo de detección.

Debido a que durante el desarrollo de este proyecto se requiere el manejo de información sensible de pacientes afectados por Leishmaiasis cutánea, con el objetivo de mantener la confidencialidad de esta información, con respecto a las imágenes provenientes del Centro de estudios en Salud de la Universidad el Valle de Guatemala ,para el desarrollo de este proyecto se tomaron las siguientes mediadas:

- Las imágenes que se utilizarán durante el proyecto fueron completamente desvinculadas de cualquier tipo de identificador personal de los pacientes de quienes proceden dichas imágenes.
- Durante el desarrollo del proyecto no se tuvo acceso a la base de datos de la Universidad del Valle de Guatemala que vincula las fotografías con la información de los pacientes.
- Ningún individuo ajeno al desarrollo del proyecto tendrá acceso a las imágenes a utilizar durante el desarrollo del mismo , y los miembros del equipo de trabajo no las compartieron con terceros.
- La confidencialidad requerida para el proyecto está amparada por la firma de un acuerdo de confidencialidad con el Centro de Estudios en Salud de la Universidad del Valle de Guatemala.

- Los miembros del equipo de trabajo del proyecto tomaron un curso de ética en investigación

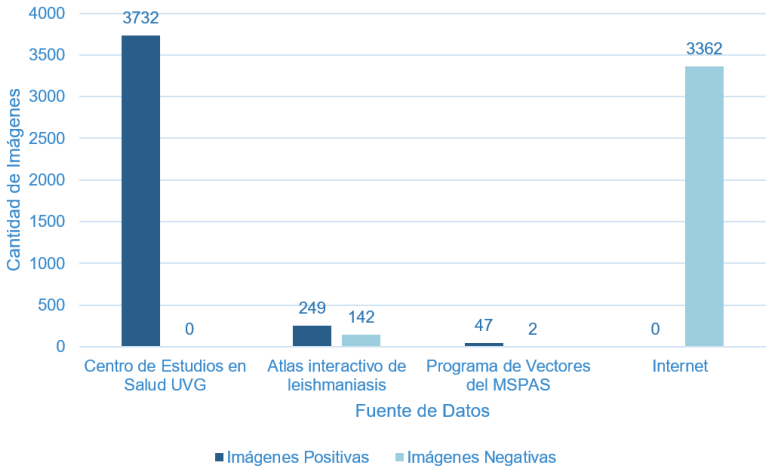


Figura 5: Distribución de los datos utilizados por el modelo por fuente

7.2. Modelo a explicar

El modelo utilizado durante el desarrollo de este proyecto consiste en una Red Neuronal Convolucional cuya estructura se puede observar en el anexo 1. El modelo fue elaborado con el framework de TensorFlow que utiliza python para su funcionamiento, para facilitar la integración de los métodos de explicación al modelo estos también se desarrollaron en Python.

El modelo fue entrenado según las indicaciones del grupo de trabajo que diseño esta red, con una distribución del 60 % de las imágenes utilizadas para el proceso de entrenamiento, 20 % para el proceso de validación y 20 % para el proceso de pruebas. Se utilizaron las mismas configuraciones de parámetros e hiperparámetros de la red, esto con el objetivo de imitar de manera más exacta el comportamiento final de la red neuronal convolucional.

7.3. Métodos de explicación realizados

Con el objetivo de introducir los tres conceptos de explicabilidad al modelo de detección de Leishmaniasis cutánea, se implementaron los métodos de explicación: Gradient-weighted Class Activation Maps, la sensibilidad a oclusión y explicaciones utilizando la libreria LIME. El primero de estos enfocado en la explicación de las capas internas del modelo, el segundo enfocado en la explicación de los aspectos de mayor prioridad al momento de ser analizados por el modelo y el tercero enfocado en la explicación por medio de la utilización de funciones sustituto.

7.3.1. Gradient-weighted Class Activation Maps

El método Gradient-weighted Class Activation Maps (GradCAM) utiliza las información de los gradientes que salen de una capa convolucional de la red para comprender la importancia de cada neurona para una decisión (Selvaraju y col., 2017). Para obtener las secciones de la imagen de mayor importancia para una capa N con relación a una clase objetivo cualquiera, en el caso de este proyecto solo tenemos una clase objetivo (positivo de Leishmaniasis cutánea), se calcula el gradiente del puntaje para la clase de interés. Estos gradientes calculados luego se propagan hacia atrás y con ellos se calcula la media global agrupada para obtener los pesos de importancia de la neurona, estos pesos capturan la importancia de un mapa de características para una clase objetivo (Selvaraju y col., 2017).

Estos pesos luego son alimentados a una función ReLU para encontrar las características que tienen un impacto positivo al momento de calcular la clase de interés, el resultado de este proceso son mapas de calor que indican que mapas de características tienen el mayor impacto al momento de obtener la puntuación para una clase objetivo. El modelo analizado en este proyecto está constituido por 5 capas convolucionales, este método de explicación nos permite adentrarnos a cada una de estas capas y observar el proceso por el cual la red neural analiza distintas secciones de las imágenes para obtener los resultados que luego presenta.

$$L^c_{Grad-CAM} = ReLU(\sum_k \alpha^c_k A^k)$$

Figura 6: Fórmula para obtener los resultados del método de GradCAM

(Selvaraju y col., 2017)

En la Figura 6, el termino α^c_k captura la importancia de un mapa de características k para una clase de interés c. Se realiza una combinación ponderada de los distintos mapas de activación y se realiza una operación $ReLU$ con el objetivo de mostrar únicamente los mapas de activación que tienen efecto positivo en la clase de interés.

7.3.2. Sensibilidad a la oclusión

El método de Sensibilidad a la oclusión consiste en alimentar al modelo de predicciones imágenes que han sido alteradas ocultando pequeñas partes de la misma y luego alimentando estas imágenes alteradas al modelo. Esto se hace con el objetivo de identificar como distintas secciones de la imagen de input afectan la confianza con la cual el modelo realiza sus predicciones. Luego, con los resultados de las imágenes alteradas podemos crear mapas de calor para representar que áreas de la imagen son las de mayor importancia para la toma de decisiones por parte del modelo.

$$\xi(x) = argminL(f, g, \pi_z) + \Omega(g)$$

Figura 7: Fórmula para generar explicaciones con LIME

7.3.3. Explicaciones utilizando LIME

Las explicaciones locales interpretables agnósticas del modelo (LIME por sus siglas en inglés) tienen como objetivo identificar un modelo interpretable basado en una representación interpretable que es localmente fiel al clasificador original (Ribeiro y col., 2016). Este método define una explicación como un modelo $g \in G$ donde G es un conjunto de modelos potencialmente explicables. g actúa sobre la parecencia o ausencia de componentes interpretables, puede suceder que un modelo $g \in G$ no sea lo suficientemente simple para ser interpretable, por lo tanto se define una función $\Omega(g)$ utilizada parar medir la complejidad de una explicación g (Ribeiro y col., 2016). Definiendo el modelo a explicar como f, para modelos de clasificación $f(x)$ representa la probabilidad de que x pertenezca a una de las clases objetivo. Luego definimos $L(f, g, \pi_z)$ esta función nos permite medir la infidelidad de una aproximación g de la función f en la localidad de π_z (Ribeiro y col., 2016). Las explicaciones dadas por lime se obtienen por la siguiente formula:

Esta fórmula tiene como objetivo asegurar la interpretabilidad y la fidelidad local del modelo g por medio de la minimización de tanto $L(f, g, \pi_z)$ como de $\Omega(g)$. Esta función es compatible con tanto diferentes clases G como con funciones de fidelidad L y funciones de complejidad Ω (Ribeiro y col., 2016).

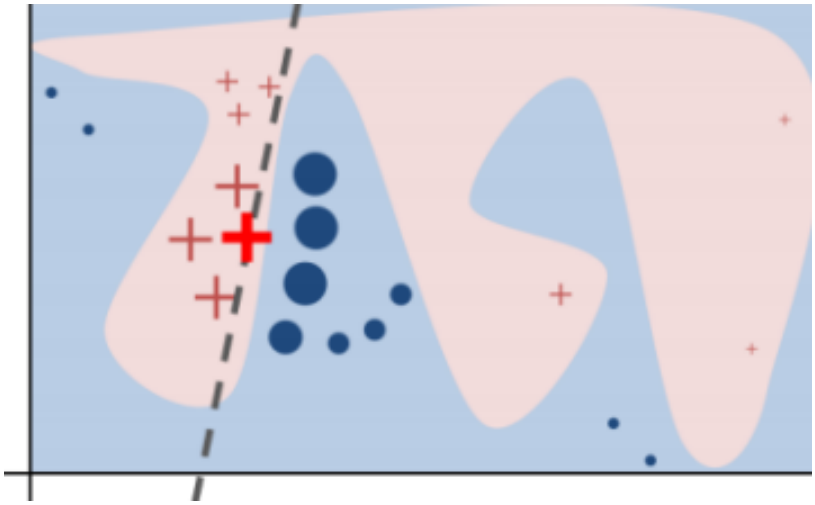


Figura 8: Representación gráfica para explicar intuitivamente el proceso de LIME.

En la figura anterior el modelo a explicar se representa por el fondo azul y rosado. La cruz roja representa la instancia a explicar. Los círculos y cruces representan las muestras utilizadas en el modelo para aproximar g (la línea punteada) que representa la explicación localmente fiel

Para la elaboración de estas explicaciones se utilizo la librería LIME desarrollado basado en el artículo *"Why Should I Trust You?": Explaining the Predictions of Any Classifier.*

Esta librería proporciona herramientas tanto para la generación de datos para alimentar a una función sustituto, como herramientas para calcular la función sustituto para un modelo proporcionado.

7.4. Validación de los métodos de explicación

Los distintos autores citados durante la elaboración de este proyecto concuerdan con el hecho que evaluar el desempeño y la calidad de una explicación generada por un método de xAI tienden a ser abstractos y dependientes del contexto en el cual se aplica el modelo a explicar. Tomando esto en cuenta, las explicaciones realizadas con los métodos mencionados anteriormente se valuaran según su desempeño en la integración de los pilares de xAI. Los métodos de explicación se mostrarán a expertos de salud del Centro de Estudios en Salud de la Universidad del Valle de Guatemala y dependiendo de la información que puedan extraer utilizando estás explicaciones se evaluara la integración de los tres pilares enumerados en los objetivos de este proyecto.

Resultados

En este capítulo se muestran los resultados de aplicar los métodos de explicación previamente mencionados al modelo de detección de Leishmaniasis. La estructura del modelo a analizar se muestra en el anexo 1. Las imágenes que se utilizan para mostrar estos resultados son de origen público encontradas en Internet o en el Atlas interactivo de Leishmaniasis.

7.5. Gradient-weighted Class Activation Maps

Este método nos permite adentrarnos a las distintas capas que conforman la Red Neural Convolucional, como se puede observar en el anexo 1 el modelo está formado por 12 distintas capas, de las cuales cinco son convolucionales y serán las que se analizan con mayor interés ya que estas son las encargadas de identificar características de las imágenes que permitan al modelo identificar la entrada como Leishamaniasis positiva o negativa.

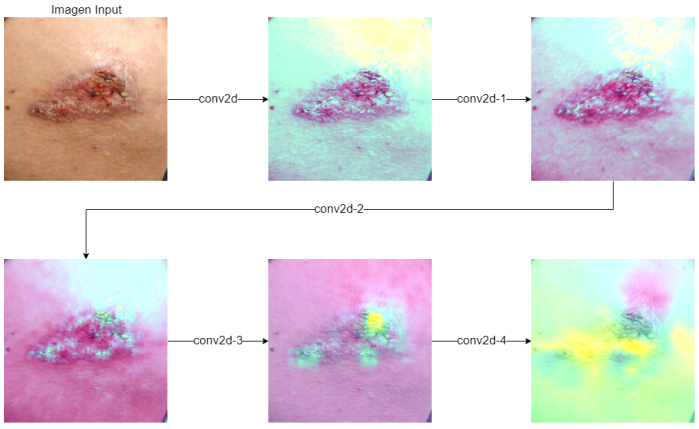


Figura 9: Gradient-weighted Class Activation Maps, caso positivo para Leishmaniasis

Se puede observar tanto en la Figura 9 como la Figura 10 el proceso de toma de decisión

del modelo empezando desde una imagen como la entrada al modelo y las secciones de interés de cada capa. Las explicaciones se muestran en un formato de mapa de calor siendo las secciones marcadas con colores azules las de menor impacto y las marcadas con amarillo las capas que tienen mayor impacto en la salida de cada una de las capas. Como se puede observar en ambas figuras el modelo parece analizar las imágenes empezando desde los alrededores de las lesiones y cada capa se adentra a analizar las secciones interiores de la lesión.

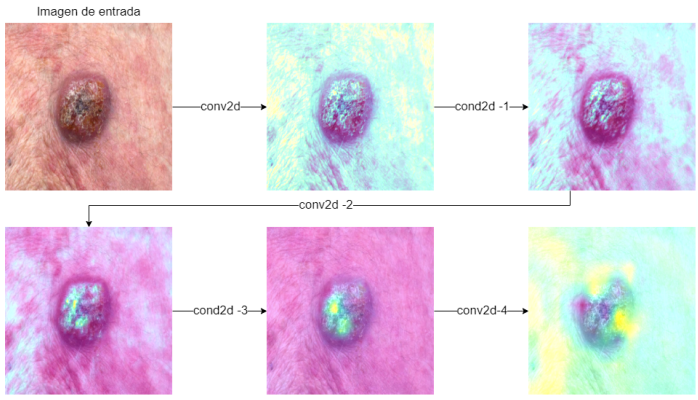


Figura 10: Gradient-weighted Class Activation Maps, caso negativo para Leishmaniasis

7.6. Sensibilidad a la oclusión

Este método nos permite observar las secciones de las imágenes que son críticas al momento momento que el agente inteligente analiza una imagen cualquiera. Como se puede observar en la Figura 11 y en la Figura 12 el resultado de este método de explicación consiste en un mapa de calor, en este las secciones con colores azules representan las áreas de menor importancia cuando el modelo toma una decisión y las secciones resaltadas de amarillo representan secciones de mayor importancia.

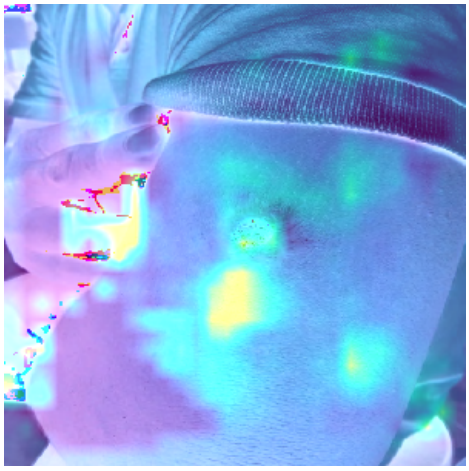


Figura 11: Explicación por medio del método de oclusión, caso positivo para Leishmaniasis

Los resultados con este método de explicación se generaron ocultando iterativamente secciones de la imagen con cuadrados de 7 x 7, estás imágenes se usan luego como entrada para el modelo para que este prediga un valor para la clasificación como positivo para Lesihmaniasis cutánea y dependiendo de los efectos que tiene la modificación de la imagen en este resultado se marcan del color respectivo las distintas secciones en la imagen.

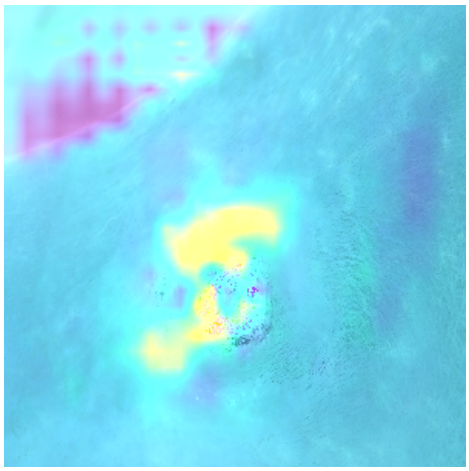


Figura 12: Explicación por medio del método de oclusión, caso negativo para Leishmaniasis

7.7. LIME

Este método de explicación permite calcular secciones que tienen un mayor impacto tanto positivo como negativo en la decisión tomada por el modelo. Los resultados de este método de explicación se pueden observar en las figuras 13 y 14.

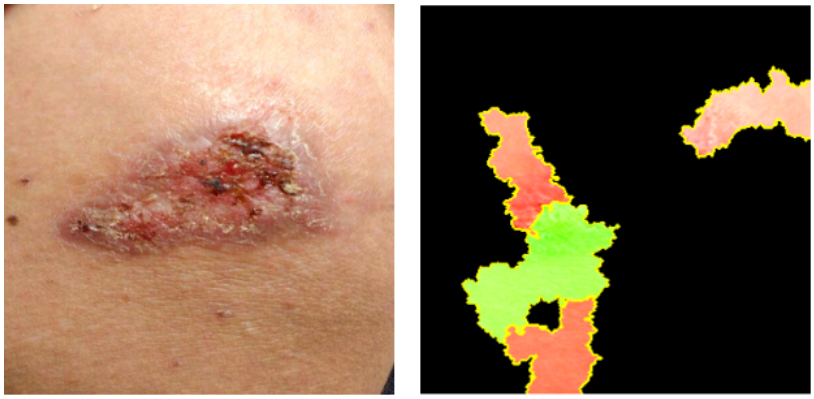


Figura 13: Explicación generada por el método LIME, caso Leishmaniasis positiva

El resultado de este método de explicación consiste en imágenes que resaltan de color verde las secciones de las imágenes que tienen un impacto positivo en la decisión del modelo y resaltan de color rojo las secciones de las imágenes tienen un impacto negativo en la decisión del modelo.

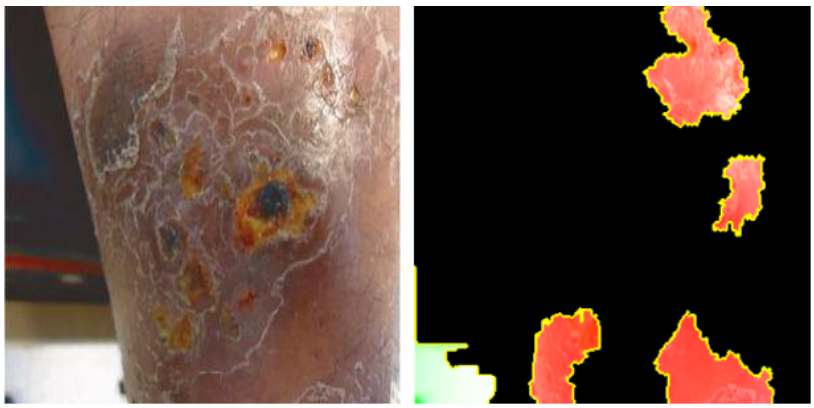


Figura 14: Explicación generada por el método LIME, caso Leishmaniasis negativa

7.8. Validación de las explicaciones con expertos

Para validar que los métodos de explicación fueran realimente útiles para integrar tanto la transparencia, como la interpretabilidad y la explicabilidad en el modelo, las explicaciones fueron presentadas a miembros del Centro de Estudios en Salud de la Universidad del Valle de Guatemala con el objetivo de obtener retroalimentación de los mismos.

En general, la retroalimentación obtenida por los miembros del CES fue positiva afirmando que los métodos de explicación ayudan a descubrir tanto comportamientos esperados como comportamientos inesperados del modelo. Un entrevistado afirmó que las explicaciones permitieron mostrar comportamientos anómalos como el siguiente: "Para poder tomar decisión respecto a si una lesión es de Leishmaniasis cutánea o no, es necesario evaluar algunas características como bordes elevados, centro profundo, etc. En este caso, el modelo está seleccionando áreas que no están sobre la úlcera.". Este comentario dirigido hacia el método de explicación por oclusión.

Al momento de observar las explicaciones generadas por el algoritmo de GradCAM los expertos mencionaron que los diagramas provenientes de este proceso tendía a ser las explicaciones más valiosas debido a la cantidad de información que muestra sobre el proceso de toma de decisioens del modelo.

Por último, las explicaciones generadas por LIME, al igual que los otros modelos, permitieron la observación de patrones y decisiones erróneas por medio del modelo. El siguiente comentario de los expertos del CES soporta esta observación: Çon este diagrama tengo duda, porque no sé si las dos imágenes serán del mismo tamaño. Pareciera que la zona de impacto positivo está fuera del área de la lesión, lo cual no sería útil". Este comentario deja entender que las explicaciones con LIME pueden llegar a ser algo confusas para los usuarios de las mismas, pero a pesar de esto se puede extraer comportamientos anómalos de los diagramas, como la influencia de áreas ajenas a la lesión a la toma de decisiones del modelo.

CAPÍTULO 8

Análisis de resultados

Durante el desarrollo de los distintos métodos de explicación se pudo observar un comportamiento no deseado en el agente de detección de Leishmaniasis. Este comportamiento consiste en que el modelo reconoce patrones incorrectos al momento de analizar una imagen de entrada. Estos patrones siendo cadenas de caracteres dentro de la imagen y la presencia de grandes secciones de color blanco en la imagen de entrada, este comportamiento se puede evidenciar en la Figura 15 y la Figura 16 que muestran el comportamiento encontrado por el método de explicación de GradCAM y las figuras 17 y 18 muestran validación de este comportamiento con el métodos de LIME, también se puede observar mayor evidencia de este comportamiento en el anexo 2.

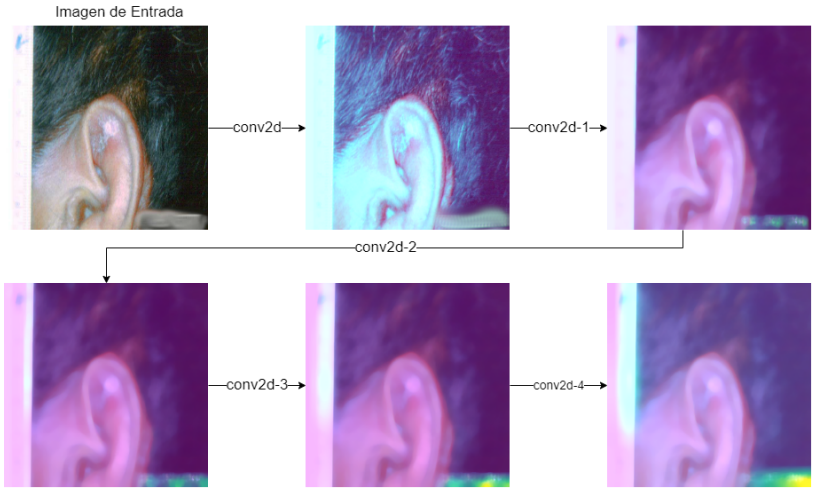


Figura 15: Evidencia del comportamiento anómalo del modelo utilizando el método de GradCAM

Luego de analizar el proceso de entrenamiento y las imágenes que se utilizan para este mismo procedimiento, se puede concluir que este comportamiento puede ser debido a que la mayoría de las imágenes etiquetadas como positivas para la enfermedad de Leishmaniasis cutánea poseen tanto una cadena de caracteres dentro de la imagen y grandes porciones de color blanco o solo uno de estos elementos, mientras que las imágenes etiquetadas como negativas tienden a no poseer ninguno de estos. Esto tiene como resultado un sesgo por parte del modelo, el cual tiende a producir falsos positivos cuando las imágenes que se le proporcionan como entrada contienen algunos de los elementos anteriormente mencionados. Por otro lado, la carencia de estos elementos en la imagen tiende a dirigir al modelo a producir falsos negativos.

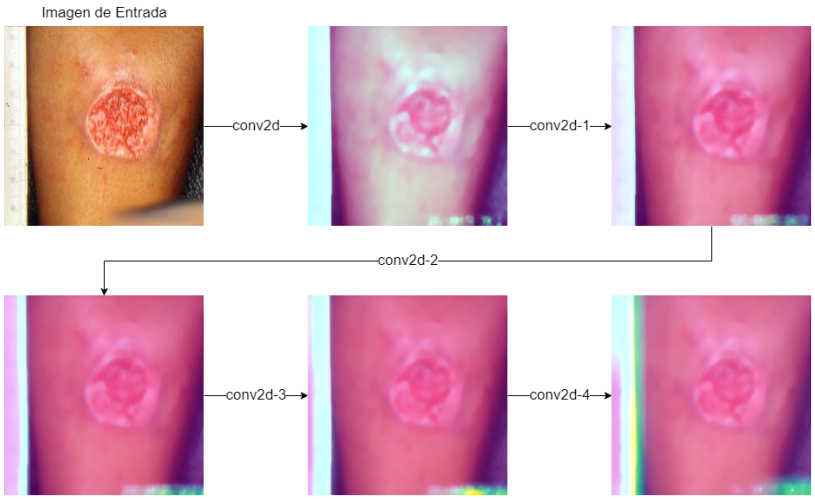


Figura 16: Evidencia del comportamiento anómalo del modelo utilizando el método de GradCAM

A pesar de este comportamiento anómalo, el modelo produce buenas métricas de evaluación ya que tanto el conjunto de datos de evaluación como los datos para las pruebas del modelo poseen estas características. Estas métricas se pueden observar en el anexo 1.

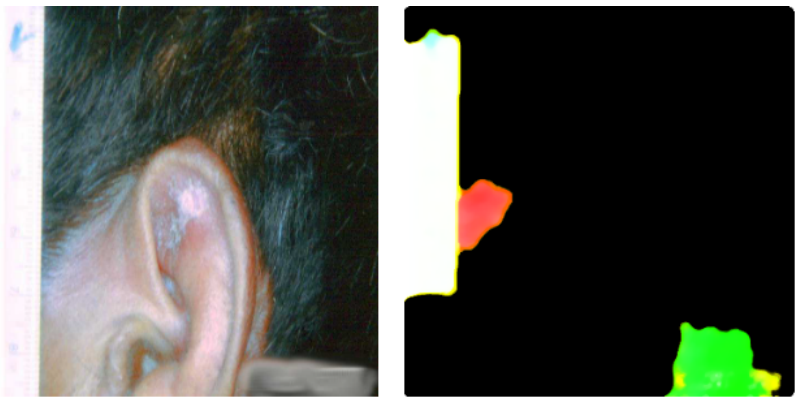


Figura 17: Evidencia de comportamiento anómalo del modelo utilizando el método de LIME

Con respecto a los objetivos del proyecto de introducir los tres pilares de la explicabilidad de inteligencia artificial (interpretabilidad, transparencia y explicabilidad) al agente de

inteligente de detección de Leishmaniasis se puede afirmar que los pilares se introdujeron de manera exitosa en el modelo ya que gracias a la interpertabilidad, transparencia y explicabilidad introducida por los métodos de explicación de Gradient-weighted Class Activation Maps, Sensibilidad a la Oclusión y las explicaciones locales interpretables agnósticas del modelo, se pudieron identificar los comportamientos anómalos anteriormente mencionados.

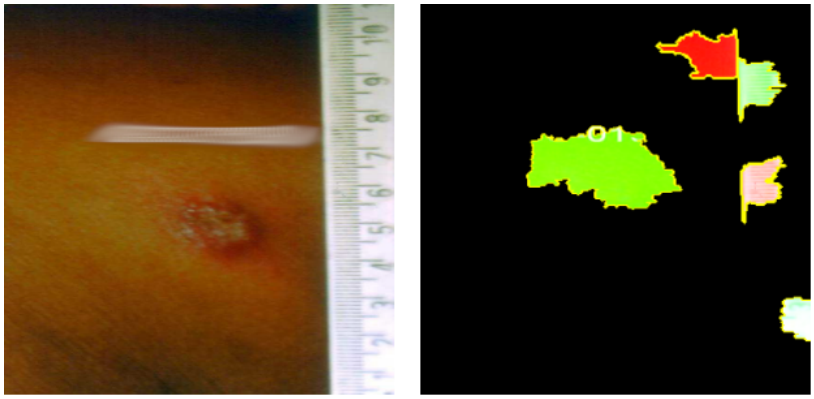


Figura 18: Evidencia de comportamiento anómalo del modelo utilizando el método de LIME

La retroalimentación de los expertos del Centro de Estudios en Salud de la Universidad del Valle de Guatemala confirma tanto la sospecha de posibles comportamientos inadecuados por parte del modelo como la exitosa integración de los pilares de la inteligencia artificial explicable. Gracias a la transparencia, explicabilidad e interpretabilidad integrada por los métodos de explicación realizados, los miembros del CES pudieron identificar tanto comportamientos adecuados como comportamientos inadecuados en la identificación de secciones de importancia por parte del modelo para la toma de decisiones.

CAPÍTULO 9

Conclusiones

- Los tres pilares de la inteligencia artificial explicable fueron integrados exitosamente al modelo de detección de Leishmaniasis cutánea y estos permitieron obtener mayor información tanto sobre buenos comportamientos como comportamientos anómalos del proceso de toma de decisiones del modelo.
- Como deja en evidencia las observaciones de los miembros del Centro de Estudios en Salud de la Universidad del Valle de Guatemala y el análisis del modelo realizado durante este trabajo el modelo requiere de distintas mejoras antes de poder ser utilizado en el campo.
- La interpretabilidad, transparencia y explicabilidad del modelo introducida por los tres métodos de explicación elaborados durante el desarrollo de este proyecto permitió la identificación de comportamientos anómalos y la causa de los mismos.
- La explicación de algoritmos de caja negra es de vital importancia especialmente cuando estos son aplicados tienen un alto impacto en los usuarios que son afectados por la toma de decisiones del algoritmo.
- Existen patrones no deseados en las imágenes utilizadas para el entrenamiento del modelo que están afectando la identificación de la enfermedad Leishamniasis cutánea en las entradas del modelo.
- Las explicaciones generadas por los métodos de GradCAM y sensibilidad a la oclusión tuvieron mejores resultados y mejores retroalimentaciones al momento de ser presentados a los expertos del Centro de Estudios de Salud de la Universidad del Valle de Guatemala
- Las explicaciones generadas por LIME tuvieron resultados positivos demostrando el comportamiento del modelo, a pesar de esto las retroalimentaciones de los expertos del CES permiten observar áreas de mejora en el ámbito de la complejidad de las mismas.

CAPÍTULO 10

Recomendaciones

- Con el objetivo de mejorar el desempeño de futuros proyectos que se enfoquen en el uso del modelo de inteligencia artificial analizado durante el transcurso de este proyecto, se recomienda la revisión del proceso de entrenamiento del modelo. Como se dejó en evidencia en este trabajo, el comportamiento actual del modelo parece estar siendo afectado por distintas características de la imagen ajenas a posibles aspectos de la enfermedad Leishmaniasis cutánea. Una posible acción a tomar para mejorar el proceso de entrenamiento del modelo consiste en realizar un proceso de limpieza de las imágenes con el objetivo de remover la mayor cantidad de información no relevante a la enfermedad posible.

- Luego del análisis del modelo y otros modelos utilizados para le reconocimiento de imágenes un posible cambio para mejorar el comportamiento del modelo es modificar la arquitectura de la red neuronal. Al momento de comparar este modelo con otros modelos de clasificación de imágenes se pueden observar tanto capas distintas como arquitecturas distintas (capas convolucionales seguidas de otras capas convolucionales etc.), debido a la complejidad de las imágenes que se están analizando distintas arquitecturas de red neuronal podrían llegar a tener mejores comportamientos en la toma de decisiones.

- Dado que los métodos de explicación requieren distintas capacidades computacionales, siendo los métodos de Sensibilidad a la oclusión y las explicaciones generadas por la librería LIME las que requieren mayor cantidad de recursos, se debe analizar el ambiente en el cual se utilizará el modelo para realizar los debido cambios. Afortunadamente se pueden variar los parámetros utilizados para generar las explicaciones de todos los métodos, con el objetivo de reducir los recursos utilizados por cada explicación, de la siguiente manera: para el caso de la Sensibilidad a la oclusión se puede aumentar el tamaño del cuadrado utilizado para ocultar secciones de la imagen requiriendo menos iteraciones del proceso. Para las explicaciones utilizando LIME se puede modificar el número de muestras utilizadas para el cálculo de la aproximación para las funciones explicables. Por último, para las explicaciones generadas por el método de GradCAM se pueden disminuir la cantidad de capas a explicar, si se requiere realizar esto se su-

giere utilizar siempre una capa convolucional localizada al principio del modelo, una a mediados del modelo y una cercana a las capas de salida ya que las capas tienden a observar características más especificas de la entrada mientras esta se encuentre a mayor profundidad del modelo. Esta distribución de capas permite observar de manera general el comportamiento del modelo.

- Se recomienda probar los métodos de explicación con otros posibles médicos, si es posible potenciales usuarios del modelo de detección de Leishmaniasis. A pesar de que los modelos fueron presentados a miembros del Centro de Estudios en Salud lo cual presento retroalimentación valiosa, presentarlo a personal que tendrá mayor contacto con el mismo podría llegar a presentar perspectivas que agreguen mayor valor a las explicaciones presentadas.

CAPÍTULO 11

Bibliografía

Alzubi, J. ., Nayyar, A. . & Kumar, A. . (2018). Machine Learning from Theory to Algorithms: An Overview. *Journal of Physics: Conference Series*, *1142*, 012012. https://doi.org/10.1088/1742-6596/1142/1/012012

Barredo Arrieta, A. ., Díaz-Rodríguez, N. ., Del Ser, J. ., Bennetot, A. ., Tabik, S. ., Barbado, A. ., Garcia, S. ., Gil-Lopez, S. ., Molina, D. ., Benjamins, R. ., Chatila, R. . & Herrera, F. . (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, *58*, 82-115. https://doi.org/10.1016/j.inffus.2019.12.012

Bhatt, D. ., Patel, C. ., Talsania, H. ., Patel, J. ., Vaghela, R. ., Pandya, S. ., Modi, K. . & Ghayvat, H. . (2021). CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope. *Electronics*, *10*(20), 2470. https://doi.org/10.3390/electronics10202470

Cai, L. ., Gao, J. . & Zhao, D. . (2020). A review of the application of deep learning in medical image classification and segmentation. *Annals of Translational Medicine*, *8*(11), 713-713. https://doi.org/10.21037/atm.2020.02.44

Caruana, R. ., Lou, Y. ., Gehrke, J. ., Koch, P. ., Sturm, M. . & Elhadad, N. . (2015). Intelligible Models for HealthCare. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2783258.2788613

Chander, A., Srinivasan, R., Chelian, S., Wang, J. & Uchino, K. (2017). Working with Beliefs: AI Transparency in the Enterprise. *IUI Workshops*. http://ceur-ws.org/Vol-2068/exss14.pdf

Chavez. (2016). Leishmaniasis Guatemala. http://epidemiologia.mspas.gob.gt/files/Publicaciones%5C%202017/Malaria/Leishmaniasis%5C%202016.pdf

Davenport, T. . & Kalakota, R. . (2019). The potential for artificial intelligence in healthcare. *Future Healthcare Journal*, *6*(2), 94-98. https://doi.org/10.7861/futurehosp.6-2-94

Goodman, B. . & Flaxman, S. . (2017). European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation". *AI Magazine*, *38*(3), 50-57. https://doi.org/10.1609/aimag.v38i3.2741

Gunning, D. ., Stefik, M. ., Choi, J. ., Miller, T. ., Stumpf, S. . & Yang, G. Z. (2019). XAI—Explainable artificial intelligence. *Science Robotics*, *4*(37). https://doi.org/10.1126/scirobotics.aay7120

Hendricks, L. A., Burns, K. ., Saenko, K. ., Darrell, T. . & Rohrbach, A. . (2018). Women Also Snowboard: Overcoming Bias in Captioning Models. *Computer Vision – ECCV 2018*, 793-811. https://doi.org/10.1007/978-3-030-01219-9_47

IBM Cloud Education. (2021a). Convolutional Neural Networks. https://www.ibm.com/cloud/learn/convolutional-neural-networks

IBM Cloud Education. (2021b). Neural Networks. https://www.ibm.com/cloud/learn/neural-networks

Langley, P. ., Meadows, B. ., Sridharan, M. . & Choi, D. . (2017). Explainable Agency for Intelligent Autonomous Systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, *31*(2), 4762-4763. https://doi.org/10.1609/aaai.v31i2.19108

Nadeem, A. ., Verwer, S. ., Moskal, S. . & Yang, S. J. (2021). Alert-driven Attack Graph Generation using S-PDFA. *IEEE Transactions on Dependable and Secure Computing*, 1-1. https://doi.org/10.1109/tdsc.2021.3117348

Norvig, P. & Russell, S. (2022). *Artificial Intelligence: A Modern Approach, Global Edition (English Edition)*. Pearson.

Organización Mundial de la Salud. (2022). Leishmaniasis. https://www.who.int/es/news-room/fact-sheets/detail/leishmaniasis

O'Shea, K. & Nash, R. (2015). An Introduction to Convolutional Neural Networks. https://doi.org/10.48550/ARXIV.1511.08458

Rani, P. ., Liu, C. ., Sarkar, N. . & Vanman, E. . (2006). An empirical study of machine learning techniques for affect recognition in human–robot interaction. *Pattern Analysis and Applications*, *9*(1), 58-69. https://doi.org/10.1007/s10044-006-0025-y

Ribeiro, M. ., Singh, S. . & Guestrin, C. . (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. https://doi.org/10.18653/v1/n16-3020

Samek, W. & Müller, K.-R. (2019). Towards explainable artificial intelligence. *Explainable AI: interpreting, explaining and visualizing deep learning* (pp. 5-22). Springer.

Selvaraju, Cogswell, Das, Vendantam, Parikh & Batra. (2017). *Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization*. ICCV.

Tosun, A. B., Pullara, F. ., Becich, M. J., Taylor, D. L., Fine, J. L. & Chennubhotla, S. C. (2020). Explainable AI (xAI) for Anatomic Pathology. *Advances in Anatomic Pathology*, *27*(4), 241-250. https://doi.org/10.1097/pap.0000000000000264

Wang, M. ., Zheng, K. ., Yang, Y. . & Wang, X. . (2020). An Explainable Machine Learning Framework for Intrusion Detection Systems. *IEEE Access*, *8*, 73127-73141. https://doi.org/10.1109/access.2020.2988359

Wilson & Greenfield. (2019). Artificial Intelligence in Medicine: Applications, implications, and limitations. https://sitn.hms.harvard.edu/flash/2019/artificial-intelligence-in-medicine-applications-implications-and-limitations/

CAPÍTULO 12

Anexos

12.1. Red Neural Convolucional

Nombre de la capa	Tipo	Forma del Output
$conv2d$	Convolucional	(None, 298, 298, 16)
$max - pooling2d$	MaxPooling	(None, 149, 149, 16)
$conv2d - 1$	Convolucional	(None, 147, 147, 32)
$max - pooling2d - 1$	MaxPooling	(None, 73, 73, 32)
$conv2d - 2$	Convolucional	(None, 71, 71, 64)
$max - pooling2d - 2$	MaxPooling	(None, 35, 35, 64)
$conv2d - 3$	Convolucional	(None, 33, 33, 64)
$max - pooling2d - 3$	MaxPooling	(None, 16, 16, 64)
$conv2d - 4$	Convolucional	(None, 14, 14, 64)
$max - pooling2d - 4$	MaxPooling	(None, 7, 7, 64)
$flatten$	Flatten	(None, 3136)
nse	Dense	(None, 512)

Cuadro 3: Estructura del modelo de detección de Leishmaniasis

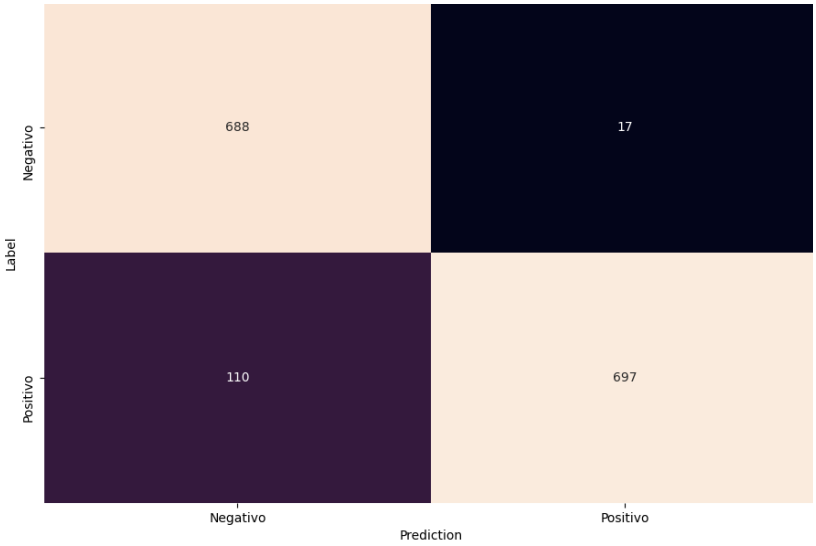


Figura 19: Matriz de confusión modelo de detección de Leishmaniasis

Perdida	Precisión
0.278754323720932	0.9173280596733093

Cuadro 4: Métricas de Evaluación para el modelo de detección de Leishmaniasis

12.2. Evidencia del comportamiento anómalo de la red neural convolucional

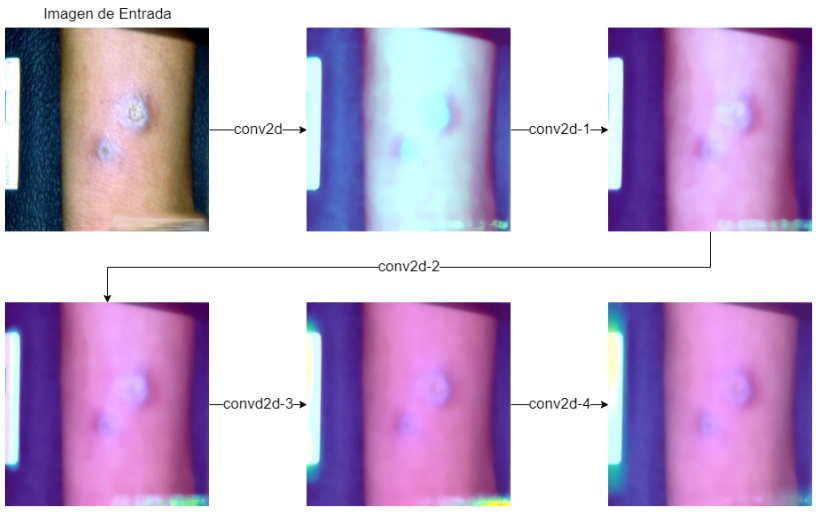


Figura 20: Evidencia de comportamiento anómalo 1

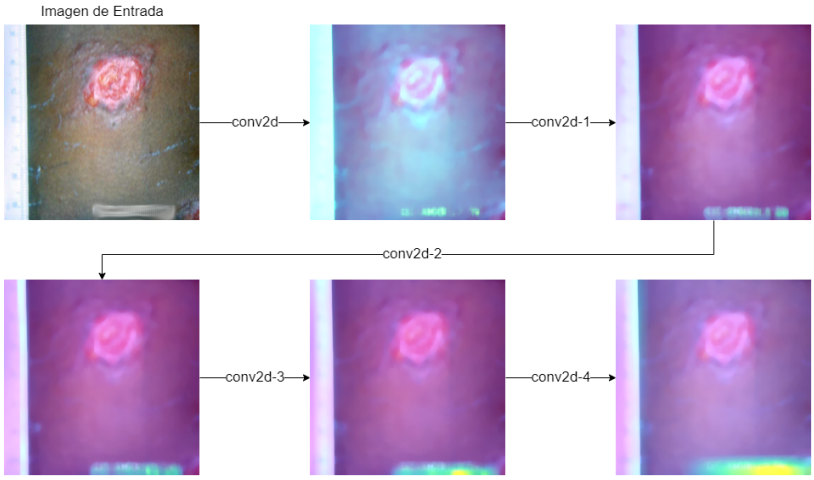


Figura 21: Evidencia de comportamiento anómalo 2

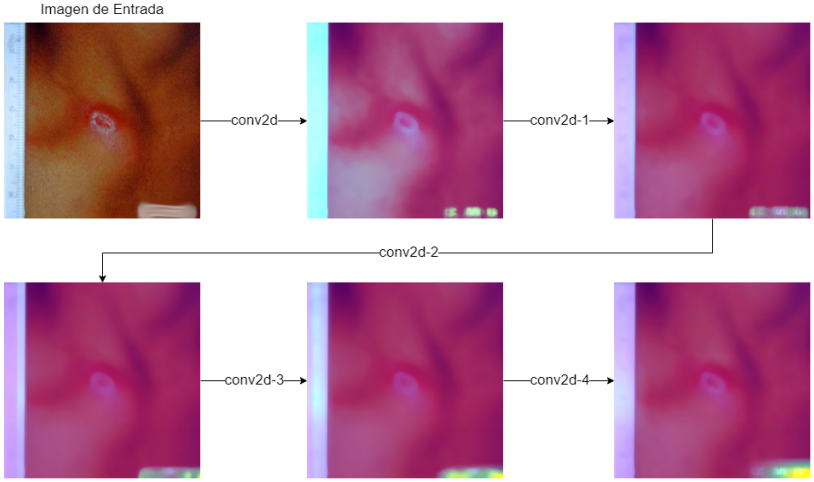


Figura 22: Evidencia de comportamiento anómalo 3

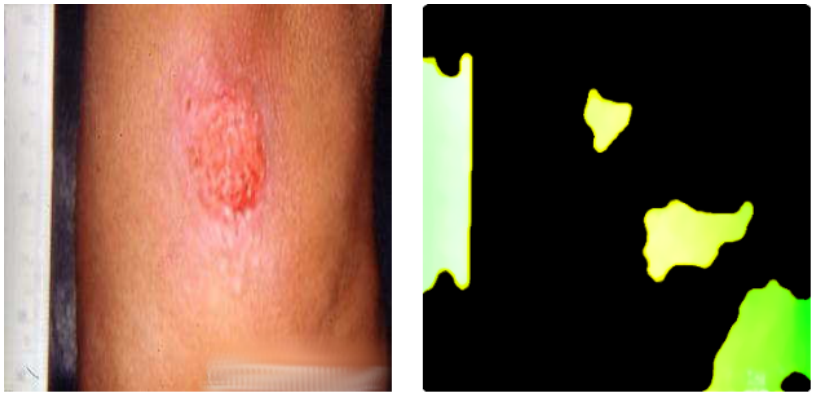


Figura 23: Evidencia de comportamiento anómalo 4

Figura 24: Evidencia de comportamiento anómalo 5

Figura 25: Evidencia de comportamiento anómalo 6

12.3. Algoritmos utilizados

12.3.1. Sensibilidad a la oclusión

```
import numpy as np
import tensorflow as tf
from matplotlib import pyplot as plt
from PIL import Image

# Create function to apply a grey patch on an image
PATCH_SIZE = 10
def apply_grey_patch(image, top_left_x, top_left_y, patch_size):
    patched_image = np.array(image, copy=True)
    patched_image[top_left_y:top_left_y + patch_size, top_left_x:top_left_x +
        patch_size, :] = 127.5
    return patched_image

def occlusion (path, model):
    img = tf.keras.preprocessing.image.load_img(path, target_size =(300,300))
    img = tf.keras.preprocessing.image.img_to_array(img)
```

```
    sensitivity_map = np.zeros((img.shape[0], img.shape[1]))
    for top_left_x in range(0, img.shape[0], PATCH_SIZE):
        for top_left_y in range(0, img.shape[1], PATCH_SIZE):
            patched_image = apply_grey_patch(img, top_left_x, top_left_y,
                PATCH_SIZE)
            predicted_classes = model.predict(np.array([patched_image]))[0]
            confidence = predicted_classes[0]


        # Save confidence for this specific patched image in map
        sensitivity_map[
            top_left_y:top_left_y + PATCH_SIZE,
            top_left_x:top_left_x + PATCH_SIZE,
        ] = confidence
    data = Image.fromarray(sensitivity_map, 'RGB')
    data.show()
    return sensitivity_map
```

12.3.2. Gradient-weighted Class Activation Maps

```
import cv2
import numpy as np
import tensorflow as tf


def gradCam(IMAGE_PATH, LAYER_NAME, model, CLASS_INDEX):
    img = tf.keras.preprocessing.image.load_img(IMAGE_PATH, target_size=(224, 224))
    img = tf.keras.preprocessing.image.img_to_array(img)


    # Create a graph that outputs target convolution and output
    grad_model = tf.keras.models.Model([model.inputs],
        [model.get_layer(LAYER_NAME).output, model.output])

    # Get the score for target class
    with tf.GradientTape() as tape:
        conv_outputs, predictions = grad_model(np.array([img]))
        loss = predictions[:, CLASS_INDEX]

    # Extract filters and gradients
    output = conv_outputs[0]
    grads = tape.gradient(loss, conv_outputs)[0]

    # Average gradients spatially
    weights = tf.reduce_mean(grads, axis=(0, 1))

    # Build a ponderated map of filters according to gradients importance
    cam = np.ones(output.shape[0:2], dtype=np.float32)

    for index, w in enumerate(weights):
        cam += w * output[:, :, index]
```

```
    # Heatmap visualization
    cam = cv2.resize(cam.numpy(), (224, 224))
    cam = np.maximum(cam, 0)
    heatmap = (cam - cam.min()) / (cam.max() - cam.min())

    cam = cv2.applyColorMap(np.uint8(255*heatmap), cv2.COLORMAP_JET)

    output_image = cv2.addWeighted(cv2.cvtColor(img.astype('uint8'),
        cv2.COLOR_RGB2BGR), 0.5, cam, 1, 0)
    return output_image
```