



Desarrollo de un prototipo de aplicación móvil para la detección de enfermedades de la piel (como el vitiligo y psoriasis), utilizando técnicas de visión artificial apoyado en machine learning.

Silva Cayambe, Dayanna Fernanda y Yupa Gallo, Carlos Vinicio

Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

Trabajo de titulación previo, a la obtención del título de Ingeniero en
Tecnologías de la Información.

Ing. Díaz Zuñiga, Magi Paul

10 de agosto del 2023



Escaneo e identificación por:
MAGI PAUL DIAZ
ZUNIGA



Plagiarism report
TESIS SILVA YUPA (1).pdf

Scan details

Scan time: October 30th, 2023 at 1:54 UTC
Total Pages: 38
Total Words: 9403

Plagiarism Detection



Types of plagiarism	Words
Identical	3.1% 293
Minor Changes	1.7% 161
Paraphrased	5.1% 481
Omitted Words	0% 0

AI Content Detection



Text coverage
AI text
Human text

Plagiarism Results: (61)

- T-ESPE-052856.pdf** 1.5%
<https://repositorio.espe.edu.ec/bitstream/21000/36712/1/t-e...>
EDISON SEBASTIAN LANDAZURI GALARZA
1 Sistema basado en IA para la detección y localización de tumores cerebrales en resonancias craneales (MRI) Changolulza Cundulli, Jeffe...
- REATEGUI PEZO ALEJANDRO DOCTORADO.pdf?sequ...** 1.2%
<https://repositorio.unfv.edu.pe/bitstream/handle/20.500.130...>
Alejandro Reategui Pezo
Vicerrectorado de INVESTIGACIÓN ESCUELA UNIVERSITARIA DE POSGRADO
"MÉTODO ALTERNATIVO BASADO EN UN SISTEMA INTELIGENTE PARA...
- ▷ Cómo programar una inteligencia artificial con Py...** 1%
<https://diariola.com/como-programar-una-inteligencia-artific...>

Firma:

.....
Ing. Diaz Zuñiga, Magi Paul

C. C: 1707249072



Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

Certificación

Certifico que el trabajo de titulación: **“Desarrollo de un prototipo de aplicación móvil para la detección de enfermedades de la piel (como el vitiligo y psoriasis), utilizando técnicas de visión artificial apoyado en machine learning”** fue realizado por los señores Silva Cayambe, Dayanna Fernanda y Yupa Gallo, Carlos Vinicio; el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Sangolquí, 16 de noviembre de 2023

MAGI
PAUL DIAZ
ZUNIGA

Firmado digitalmente por
MAGI PAUL DIAZ
ZUNIGA
Fecha: 2023.11.17
08:44:50 -05'00'

.....
Ing. Díaz Zuñiga, Magi Paul

C. C: 1707249072



Departamento de Ciencias de la Computación
Carrera de Tecnologías de la Información
Responsabilidad de Autoría

Nosotros, **Silva Cayambe, Dayanna Fernanda y Yupa Gallo, Carlos Vinicio**; con cédulas de ciudadanía N. 1718592312 y N. 1725075236, declaramos que el contenido, ideas y criterios del trabajo de titulación **“Desarrollo de un prototipo de aplicación móvil para la detección de enfermedades de la piel (como el vitiligo y psoriasis), utilizando técnicas de visión artificial apoyado en machine learning”**, es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 16 de noviembre de 2023

Silva Cayambe, Dayanna Fernanda

C. C: 1718592312

Yupa Gallo, Carlos Vinicio

C. C: 1725075236



Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

Autorización de Publicación

Nosotros, **Silva Cayambe, Dayanna Fernanda y Yupa Gallo, Carlos Vinicio**; con cédulas de ciudadanía N. 1718582312 y N.1725075236, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación "**Desarrollo de un prototipo de aplicación móvil para la detección de enfermedades de la piel (como el vitiligo y psoriasis), utilizando técnicas de visión artificial apoyado en machine learning**", en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Sangolquí, 16 de noviembre de 2023

Silva Cayambe, Dayanna Fernanda

C. C: 1718592312

Yupa Gallo, Carlos Vinicio

C. C: 1725075236

Dedicatoria

Con gratitud y satisfacción, dedico este logro a mi familia, por ser mi fuente inagotable de amor, aliento y comprensión, y por creer en mí en cada paso que he dado a lo largo de mi camino académico.

Dayanna Fernanda Silva Cayambe

A mi querida familia, Su constante apoyo ha sido mi impulso en la realización de esta tesis. Cada página refleja su amor y aliento. Con cariño, dedico este logro a todos ustedes que son mis pilares, mi fuerza y mi inspiración.

Carlos Vinicio Yupa Gallo

Agradecimiento

A la culminación de una de las etapas más importante de mi vida, deseo expresar mi sincero agradecimiento primero a Dios y a todas las personas que han contribuido de manera significativa a la realización de este logro.

A mis padres, por su amor, paciencia, apoyo constante a lo largo de este arduo camino y por nunca dejar de creer en mí. A mis hermanos, Vivi y Ariel, ya que siempre estuvieron apoyándome y nunca faltaron esas palabras de aliento para no rendirme.

A mis amigos, Carlitos y Sebas, por su apoyo incondicional, sus palabras de aliento en las etapas más difíciles de este camino y por cada risa que han iluminado incluso los días más oscuros de estudio.

Dayanna Fernanda Silva Cayambe

Deseo expresar mi sincero agradecimiento a Dios por ser mi guía constante y por brindarme la fortaleza y la sabiduría para completar este emocionante viaje. A mi familia, les estoy infinitamente agradecido por su apoyo inquebrantable y amor constante. A mis amigos Ali, Raque, Marilyn, Nilson, Andrés, Sebas, Dayanna y Naty, quienes han sido compañeros en innumerables aventuras y han brindado risas y amistad en cada paso del camino. Su presencia ha hecho que este camino sea memorable y gratificante. Juntos hemos enfrentado desafíos, celebrado logros y compartidos momentos inolvidables. Este trabajo no solo es un reflejo de mi esfuerzo, sino también de la comunidad valiosa que me ha rodeado. Gracias por ser una parte fundamental de este capítulo de mi vida.

Carlos Vinicio Yupa Gallo

Índice de contenido

Resumen.....	13
Abstract	14
Capítulo I.....	15
Introducción.....	15
Planteamiento del Problema	16
Justificación	16
Objetivos.....	17
Capítulo II.....	19
Marco Conceptual y Estado del Arte	19
Marco Conceptual	19
Vitíligo	19
Psoriasis	19
Visión Artificial	20
Aprendizaje de Máquina (Machine Learning)	21
Aprendizaje profundo (Deep learning)	21
Redes neuronales	22
Redes neuronales convolucionales	24
Algoritmo de Descenso de Gradiente Estocástico (SGD).....	25
Dataset.....	25
Lenguaje de programación.....	26
Estado del arte	27
Planteamiento de la revisión de literatura preliminar.....	27
Preguntas de investigación	27
Estrategia de búsqueda	27
Selección de estudios primarios	28
Evaluación de calidad.....	29

Resumen de los Estudios Primarios	29
Resumen general y conclusión del estado del arte	30
Capítulo III.....	32
Metodología	32
Metodología KDD	32
Capítulo IV	35
Desarrollo.....	35
Desarrollo del sistema.....	35
Capítulo V	44
Pruebas y Análisis de Resultados	44
Conclusiones y Recomendaciones	55
Conclusiones	55
Recomendaciones.....	56
Trabajos futuros	57
Bibliografía	58

Índice de tablas

Tabla 1 <i>Palabras clave de búsqueda de literatura</i>	28
Tabla 2 <i>Matriz de clasificación multiclase</i>	46

Índice de figuras

Figura 1 <i>Fases del proceso de funcionamiento de la visión artificial al determinar una imagen.....</i>	20
Figura 2 <i>Fases del método de backpropagation.....</i>	23
Figura 3 <i>Etapas del proceso KDD.....</i>	33
Figura 4 <i>Dataset Clasificados.....</i>	35
Figura 5 <i>Arquitectura del proyecto.....</i>	37
Figura 6 <i>Cargar dataset.....</i>	38
Figura 7 <i>Entrenamiento y Prueba.....</i>	38
Figura 8 <i>Convertir las etiquetas de enteros en vectores.....</i>	38
Figura 9 <i>Construir modelo y compilar.....</i>	39
Figura 10 <i>Entrenamiento del Modelo de Clasificación de Enfermedades de la Piel.....</i>	39
Figura 11 <i>Evaluación del Rendimiento del Modelo en Datos de Prueba.....</i>	40
Figura 12 <i>Visualización del Rendimiento de Entrenamiento y Validación.....</i>	40
Figura 13 <i>Exportación del Modelo Entrenado a Formato TFLite.....</i>	40
Figura 14 <i>Carga Archivo TfLite.....</i>	41
Figura 15 <i>Código fuente para elegir Image.....</i>	42
Figura 16 <i>Código fuente para la Predicción.....</i>	42
Figura 17 <i>Training Loss and Accuracy.....</i>	44
Figura 18 <i>Directorio Imágenes para el test.....</i>	47
Figura 19 <i>Archivo pre entrenado para realizar las pruebas.....</i>	48
Figura 20 <i>Clasificación etiquetas de clase.....</i>	48
Figura 21 <i>Prueba de clasificación de Imágenes caso vitíligo.....</i>	49
Figura 22 <i>Prueba de clasificación de Imágenes caso psoriasis.....</i>	50
Figura 23 <i>Prueba de clasificación de Imágenes caso no favorable.....</i>	50
Figura 24 <i>Interfaz Inicial APP.....</i>	51
Figura 25 <i>Selección Foto a Clasifica.....</i>	51

Figura 26 <i>Resultados caso vitíligo</i>	52
Figura 27 <i>Resultados caso psoriasis</i>	52
Figura 28 <i>Resultados caso vitíligo imagen más detalla</i>	53

Resumen

Este trabajo aborda el desarrollo de un prototipo de aplicación móvil innovadora diseñada para la detección temprana y precisa de enfermedades de la piel, específicamente el vitíligo y la psoriasis. El enfoque central se basa en la aplicación de técnicas avanzadas de visión artificial, respaldadas por el aprendizaje automático (machine learning).

El prototipo de aplicación móvil aprovecha algoritmos de visión artificial para analizar imágenes de la piel proporcionadas por los usuarios. Estas imágenes son procesadas y evaluadas mediante modelos de machine learning entrenados con datos médicos relevantes. La aplicación es capaz de detectar patrones y anomalías característicos de enfermedades de la piel, lo que permite identificar posibles casos de vitíligo y psoriasis de manera temprana.

La investigación implica la recolección y preparación de datos médicos, así como el diseño y entrenamiento de modelos de machine learning específicos para la detección de estas enfermedades. Además, se desarrolla una interfaz de usuario intuitiva y amigable en forma de aplicación móvil, que permite a los usuarios capturar imágenes de la piel y recibir evaluaciones preliminares sobre su condición.

Los resultados y conclusiones de esta investigación muestran el potencial de la visión artificial y el machine learning en la detección de enfermedades de la piel. El prototipo de aplicación móvil representa una herramienta prometedora para la detección temprana y la concienciación sobre el vitíligo y la psoriasis, lo que podría tener un impacto significativo en la salud pública al facilitar la identificación oportuna y el tratamiento adecuado de estas condiciones cutáneas.

Palabras claves: machine learning, aplicación móvil, enfermedades en la piel, visión artificial, inteligencia artificial.

Abstract

This work addresses the development of a prototype of an innovative mobile application designed for the early and accurate detection of skin diseases, specifically vitiligo and psoriasis. The central approach is based on the application of advanced computer vision techniques, supported by machine learning.

The prototype mobile application leverages machine vision algorithms to analyze skin images provided by users. These images are processed and evaluated using machine learning models trained with relevant medical data. The app is able to detect patterns and abnormalities characteristic of skin diseases, allowing potential cases of vitiligo and psoriasis to be identified early.

The research involves the collection and preparation of medical data, as well as the design and training of specific machine learning models for the detection of these diseases. In addition, an intuitive and user-friendly user interface is developed in the form of a mobile application, which allows users to capture skin images and receive preliminary assessments about their condition.

The results and conclusions of this research show the potential of machine vision and machine learning in skin disease detection. The prototype mobile application represents a promising tool for early detection and awareness of vitiligo and psoriasis, which could have a significant impact on public health by facilitating the timely identification and appropriate treatment of these skin conditions.

Keywords: machine learning, mobile application, skin diseases, computer vision, artificial intelligence

Capítulo I

Introducción

En la era digital en la que vivimos, los avances tecnológicos están transformando diversos aspectos de nuestra vida cotidiana. Uno de los campos en los que la tecnología móvil ha tenido un impacto significativo es el cuidado de la salud. En particular, las aplicaciones móviles han surgido como herramientas útiles y accesibles para el diagnóstico y monitoreo de diversas enfermedades, incluyendo las relacionadas con la piel.

Entre las afecciones dermatológicas más comunes en la población general se encuentran el vitíligo y la psoriasis, con tasas de prevalencia informadas de alrededor del 1% al 3% y 0.5%, respectivamente. Sin embargo, en diferentes regiones, como en India, se han registrado tasas de prevalencia más altas, llegando al 3% al 4% para el vitíligo.

En el contexto de Ecuador, el vitíligo es una de las principales razones para buscar atención dermatológica, con aproximadamente el 5% de las consultas, mientras que la psoriasis representa alrededor del 2%. Este escenario puede tener un impacto significativo en la calidad de vida de las personas afectadas. La falta de acceso a servicios médicos y la escasa conciencia sobre la importancia de un diagnóstico temprano han llevado a que muchas personas no reciban el tratamiento adecuado para estas condiciones dermatológicas.

Por lo tanto, se ha vuelto cada vez más importante buscar soluciones tecnológicas innovadoras para mejorar la detección y tratamiento de estas enfermedades.

En este contexto, la presente investigación se enfoca en el desarrollo de un prototipo de aplicación móvil que utilice técnicas de visión artificial apoyado en machine learning para detectar enfermedades de la piel, en particular el vitíligo y la psoriasis. La aplicación móvil permitirá a los usuarios tomar fotografías de su piel y, mediante el uso de algoritmos de aprendizaje automático, analizar las imágenes para detectar signos tempranos de estas enfermedades, con el fin de permitir a los usuarios realizarse un autodiagnóstico de manera sencilla y eficiente, lo que puede ayudar significativamente la detección temprana de estas enfermedades y, por lo tanto, mejorar la calidad de vida de las personas

afectadas.

La aplicación móvil también proporcionará recomendaciones personalizadas de tratamiento, basadas en la información obtenida del análisis de la imagen y los datos del usuario. De esta manera, la aplicación móvil contribuirá a mejorar la calidad de vida de las personas afectadas por estas enfermedades al brindarles una herramienta accesible y efectiva para la detección temprana y el tratamiento de sus síntomas.

Planteamiento del Problema

La falta de acceso a especialistas y la falta de información sobre los síntomas y tratamientos disponibles son barreras importantes para la atención efectiva de las enfermedades de la piel. Por lo tanto, se necesita una solución innovadora y accesible para mejorar la detección temprana y el tratamiento de estas enfermedades. El uso de la visión artificial apoyada en machine learning en una aplicación móvil puede ser una herramienta efectiva para la detección temprana y el diagnóstico preciso de enfermedades de la piel, lo que puede mejorar la calidad de vida de las personas afectadas y contribuir a una atención más efectiva de estas afecciones.

Sin embargo, se requiere investigación para determinar la eficacia de dichas técnicas en la detección de vitíligo y psoriasis, así como para desarrollar un algoritmo que funcione con un alto grado de precisión en la identificación de estas enfermedades, se deben abordar posibles desafíos técnicos, como la elección de las técnicas de visión artificial y machine learning más adecuadas y rentables para la aplicación móvil, la recolección de datos y su integración con el prototipo de aplicación móvil.

Por lo tanto, el planteamiento de problema en esta investigación se enfoca en cómo desarrollar una aplicación móvil que sea precisa, eficaz y fácil de usar para detectar enfermedades de la piel como el vitíligo y la psoriasis, y de cómo esta tecnología puede mejorar el acceso a la atención y el tratamiento adecuado para personas con enfermedades de la piel.

Justificación

En Ecuador, la salud es uno de los principales problemas que enfrenta la población,

y la detección temprana de enfermedades de la piel como el vitíligo y la psoriasis es fundamental para mejorar la calidad de vida de las personas afectadas. Sin embargo, en muchos casos, la detección de estas enfermedades es difícil debido a la falta de acceso a especialistas y a herramientas de diagnóstico precisas.

En este contexto, el desarrollo de una aplicación móvil que utilice técnicas de visión artificial apoyadas en machine learning puede ser una solución innovadora y accesible para la detección y diagnóstico temprano de enfermedades de la piel en Ecuador. Esta tecnología podría permitir a los pacientes realizar un autodiagnóstico y recibir recomendaciones personalizadas de tratamiento, lo que podría mejorar significativamente la atención médica de estas enfermedades en un país con limitaciones en cuanto a acceso y recursos.

Además, el uso de la tecnología móvil es cada vez más común en Ecuador, lo que facilita la implementación y el acceso a este tipo de soluciones para un público amplio. En este sentido, la presente investigación busca contribuir al desarrollo de soluciones tecnológicas innovadoras y accesibles que puedan mejorar la calidad de vida de las personas afectadas por enfermedades de la piel en Ecuador, promoviendo una atención más efectiva y temprana de estas afecciones.

Objetivos

Objetivo general

- Implementar un prototipo de aplicación móvil para la detección de enfermedades de la piel (como el vitíligo y psoriasis), utilizando técnicas de visión artificial apoyado en machine learning.

Objetivos específicos

- Realizar una revisión exhaustiva de la literatura existente sobre el vitíligo y la psoriasis, incluyendo los métodos de detección actuales, la evaluación de lesiones en la piel y el uso de técnicas de visión artificial en la detección de enfermedades de la piel.
- Desarrollar el modelo de machine learning basado en la metodología KDD, para la

detección temprana de enfermedades de la piel.

- Desarrollar el aplicativo móvil basado en la metodología Scrum que permita la toma de datos para la detección temprana de enfermedades en la piel.

Capítulo II

Marco Conceptual y Estado del Arte

Marco Conceptual

Vitiligo

El vitiligo es una afección cutánea degenerativa que se caracteriza por la pérdida progresiva de pigmentación en placas localizadas en la piel. Según (Molé & Coringrato, 2019) esta afección se manifiesta clínicamente mediante la aparición gradual de áreas de la piel desprovistas de pigmentación (máculas acrómicas e hipocrómicas) que tienen un impacto significativo en la estética del individuo. El vitiligo tiende a afectar áreas visibles de la piel. El enfoque del tratamiento del vitiligo se centra en detener la progresión de la enfermedad, fomentar la hiperpigmentación y prevenir la recurrencia. Esto se logra a menudo a través del uso de esteroides tópicos, con el objetivo final de mejorar la calidad de vida del paciente.

Psoriasis

La psoriasis es una afección cutánea que conduce a la aparición de enrojecimiento, escamas e irritación en la piel. La mayoría de las personas que la padecen experimentan la presencia de parches de piel y escamas, los cuales son rojos y bien definidos. De acuerdo con (Tamayo Otero, Prieto Cuesta, Villegas Llangary, & Menéndez Reyes, 2022) la psoriasis es considerada una condición que puede llegar a ser crónica e inflamatoria. Se ha identificado que puede ser genética, también por afección que depende de varios factores desencadenantes, como infecciones bacterianas, estrés, traumatismos y medicamentos, que pueden estar relacionados con su origen.

Según (Das, 2020), la psoriasis puede ocurrir en cualquier tipo de piel, pero es más común en personas de piel clara, especialmente de ascendencia europea. Sin embargo, también puede afectar a personas de piel más oscura. El rango de edad es de 16 a 22, y también de los 87 a 60 años.

Algunos estudios sugieren que la psoriasis puede ser más difícil de diagnosticar en personas de piel oscura debido a la diferencia en la apariencia de las lesiones en

comparación con las personas de piel clara. Las lesiones de psoriasis en personas de piel oscura pueden parecer más oscuras, violáceas o incluso grises, lo que puede llevar a un diagnóstico erróneo o retrasado.

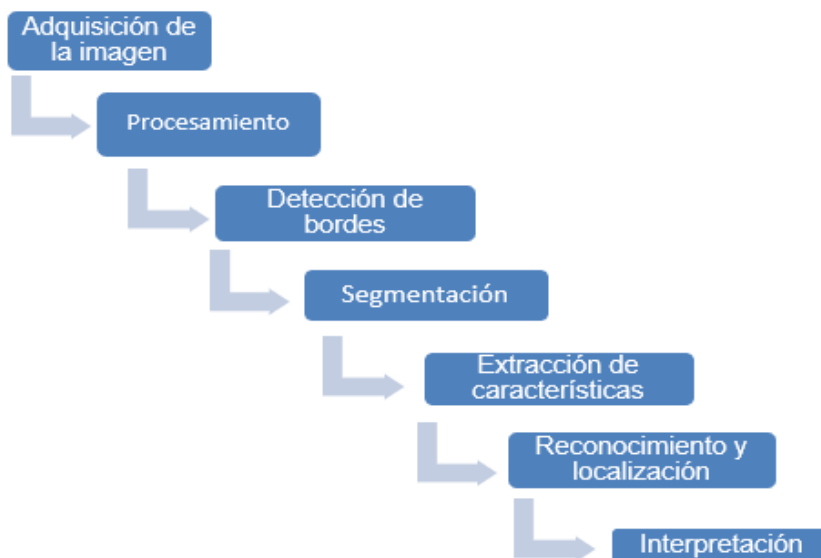
Visión Artificial

La visión artificial es un componente de la inteligencia artificial (IA) que permite que los ordenadores y sistemas extraer información importante a partir de imágenes digitales, videos y diferentes entradas visuales.

En su trabajo (Cárdenas Vera, Llerena Pizarro, & Calle, 2012), nos dice que la visión artificial es un campo de la IA y la informática que se centra en capacitar a las máquinas para comprender y analizar el mundo visual de la misma manera que lo hacen los seres humanos. Su objetivo es permitir que las máquinas procesen y comprendan imágenes y videos, extrayendo información útil y tomando decisiones basadas en esa información

Figura 1

Fases del proceso de funcionamiento de la visión artificial al determinar una imagen.



Nota. El gráfico representa el proceso de como son la fase del proceso de funcionamiento de la visión artificial. Tomado de (Autoría propia).

En el proceso de adquisición de imágenes, se transforma la imagen física en formato digital. Una vez digitalizada, se somete a una mejora inicial para eliminar elementos no

deseados, lo que facilita las siguientes etapas. Luego, se detectan los bordes para separar los objetos de interés del fondo, reduciendo así la cantidad de datos. Después, se lleva a cabo la segmentación, que selecciona píxeles según sus valores de color en el espacio RGB o HSV. El modelo RGB utiliza tonalidad, saturación y valor para combinar colores y resaltar objetos específicos. Finalmente, en el proceso de reconocimiento y ubicación, los objetos con características similares se agrupan en un espacio tridimensional utilizando técnicas de triangulación. Este paso permite seleccionar e interpretar efectivamente la imagen.

Aprendizaje de Máquina (Machine Learning)

El aprendizaje de máquina es un campo de la inteligencia artificial, que nos permite que los sistemas aprendan y mejoren a partir de la experiencia. Utiliza técnicas estadísticas para permitir que las computadoras aprendan y tomen decisiones sin necesidad de ser programadas.

Según (Ricardo Coronado, 2018), en su trabajo nos menciona que el aprendizaje de máquina es un campo interdisciplinario, relacionado con el desarrollo de programas de cómputo que mejora su desempeño a través del entrenamiento.

El Aprendizaje de máquina según (Mohammad, y otros, 2016), es cómo pueden las computadoras aprender (o mejorar su rendimiento) en base a datos conocidos.

Aprendizaje profundo (Deep learning)

El aprendizaje profundo forma parte del ámbito del aprendizaje automático y se trata de una variante de las redes neuronales que consta de tres capas o más. Estas redes neuronales buscan imitar el funcionamiento del cerebro humano, aunque no logran igualar su capacidad para "aprender" a partir de grandes cantidades de información. Aunque una red neuronal de una única capa puede generar aproximaciones de predicciones, agregar capas ocultas adicionales puede mejorar y perfeccionar la precisión.

De acuerdo con lo planteado por (LeCun, Bengio, & Geoffrey, 2015), el aprendizaje profundo posibilita que los modelos computacionales, compuestos por múltiples capas de procesamiento, adquieran representaciones de datos con diferentes niveles de abstracción.

Redes neuronales

Una red neuronal artificial (ANN) es un modelo de procesamiento distribuido que se inspira en la estructura del sistema nervioso humano. La arquitectura de una red neuronal implica la conexión de múltiples procesadores elementales, lo que crea un sistema adaptable que utiliza un algoritmo para ajustar sus pesos (parámetros libres) con el objetivo de cumplir con los requisitos de rendimiento del problema, basándose en ejemplos representativos. Según (Basogain Olabe, 2015), es importante destacar que la característica más distintiva de las redes neuronales artificiales es su capacidad para aprender a partir de un conjunto de patrones de entrenamiento, lo que les permite identificar un modelo que se ajuste a los datos. El proceso de aprendizaje, también conocido como entrenamiento de la red, puede ser supervisado o no supervisado.

Redes neuronales – backpropagation learning

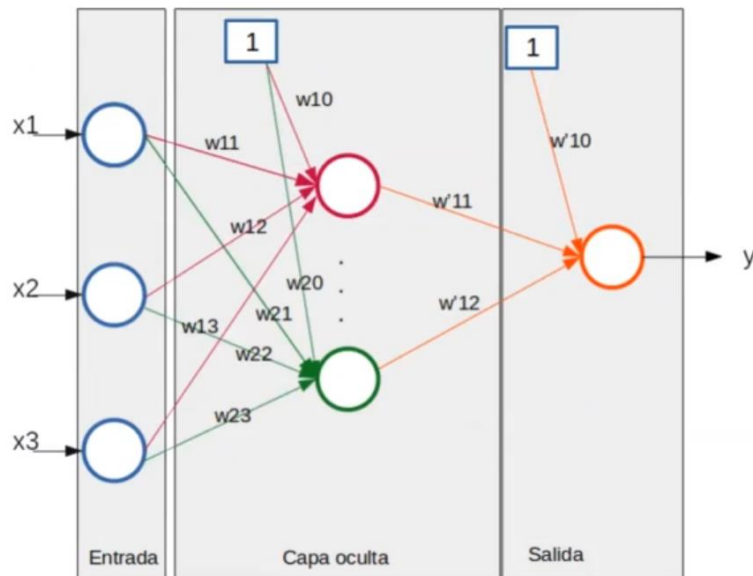
Backpropagation es un método utilizado en el entrenamiento de redes neuronales artificiales que tiene como objetivo reducir los errores en el proceso de aprendizaje automático de una máquina.

Según (Hidalgo Cajo, Yasaca Pucuna, Hidalgo Cajo, Hidalgo Cajo, & Latorre Benalcázar, 2020), el algoritmo de backpropagation (BP) se clasifica como un enfoque de entrenamiento supervisado. En este algoritmo, se divide en dos etapas: una etapa de funcionamiento en la cual se introduce un patrón de entrada a la red ya entrenada, y este patrón se propaga a través de las diferentes capas de neuronas hasta generar una salida. Luego, sigue una etapa de entrenamiento o aprendizaje donde se ajustan los pesos de la red para lograr que la salida producida coincida con la salida deseada.

Fases del método de backpropagation

Figura 2

Fases del método de backpropagation



Nota. En la figura se detalla las fases del algoritmo de backpropagation. Obtenido de (<https://keepcoding.io/blog/backpropagation-solucion-perceptron-multicapas/>).

Las etapas del proceso del algoritmo de backpropagation son las siguientes:

- Selección del Punto de Entrada en la Capa de Entrada: Se elige el punto de entrada inicial para el proceso.
- Asignación de Valores Secundarios: Una vez que los valores de entrada y salida están configurados, el algoritmo de backpropagation asigna valores secundarios que permiten ajustar los parámetros en cada capa y nodo de la red neuronal.
- Análisis de Nodos y Capas en la Capa Oculta: Se analizan los nodos y capas en la capa oculta para determinar el error total, que es la diferencia entre la salida real y la salida deseada.
- Minimización del Error en Toda la Red Neuronal: El algoritmo trabaja para minimizar el error en toda la red neuronal.
- Ajuste de Parámetros: Backpropagation ajusta los parámetros (pesos y sesgos) para reducir la tasa de error. Este proceso se repite iterativamente hasta que el error se reduce al mínimo posible.

- Preparación para la Predicción: Una vez ajustados los parámetros, el modelo está listo para hacer predicciones. Al introducir datos de entrada, el modelo genera una salida correspondiente.

El objetivo fundamental del algoritmo de backpropagation es reducir los errores durante el proceso de aprendizaje y mejorar la precisión de las predicciones ajustando los pesos y sesgos de cada nodo en la red. Backpropagation identifica la contribución de cada nodo en posibles errores y ajusta los parámetros de estos nodos para corregir su configuración. Estos ajustes influyen en nodos y capas subsiguientes, lo que contribuye a la mejora del rendimiento global de la red.

Redes neuronales convolucionales

Las Redes Neuronales Convolucionales (CNN) son una serie de redes diseñadas para imitar el funcionamiento cerebral y tienen la capacidad de aprender en distintos niveles de abstracción. En la primera capa, distinguen elementos simples como colores y bordes; en la capa siguiente, combinaciones de bordes y colores; finalmente, en la última capa se enfocan en la forma para determinar la identidad exacta del objeto.

En este proceso, las computadoras emplean filtros o lentes para examinar las diversas características de las redes neuronales convolucionales: un filtro puede enfocarse en bordes diagonales, otro en colores, etc. La técnica implica pasar estos filtros por toda la imagen, explorándola, para posteriormente describirla y categorizarla. Las Redes Neuronales Convolucionales pueden ser usadas para capacitar a las computadoras a "ver", permitiendo la creación de sistemas "inteligentes".

Según (Artola Moreno, Á. 2019), la red neuronal debe procesar una amplia variedad de imágenes para captar las características únicas de cada objeto y lograr una generalización efectiva. Cada imagen es representada como una matriz de píxeles, con valores que van de 0 a 255, aunque para su procesamiento en la red neuronal se normalizan en un rango de 0 a 1. La entrada inicial para la red consiste en los píxeles de una imagen.

Las CNNs automáticamente aprenden las características más relevantes de los datos de entrada a través del proceso de entrenamiento. Esto se logra mediante la propagación hacia

atrás del error y el ajuste de los pesos de las capas mediante algoritmos de optimización, como el descenso de gradiente estocástico (SGD) o Adam, para minimizar la función de pérdida.

Algoritmo de Descenso de Gradiente Estocástico (SGD)

Este algoritmo es utilizado para optimizar la función de pérdida de un modelo de aprendizaje automático durante el proceso de entrenamiento. El objetivo es encontrar los parámetros del modelo que minimizan la función de pérdida y, por lo tanto, hacen que el modelo sea más preciso en sus predicciones.

Dataset

Un dataset, tal como su nombre indica, constituye un conjunto de datos que usualmente se encuentra estructurado de manera específica. A modo de ejemplo, podemos considerar una tabla en una base de datos SQL como una instancia de dataset. En esta tabla, cada columna representa una variable particular, mientras que las filas representan registros individuales asociados a cada columna o variable.

En resumen, un dataset constituye una agrupación organizada de información que puede adoptar la forma de una tabla en una base de datos u otra estructura similar.

Características de los dataset

Un conjunto de datos, o data set en inglés, puede englobar una variedad de tipos de información, como texto, números, imágenes, videos, registros de eventos, y más. Estos datos son recopilados y almacenados con el propósito de llevar a cabo análisis, investigaciones, tomar decisiones informadas o entrenar modelos de aprendizaje automático.

Los conjuntos de datos pueden diferir en términos de tamaño y complejidad. Algunos pueden ser pequeños, conteniendo apenas unas pocas filas o registros, mientras que otros pueden ser inmensamente grandes, con millones o incluso miles de millones de registros. Además, estos conjuntos de datos pueden ser generados manualmente por individuos, generados automáticamente por máquinas, o recopilados de fuentes externas, como bases de datos, encuestas, experimentos científicos, registros históricos o plataformas de redes sociales, entre otros.

Es de vital importancia disponer de conjuntos de datos de alta calidad para garantizar un análisis sólido y obtener conclusiones válidas. Esto involucra asegurarse de que los datos sean precisos, estén completos, sean consistentes y cuenten con documentación adecuada. Además, es esencial cumplir con las regulaciones y políticas de privacidad pertinentes en el manejo de los datos, protegiendo información delicada y respetando la privacidad de las personas involucradas.

Lenguaje de programación

Un lenguaje de programación es un idioma artificial que se emplea para controlar el comportamiento de máquinas, especialmente computadoras. Similar a los lenguajes humanos, los lenguajes de programación están definidos por reglas sintácticas y semánticas que determinan su estructura y significado, respectivamente.

Estos lenguajes son utilizados para facilitar la comunicación en tareas de organización y manipulación de información, y para expresar algoritmos con precisión. En ocasiones, el término "lenguaje de programación" se restringe a aquellos capaces de expresar todos los algoritmos posibles. En otros casos, se utiliza "lenguaje informático" para lenguajes más limitados. Sin embargo, en este contexto, se implementó el lenguaje de programación Python para la parte del backend y se usó el IDE Android Studio con el lenguaje de programación Java para el desarrollo del frontend.

Python fue creado por Guido van Rossum en la década de los 80 y comienzos de los 90. En 2000, se lanzó Python 2.0 con características como recolección de basura y soporte completo a Unicode. Python 3.0, una versión mayor, introdujo cambios significativos. En este proyecto, se usa la versión 3.11.3. Java, por su parte, fue lanzado por Sun Microsystems en 1995 y adquirido por Oracle en 2010.

Java se destaca por su arquitectura WORA ("write once, run anywhere"), lo que significa que un programa escrito en Java puede ejecutarse en cualquier plataforma con una máquina virtual de Java instalada. Java se usa para desarrollar aplicaciones de escritorio, servidores, sistemas operativos, aplicaciones móviles, plataformas IoT, televisores inteligentes, sistemas empresariales y más. En este proyecto, Java se emplea para el

desarrollo del frontend, permitiendo la conexión con el modelo de aprendizaje desarrollado en Python.

Estado del arte

Este estudio busca información sobre la detección de enfermedades de la piel (como el vitíligo y psoriasis), utilizando técnicas de visión artificial apoyado en machine learning.

En la exploración de datos, se incluyen pasos importantes como la formulación de preguntas de investigación, la revisión de los objetivos, la realización de búsquedas, la presentación de resultados y la selección de investigaciones pertinentes

Planteamiento de la revisión de literatura preliminar

El objetivo principal de esta revisión de la literatura es determinar los elementos apropiados para llevar a cabo una revisión del estado del arte, incluyendo la formulación de la pregunta de investigación, la estrategia de búsqueda y la selección de estudios primarios.

Preguntas de investigación

La presente investigación tiene como objetivo conocer información sobre la detección de enfermedades de la piel (como el vitíligo y psoriasis), utilizando técnicas de visión artificial apoyado en machine learning, para lo cual se planteó la siguiente pregunta de investigación: "¿Cómo se puede desarrollar un prototipo de aplicación móvil utilizando técnicas de visión artificial apoyadas en machine learning para la detección temprana y precisa de enfermedades de la piel, como el vitíligo y la psoriasis?. Esta pregunta de investigación plantea el objetivo de desarrollar una aplicación móvil que utilice técnicas de visión artificial y machine learning para detectar enfermedades de la piel, específicamente el vitíligo y la psoriasis. Al centrarse en la detección temprana y precisa, estás abordando dos aspectos clave que pueden tener un impacto significativo en el diagnóstico y tratamiento de estas enfermedades.

Estrategia de búsqueda

Para realizar la búsqueda de los estudios primarios, se hizo uso del motor de búsqueda académica Google Scholar. Se empleó la siguiente cadena de búsqueda durante el proceso de investigación: (aplicación móvil OR software) AND (visión artificial) AND

(aprendizaje de maquina and machine learning) AND (enfermedades en la piel). Se tomará en cuenta lo siguiente el término OR se usa para unir palabras semejantes y el término AND para unir diferentes.

Tabla 1

Palabras clave para la

Concepto	Términos alternativos	Conector
Aplicación móvil	(aplicación móvil OR software)	AND
visión artificial	(visión artificial)	AND
aprendizaje de maquina	(aprendizaje de maquina and machine learning)	AND
enfermedades en la piel	(enfermedades en la piel)	

Selección de estudios primarios

Se evaluaron los resultados de la búsqueda examinando el título, el resumen y las palabras clave de los artículos para determinar cuáles eran relevantes. Además, se incluyeron estudios que cumplieran con uno o más de los criterios de inclusión que se describen a continuación.

- Artículos que se relacionen con implementación de modelos de aprendizaje sobre enfermedades.
- Artículos donde se desarrolle aplicaciones móviles para detección de enfermedades en la piel.
- Artículos entre 2017 a 2023

Se eliminaron los artículos que cumplan con al menos uno de los criterios de exclusión detallados a continuación:

- Artículos que no se detallen sobre modelos de aprendizaje de enfermedades
- Artículos en idiomas diferentes a español o ingles
- Artículos que únicamente muestren definiciones referentes a las palabras claves.
- Artículos publicados antes del 2017

Evaluación de calidad

Se baso en las siguientes preguntas para evaluar la calidad del contenido de los artículos seleccionados:

- El estudio muestra modelos de aprendizaje sobre enfermedades.
- El estudio muestra el desarrollo de una aplicación para detección de enfermedades.
- El estudio muestra algoritmos de modelos de aprendizaje.
- El estudio presenta la implementación que permitan la detección de enfermedades.

Es necesario que los artículos cumplan con al menos tres de las cinco interrogantes planteadas, lo que facilita la identificación y posterior selección de artículos que sean pertinentes para cada pregunta de investigación.

Resumen de los Estudios Primarios

1. Título: "Automated Skin Disease Identification using Machine Learning Techniques"
Autores: S. Alouane, S. Boukhris, M. Quidar Publicado en: 2018 15th International Multi-Conference on Systems, Signals & Devices Resumen: Este estudio propone un sistema automatizado para identificar enfermedades de la piel utilizando técnicas de aprendizaje automático. Se aplicaron algoritmos de clasificación para detectar enfermedades específicas, como el vitíligo y la psoriasis, con resultados prometedores.
2. Título: "Mobile Application for Skin Disease Detection using Convolutional Neural Networks" Autores: S. Verma, S. Rani, A. Kaur Publicado en: 2020 IEEE 6th International Conference on Computer and Communications (ICCC) Resumen: En este trabajo se presenta una aplicación móvil que utiliza redes neuronales convolucionales (CNN) para detectar enfermedades de la piel. El sistema permite a los usuarios capturar imágenes de su piel y obtener una clasificación basada en la detección y el análisis de características específicas relacionadas con el vitíligo y la psoriasis.

3. Título: " Sistema automatizado de reconocimiento de enfermedades de la piel mediante Deep Learning" Autores: S. L. Ting, W. H. Heng, A. M. H. Soh, K. H. Lim Publicado en: 2017 International Conference on Machine Learning and Cybernetics (ICMLC) Resumen: Este trabajo presenta un sistema de reconocimiento automatizado de enfermedades de la piel utilizando técnicas de aprendizaje profundo. Se utilizó una red neuronal convolucional (CNN) para clasificar imágenes de enfermedades de la piel, incluyendo el vitíligo y la psoriasis.
4. Título: "Development of a Mobile Application for Skin Disease Diagnosis based on Deep Learning" Autores: S. Ghosh, S. Das, S. Pal Publicado en: 2019 International Conference on Electrical, Computer and Communication Technologies (ICECCT) Resumen: En este estudio, se desarrolló una aplicación móvil que utiliza algoritmos de aprendizaje profundo para el diagnóstico de enfermedades de la piel. Se entrenó una red neuronal convolucional profunda utilizando un conjunto de datos etiquetados y se implementó en un dispositivo móvil para la detección y clasificación de enfermedades cutáneas.
5. Título: " Diagnóstico automatizado de psoriasis vulgar mediante técnicas Machine Learning " Autores: S. R. Joshi, D. S. Doye Publicado en: 2018 IEEE 5th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON) Resumen: En este trabajo, se propone un sistema de diagnóstico automatizado para la psoriasis utilizando técnicas de aprendizaje automático. Se aplicaron algoritmos de clasificación, como el SVM (Máquinas de Vectores de Soporte), para detectar y clasificar la psoriasis en imágenes de la piel.

Resumen general y conclusión del estado del arte

Resumen general

Se realizó la búsqueda de información y artículos relacionados referente a Desarrollo de un prototipo de aplicación móvil para la detección de enfermedades de la piel (como el vitíligo y psoriasis), utilizando técnicas de visión artificial apoyado en machine learning. El proceso comenzó con la definición de preguntas, la revisión de los objetivos, la realización

de la búsqueda, la presentación de la información y la selección de estudios pertinentes. Estas etapas son fundamentales para el cuestionamiento que guía esta revisión del estado del arte en el ámbito de la investigación literaria.

A continuación, se establecieron los objetivos de la revisión literaria con el propósito de definir el alcance de la revisión y contribuir a la delimitación de la búsqueda realizada. En la fase de búsqueda, se aplicaron técnicas de validación y selección de artículos primarios, lo que permitió eliminar información irrelevante o duplicada de los artículos encontrados.

Finalmente, se presenta la información recopilada y se seleccionan las investigaciones pertinentes que respaldan el desarrollo del proyecto abordado en este trabajo.

Conclusiones del estudio del arte

Finalizando con la revisión de literatura sobre el tema Desarrollo de un prototipo de aplicación móvil para la detección de enfermedades de la piel (como el vitíligo y psoriasis), utilizando técnicas de visión artificial apoyado en machine learning, nos ayudó para validar la implementación de una modelo de aprendizaje baso en la visión artificial, obteniendo como resultado positivo la detención temprana de enfermedades en la piel, este material bibliográfico se utilizará como referencia en la creación e implementación del presente trabajo de investigación.

Capítulo III

Metodología

El enfoque de este estudio es el desarrollo de un prototipo de aplicación móvil para la detección de enfermedades de la piel (como el vitíligo y psoriasis), utilizando técnicas de visión artificial apoyado en machine learning. Para lo cual se utilizará dos metodologías detalladas a continuación:

- Para el modelo de machine learning se utilizará una metodología KDD.
- Para el desarrollo del aplicativo móvil se desarrollará basado en la metodología Scrum

Metodología KDD

La metodología KDD (Knowledge Discovery in Databases) es un procedimiento empleado para la extracción automatizada de conocimiento a partir de extensas cantidades de datos. Este proceso se caracteriza por ser de naturaleza iterativa, lo que significa que se puede aplicar repetidamente hasta obtener la información requerida, adaptándose a las necesidades específicas

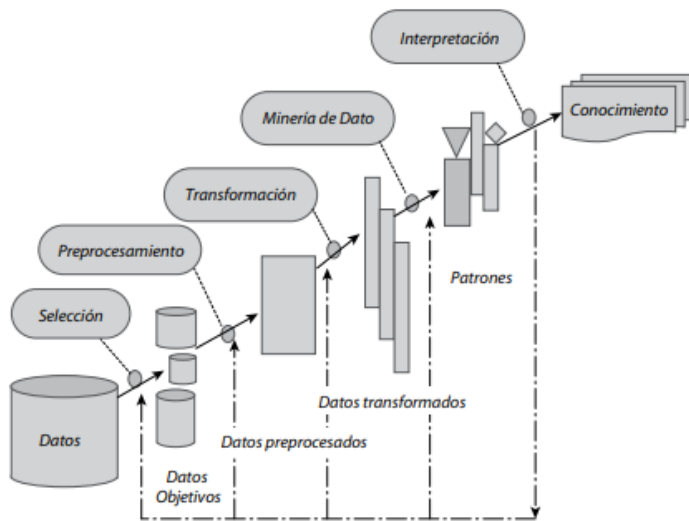
La metodología KDD es un proceso utilizado para llevar a cabo la extracción automatizada de conocimiento partiendo de grandes volúmenes de datos, el cual es de naturaleza iterativa, por lo tanto, es aplicable tantas veces como sea necesario hasta obtener la información necesaria.

Según (Hidalgo Troya, Alvarado Pérez, Timarán Pereira, Caicedo Zambrano, & Hernández Arteaga, 2018), se resume en las siguientes etapas:

- Selección
- Preprocesamiento/limpieza.
- Transformación/reducción.
- Minería de datos (data mining).
- Interpretación/evaluación.

Figura 3

Etapas del proceso KDD



Nota. En la figura se muestra las etapas para el proceso KDD. Obtenida de [https://ediciones.ucc.edu.co/index.php/ucc/catalog/download/36/40/230?inline=1#:~:text=El%20Descubrimiento%20de%20conocimiento%20en,que%20el%20usuario%20los%](https://ediciones.ucc.edu.co/index.php/ucc/catalog/download/36/40/230?inline=1#:~:text=El%20Descubrimiento%20de%20conocimiento%20en,que%20el%20usuario%20los%20)

El proceso de descubrimiento de conocimiento, conocido como KDD (por sus siglas en inglés), se compone de varias etapas fundamentales:

- **Selección de datos:** En esta etapa, se inicia identificando el conocimiento relevante y los objetivos del proceso de KDD desde la perspectiva del usuario final. Luego, se establece un conjunto de datos objetivo que puede ser la totalidad de los datos o una muestra representativa de los mismos, sobre el cual se realizará el proceso de descubrimiento.
- **Preprocesamiento/Limpieza de datos:** Durante esta fase, se lleva a cabo una evaluación de la calidad de los datos. Aquí se aplican operaciones básicas como la eliminación de datos ruidosos y la toma de decisiones sobre cómo abordar datos desconocidos, valores faltantes, valores nulos y duplicados. Los datos ruidosos son valores atípicos que se encuentran significativamente fuera del rango de valores esperados y pueden deberse a errores humanos, cambios en el sistema o a la falta

de información disponible en el momento adecuado.

- Transformación/Reducción de datos: En esta etapa, se buscan características útiles que permitan representar los datos de acuerdo a los objetivos del proceso. Se emplean técnicas de reducción de dimensiones o transformación para disminuir el número de variables bajo consideración.
- Minería de datos: El propósito de esta etapa es descubrir patrones de interés e inesperados mediante la aplicación de tareas de descubrimiento. Las técnicas de minería de datos generan modelos que pueden ser predictivos o descriptivos. Los modelos predictivos estiman valores futuros o desconocidos, mientras que los modelos descriptivos identifican patrones que explican o resumen los datos. Estos últimos se utilizan para explorar las propiedades de los datos examinados, no para predecir nuevos datos.
- Interpretación/Evaluación de datos: En esta fase, se interpreta los patrones descubiertos y, en ocasiones, se regresa a etapas anteriores para realizar iteraciones adicionales. Aquí se pueden incluir la visualización de los patrones extraídos, la eliminación de patrones redundantes o irrelevantes, y la traducción de patrones útiles en términos comprensibles para el usuario.

Capítulo IV

Desarrollo

Desarrollo del sistema

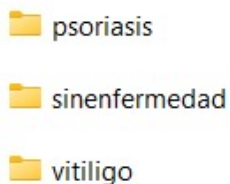
El Sistema de Clasificación de Enfermedades de la Piel mediante Aprendizaje Profundo está desarrollado en el lenguaje de programación, Python es una aplicación informática diseñada para diagnosticar y clasificar diferentes enfermedades de la piel a partir de imágenes digitales. Este sistema utiliza técnicas avanzadas de aprendizaje profundo, específicamente redes neuronales convolucionales (CNN), para lograr una clasificación precisa y automatizada, adicional se utiliza un modelo de red neuronal convolucional MiniVGGNet. Este modelo es una versión reducida del popular modelo VGGNet y se define mediante la función "MiniVGGNet.build()". El modelo consta de varias capas convolucionales y capas totalmente conectadas para la clasificación.

El proceso que se está desarrollando en el algoritmo es:

Recopilación y Preparación de Datos: Se recopilaron imágenes de enfermedades de la piel, específicamente de las clases "psoriasis", "sin_enfermedad" y "vitiligo". Estas imágenes se organizaron etiquetando de acuerdo con los tipos de enfermedades. Las imágenes se redimensionaron a un tamaño uniforme y se normalizaron para que los valores de píxeles estén en un rango entre 0 y 1 estas imágenes están clasificadas en 3 diferentes carpetas las cuales son:

Figura 4

Data set Clasificados



Nota: En cada carpeta se encuentra las imágenes para el aprendizaje centrándonos en las características de esta mismas.

Carga y Preprocesamiento de Datos: El código, carga un conjunto de imágenes de

enfermedades de la piel desde el directorio "dataset/enfermedad/". Las imágenes se redimensionan a un tamaño de 32x32 píxeles y se convierten en tensores para poder ser procesadas por el modelo. Además, se realiza la normalización de las imágenes al escalar los valores de píxeles entre 0 y 1. (Con la finalidad que el modelo tenga un mejor comportamiento se procedió a redimensionar y unificar el tamaño a 32 x 32 y se convierte en tensores para ser procesadas por el mismo.

División del Conjunto de Datos: El conjunto de datos se dividió en un conjunto de entrenamiento y un conjunto de pruebas. El 75% de los datos se utilizan para el entrenamiento y el 25% para la evaluación del modelo.

Codificación One-Hot: Las etiquetas de las clases se codifican en formato one-hot utilizando la clase "LabelBinarizer" de la biblioteca Scikit-learn. Esto convierte las etiquetas en vectores binarios, lo que facilita el entrenamiento del modelo para la clasificación.

Optimizador: Se utiliza el optimizador "Stochastic Gradient Descent" (SGD) para el entrenamiento del modelo. SGD es un algoritmo de optimización que ajusta los pesos del modelo para minimizar la función de pérdida durante el proceso de entrenamiento.

Entrenamiento del Modelo: El modelo comienza a aprender utilizando el conjunto de entrenamiento y se valida en el conjunto de pruebas durante 20 épocas. En cada época, se ajustan los pesos del modelo para reducir la pérdida y mejorar la precisión en la clasificación.

Evaluación del Modelo: Después del entrenamiento, se evalúa el resultado del modelo en el conjunto de pruebas y se muestra un informe de clasificación que incluye métricas como precisión, recall y F1-score para cada clase, así como la precisión global del modelo.

Guardado del Modelo en Formato TFLite: Finalmente, se guarda el modelo entrenado en un archivo con extensión ".tflite" en formato TensorFlow Lite, que es una versión optimizada del modelo para su uso en dispositivos móviles o en aplicaciones con recursos limitados.

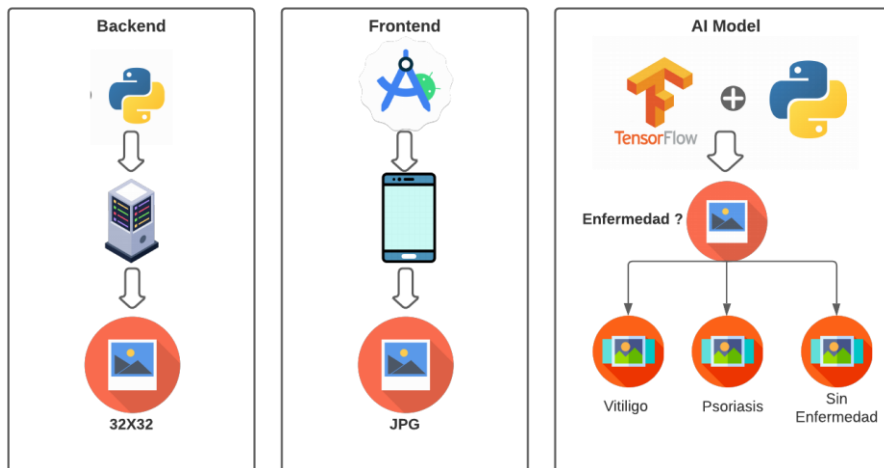
Despliegue y Uso: El modelo entrenado puede ser desplegado en un entorno de

producción para su uso en la clasificación automatizada de enfermedades de la piel.

Arquitectura

Figura 5

Arquitectura del proyecto



Nota: Se detalla el proceso dividido en capas para el funcionamiento de nuestro prototipo.

Backend: Desarrollar y entrenar un modelo de clasificación de enfermedades de la piel utilizando una red neuronal convolucional (CNN). El propósito es que este modelo pueda analizar imágenes de piel y predecir a qué categoría o clase pertenece la enfermedad presente en la imagen. Las clases en este contexto son "psoriasis", "sano" y "vitiligo", se llevan a cabo las siguientes etapas:

Carga y Preprocesamiento de Datos:

- Las imágenes se cargan desde el directorio utilizando `cv2.imread` y se redimensionan a un tamaño específico.
- Las imágenes se convierten en arrays y se normalizan en la sección donde se carga el data set y se realiza el preprocesamiento.

Figura 6

Cargar dataset

```
for directory_path in glob.glob(dataset_dir):
    label = directory_path.split("\\")[-1]
    for img_path in glob.glob(os.path.join(directory_path, "*.jpg")):
        img = cv2.imread(img_path)
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        img = cv2.resize(img, (SIZE, SIZE), interpolation = cv2.INTER_AREA)
        img = img_to_array(img)
        data.append(img)
        labels.append(label)
        if verbose > 0 and i > 0 and (i+1)%verbose == 0:
            print("[INFO] Procesado {}/{}".format(i+1, number_images))
        i += 1

data = np.array(data)
labels = np.array(labels)
```

Nota: En esta sección, se cargan las imágenes de enfermedades de la piel desde el directorio especificado y se realizan operaciones de preprocesamiento, como la conversión a formato RGB, redimensionamiento y conversión a arrays para su posterior procesamiento.

División en los Conjuntos de Entrenamiento y Prueba:

La división de los datos en conjuntos de entrenamiento y prueba se realiza usando `train_test_split` de la biblioteca Scikit-Learn.

Figura 7

Entrenamiento y Prueba

```
(x_train, x_test, y_train, y_test) = train_test_split(data,
                                                    labels,
                                                    test_size=0.25,
                                                    random_state=42)
```

Nota: Aquí se divide el conjunto de imágenes en conjuntos de entrenamiento y prueba para permitir la evaluación del modelo en datos no vistos durante el entrenamiento.

Codificación "One-Hot" de Etiquetas:

- Las etiquetas se codifican en formato "one-hot" utilizando `LabelBinarizer` de Scikit-Learn.

Figura 8

Convertir las etiquetas de enteros en vectores

```
lb = LabelBinarizer()
y_train = lb.fit_transform(y_train)
y_test = lb.fit_transform(y_test)
```

Nota: En esta etapa, las etiquetas de las clases se codifican en formato "one-hot" para que el modelo pueda comprender y procesar las diferentes categorías de enfermedades de la piel.

Diseño de la Arquitectura de la CNN:

- La arquitectura de la red neuronal convolucional (MiniVGG) se define en la sección donde se construye el modelo usando Keras.

Figura 9

Construir modelo y compilar

```
model = MiniVGGNet.build(width=32, height=32, depth=3, classes=3)
model.compile(loss="categorical_crossentropy", optimizer=opt, metrics=["accuracy"])
```

Nota: Se define la arquitectura de la red neuronal convolucional (MiniVGG) que se utilizará para el proceso de clasificación de imágenes de enfermedades de la piel.

Entrenamiento del Modelo:

- El modelo se compila con la función de pérdida y un optimizador de la sección donde se construye el modelo.
- El entrenamiento del modelo se realiza en la sección donde se utiliza `model.fit` para ajustar los pesos de la red.

Figura 10

Entrenamiento del Modelo de Clasificación de Enfermedades de la Piel

```
H = model.fit(x_train, y_train,
              validation_data=(x_test, y_test),
              batch_size=64,
              epochs=epocas,
              callbacks=callbacks,
              verbose=2)
```

Nota: En esta fase, el modelo se compila y se entrena utilizando el conjunto de entrenamiento. Los pesos de la red se ajustan iterativamente para aprender a identificar características distintivas en las imágenes.

Evaluación del Rendimiento del Modelo:

- La evaluación del modelo se realiza usando el conjunto de prueba y se calculan métricas como precisión, "recall" y "F1-score" en la sección donde se utiliza

```
model.predict.
```

Figura 11

Evaluación del Rendimiento del Modelo en Datos de Prueba

```
predictions = model.predict(x_test, batch_size=64)
print(classification_report(y_test.argmax(axis=1),
                             predictions.argmax(axis=1),
                             target_names=labelNames))
```

Nota: Aquí se evalúa el rendimiento del modelo utilizando métricas de clasificación, como precisión, "recall" y "F1-score", para determinar su capacidad para predecir con precisión las categorías de enfermedades de la piel.

Generación de Gráficos de Rendimiento:

- Los gráficos de pérdida y precisión se generan utilizando matplotlib y se guardan en archivos de imagen en la sección donde se crean los gráficos.

Figura 12

Visualización del Rendimiento de Entrenamiento y Validación

```
plot_dir = "logs/graficoenf.png"
plt.style.use("ggplot")
plt.figure()
plt.plot(np.arange(0, epocas), H.history["loss"], label="train_loss")
plt.plot(np.arange(0, epocas), H.history["val_loss"], label="val_loss")
plt.plot(np.arange(0, epocas), H.history["accuracy"], label="train_acc")
plt.plot(np.arange(0, epocas), H.history["val_accuracy"], label="val_acc")
plt.title("Training Loss and Accuracy on Piel")
plt.xlabel("Epoch #")
plt.ylabel("Loss/Accuracy")
plt.legend()
plt.savefig(plot_dir)
```

Nota: Se generan gráficos que representan la evolución de la pérdida y la precisión del modelo en función del número de épocas, lo que permite visualizar su aprendizaje y generalización.

Exportación del Modelo a Formato TFLite:

- El modelo entrenado se guarda en formato TFLite en la sección final del código utilizando el convertidor TFLite de TensorFlow.

Figura 13

Exportación del Modelo Entrenado a Formato TFLite

```
keras_model = tf.keras.models.load_model('logs/modelo3.h5')
converter = tf.lite.TFLiteConverter.from_keras_model(keras_model)
tfmodel = converter.convert()
file = open('enfermedad.tflite', 'wb')
file.write(tfmodel)
```


Nota: El modelo entrenado se exporta en formato TFLite, lo que facilita su implementación en plataformas y dispositivos diversos para aplicaciones en tiempo real.

Frontend: En esta interfaz los usuarios cargarán las imágenes para la clasificación y que permita visualizar las predicciones del modelo junto con la confianza. El modelo procesa la imagen de entrada y mediante sus técnicas predice de a qué categoría de enfermedad pertenece la misma. El código incluye funciones para cargar el modelo de TensorFlow Lite, convertir la imagen seleccionada en un ByteBuffer e interpretar la salida para producir una predicción.

Carga del modelo: La carga del archivo del modelo TensorFlow Lite (archivo tflite) se realiza en el método loadModelFile() dentro de la clase MainActivity en Android Studio.

Figura 14

Carga Archivo Tflite

```
private MappedByteBuffer loadModelFile() throws IOException {
    AssetFileDescriptor fileDescriptor = this.getAssets().openFd("enfermedad.tflite");
    FileInputStream inputStream = new FileInputStream(fileDescriptor.getFileDescriptor());
    FileChannel fileChannel = inputStream.getChannel();
    long startOffset = fileDescriptor.getStartOffset();
    long declaredLength = fileDescriptor.getDeclaredLength();
    return fileChannel.map(FileChannel.MapMode.READ_ONLY, startOffset, declaredLength);
}
```

Nota: Código fuente para la carga del archivo del modelo TensorFlow Lite la que se realiza en el método loadModelFile()

Selección de Imagen: Al iniciar la aplicación, mediante el botón "Elegir" se accede a la galería de imágenes del dispositivo, posteriormente se seleccionará una imagen que se desea someter a una prueba de clasificación.

Figura 15

Código fuente para elegir Imagen

```

btnElegir.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(Build.VERSION.SDK_INT == Build.VERSION_CODES.M){
            if(checkSelfPermission(Manifest.permission.READ_EXTERNAL_STORAGE) ==
                PackageManager.PERMISSION_DENIED){
                String[] permissions = {Manifest.permission.READ_EXTERNAL_STORAGE};
                requestPermissions(permissions, PERMISSION_CODE);
            }else{
                chooseImageFromGallery();
            }
        }else{
            chooseImageFromGallery();
        }
    }
});

```

Nota: En el gráfico se indica el código para la selección de la imagen.

Procesamiento de Imagen: Después de seleccionar la imagen, la aplicación muestra en un ImageView, antes de realizar la predicción, se redimensiona a las dimensiones requeridas por el modelo, utilizando la función `getResizedBitmap()`. Esto asegura que la imagen se ajuste al tamaño de entrada del modelo.

Predicción de Clasificación: Una vez que se tiene la imagen redimensionada, se utiliza el botón "Predecir". La aplicación toma la imagen redimensionada, la convierte en un formato adecuado para el modelo utilizando la función `convertBitmapToByteBuffer()`, y luego realiza una predicción utilizando el modelo TensorFlow Lite mediante la instancia del `Interpreter(separar)`.

Figura 16

Código fuente para la Predicción

```

btnPredecir.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        imgInput.invalidate();
        BitmapDrawable drawable = (BitmapDrawable) imgInput.getDrawable();
        Bitmap bitmap = drawable.getBitmap();
        Bitmap bitmap_resize = getResizedBitmap(bitmap, IMG_SIZE, IMG_SIZE); //bitmap.. ancho, altura
        convertBitmapToByteBuffer(bitmap_resize);

        tflite.run(imgData, result);
        txtPrediccion.setText("");
        String labels[] = {"psoriasis", "sinenfermedad", "vitiligo"};
        //txtPrediccion.setText("result= "+ Arrays.toString(result[0]));
        //txtPrediccion.setText("result= "+ argmax(result[0])+" normal=> "+Arrays.toString(result[0]));

        txtPrediccion.setText(labels[argmax(result[0])]+"\nProbs:"+Arrays.toString(result[0]));
    }
});

```

Nota: El procesamiento de la imagen incluye redimensionamiento, conversión a un formato adecuado.

Mostrar Resultados: Después de ejecutar la predicción, la aplicación determina la clase predicha y las probabilidades asociadas. La clase predicha se muestra en el TextView llamado txtPrediccion. Además, las probabilidades para cada clase se presentan en formato de arreglo.

Como se muestra en la Figura 9, la muestra de resultados, incluyendo la clase predicción y las probabilidades asociadas, se realiza en la función `btnPredecir.setOnClickListener()`. Después de ejecutar el modelo TensorFlow Lite y obtener los resultados de la predicción, estos resultados se muestran en el TextView llamado `txtPrediccion`.

En esta parte del código, el modelo ha realizado la predicción; el resultado se pasa a la función `argmax()` para obtener el índice de la clase predicha con la probabilidad más alta. Luego, se utiliza ese índice para obtener la etiqueta de clase correspondiente del arreglo `labels[]`. Además, se muestran las probabilidades asociadas con cada clase en forma de arreglo.

Pruebas Iterativas: Se puede repetir este proceso de selección, redimensionamiento, predicción y visualización de resultados para diferentes imágenes tantas veces como se desee. Esto permitirá probar cómo el modelo clasifica diferentes tipos de imágenes.

Interpretación de Resultados: Una vez que se muestra el resultado de la predicción, se analiza cómo el modelo ha clasificado la imagen. Si las probabilidades son altas y la clase predicha es coherente con lo que esperabas para esa imagen, entonces el modelo está funcionando correctamente. Si las predicciones no son precisas, podría ser necesario ajustar o mejorar el modelo.

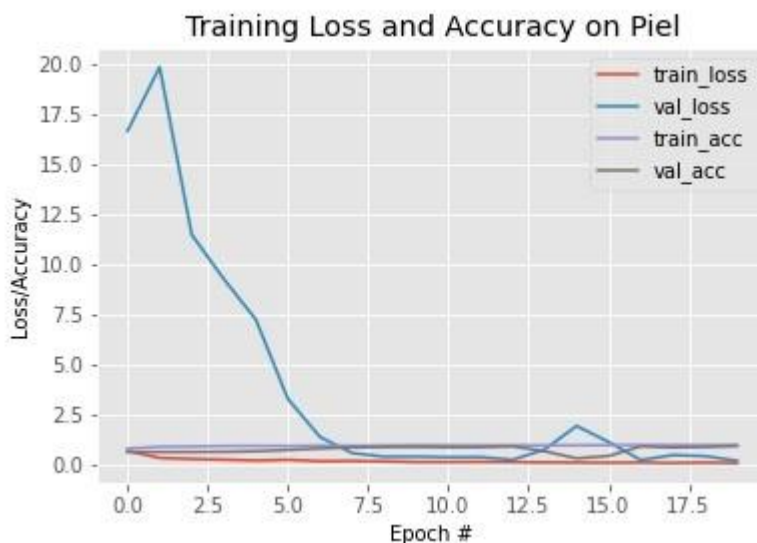
Capítulo V

Pruebas y Análisis de Resultados

Tras culminar el desarrollo del sistema, se procedió a realizar una serie de evaluaciones con el propósito de asegurar la satisfacción de todos los criterios requeridos. En esta medida, se llevaron a cabo pruebas dedicadas a medir tanto el desempeño como la usabilidad de la plataforma, con las pautas establecidas por la norma ISO 25010. Los detalles acerca de los resultados de estas evaluaciones se presentan en la siguiente sección.

Figura 17

Training Loss and Accuracy



Nota: Estos valores son métricas clave para evaluar el rendimiento y la capacidad de generalización de un modelo de aprendizaje automático durante el entrenamiento y la validación.

Al finalizar la primera época, se muestra la pérdida (loss) en el conjunto de validación (val_loss). En este caso, la pérdida en el conjunto de validación se redujo de un valor inicialmente infinito a 3.09344.

Durante la primera época, se procesaron un total de 17 lotes de entrenamiento. Cada lote tomó aproximadamente 5 segundos para completarse. Se muestra la pérdida y precisión (accuracy) obtenida en el conjunto de entrenamiento: la pérdida es 0.5962 y la precisión es 0.8127. También se muestra la pérdida y precisión en el conjunto de validación:

la pérdida es 3.0934 y la precisión es 0.6772. Además, se muestra el tiempo promedio por época y por lote (5 segundos por época, 279 milisegundos por lote).

Al finalizar la segunda época, se muestra que la pérdida en el conjunto de validación (`val_loss`) no mejoró en comparación con la época anterior.

Durante la segunda época, se procesaron 17 lotes de entrenamiento nuevamente. Cada lote tomó aproximadamente 3 segundos para completarse. Se muestra la pérdida y precisión en el conjunto de entrenamiento y validación, similar al registro de la primera época.

El proceso continúa para cada una de las 20 épocas, donde se muestra la pérdida y precisión en cada conjunto, si la pérdida en el conjunto de validación mejoró en comparación con la época anterior, y el tiempo promedio por época y por lote

Teniendo como resultado:

Pérdida en el conjunto de validación: La pérdida en el conjunto de validación disminuyó a medida que avanzaban las épocas. Al principio, la pérdida en el conjunto de validación fue muy alta (inicialmente infinita), pero después de algunas épocas, se redujo significativamente, lo que indica que el modelo se ajustó mejor a los datos y redujo el error en las predicciones.

Precisión en el conjunto de validación: La precisión en el conjunto de validación también mejoró con el tiempo. Comenzó en torno al 67.7% en la primera época y aumentó hasta aproximadamente 92.8% en la última época. Esto señala que a medida que progresaba el proceso de entrenamiento, las predicciones del modelo adquirieron una mayor precisión.

Tendencia general: A lo largo de las épocas, el modelo muestra una mejora constante en la precisión y la pérdida en el conjunto de validación, lo que sugiere que está aprendiendo de manera efectiva y mejorando su capacidad de generalización para clasificar nuevas imágenes.

Es crucial considerar que la evaluación del desempeño del modelo también se lleva a cabo en el conjunto de pruebas, con el propósito de obtener una estimación más precisa

de su habilidad para categorizar información que no ha sido previamente observada. En la tabla de resultados, vemos que la precisión general en el conjunto de pruebas es de aproximadamente 93%, lo que respalda la afirmación de que el modelo mejoró su aprendizaje y logró una buena capacidad de clasificación en datos nuevos.

Tabla 2

Matriz de clasificación multiclase

	Precision	Recall	f1-score	Support
Psoriasis	0,78	1	0,88	79
No tiene enfermedad	0,99	1	0,99	216
Vitiligo	1	0,52	0,68	52

Nota: Esta matriz es útil para visualizar el rendimiento del modelo en cada clase individual, permitiendo analizar en qué clases el modelo tiene un buen rendimiento y en cuáles necesita mejorar.

El modelo de red neuronal convolucional (CNN) implementado tuvo un buen desempeño en la tarea de clasificación de imágenes de enfermedades de la piel. A continuación, se resume el rendimiento del modelo:

La precisión global (accuracy) en el conjunto de pruebas: 93%. Esto significa que aproximadamente el 93% de las predicciones del modelo fueron correctas en el conjunto de pruebas.

En general, el modelo demostró una alta capacidad para clasificar la clase "sano", con una precisión de 99%, lo que indica que fue muy preciso en identificar las imágenes de piel sana.

También tuvo un buen desempeño en la clasificación de la clase "psoriasis" con una precisión del 78%. Sin embargo, el recall (sensibilidad) de esta clase fue del 100%, lo que significa que todas las muestras reales de "psoriasis" fueron correctamente identificadas.

La clase "vitiligo" fue la que presentó un rendimiento inferior en comparación con las otras dos clases. Aunque el modelo logró una precisión del 100%, el recall fue del 52%, lo que indica que no logró identificar todas las muestras reales de "vitiligo".

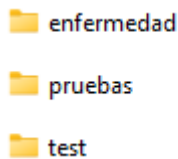
A lo largo de las 20 épocas de entrenamiento, el modelo mostró una mejora constante en su capacidad de clasificación, ya que la pérdida en el conjunto de validación se redujo significativamente.

Para las primeras pruebas se realizó lo siguiente:

Selección Aleatoria de Imágenes: El script selecciona aleatoriamente 10 imágenes de prueba del directorio especificado (`dir_imagenes`). Esto podría ser útil para obtener una vista rápida de cómo el modelo se desempeña en una muestra pequeña pero diversa de imágenes.

Figura 18

Directorio Imágenes para el test



Nota: Para una mejor validación de nuestro modelo inicialmente probamos con imágenes en carpetas separadas.

Procesamiento de Imágenes: Cada imagen se procesa para asegurarse de que tenga el mismo tamaño antes de ingresar al modelo. Se redimensiona a 32x32 píxeles y se convierte de BGR a RGB para que coincida con el formato esperado por la mayoría de las redes neuronales convolucionales.

Normalización de Datos: Las imágenes se normalizan dividiendo sus valores de píxeles entre 255.0. Esto escala los valores de píxeles al rango [0, 1], lo que es común en la mayoría de los casos de entrenamiento de modelos de aprendizaje profundo.

Carga del Modelo Preentrenado: El modelo MiniVGG preentrenado se carga desde el archivo especificado (`dir_model`). El uso de un modelo preentrenado permite aprovechar el conocimiento adquirido durante un entrenamiento anterior en un conjunto de datos más grande y puede mejorar la precisión de las predicciones en comparación con entrenar desde cero en un conjunto de datos pequeño.

Figura 19

Archivo pre entrenado para realizar las pruebas



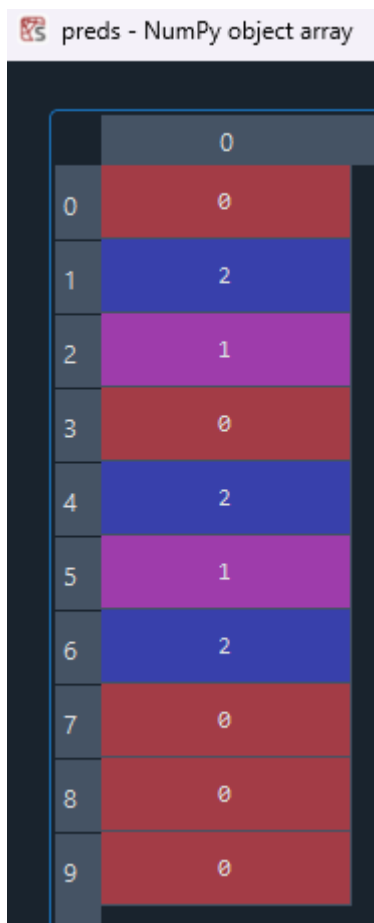
modelo3.h5

Nota: Archivo que utilizaremos para las pruebas de validación

Predicciones y Etiquetas: El modelo se utiliza para predecir las etiquetas de clase para las imágenes de prueba. Las etiquetas predichas se almacenan en la variable preds. Cada etiqueta el valor de índice correspondiente a la clase con la probabilidad más elevada para una imagen determinada.

Figura 20

Clasificación etiquetas de clase



	0
0	0
1	2
2	1
3	0
4	2
5	1
6	2
7	0
8	0
9	0

Nota: Se tienen los valores 0,1 y 2 donde el valor 0 corresponde a psoriasis el valor 1 a sano y el valor 2 ha vitíligo.

Visualización de Resultados: El script recorre cada imagen de prueba y muestra la imagen junto con la etiqueta predicha utilizando OpenCV. Esto permite verificar visualmente cómo se comparan las predicciones del modelo con las imágenes reales.

Interpretación de Resultados: Al observar las imágenes junto con las etiquetas predichas, puedes evaluar cómo está funcionando el modelo en estas muestras específicas. Si las etiquetas predichas son coherentes con lo que se espera de las imágenes (por ejemplo, una imagen de psoriasis se clasifica como "psoriasis"), entonces el modelo parece estar funcionando bien en esta selección aleatoria de imágenes.

Figura 21

Prueba de clasificación de Imágenes caso vitíligo



Figura 22

Prueba de clasificación de Imágenes caso psoriasis

**Figura 23**

Prueba de clasificación de Imágenes caso no favorable



Nota: Visualización de una imagen de un paciente que tiene vitiligo sin embargo tiene psoriasis esta clasificación de imagen no fue favorable para la prueba.

Pruebas con la Interfaz de Usuario

La funcionalidad de la aplicación está diseñada para realizar pruebas de clasificación de imágenes utilizando un modelo Tensor Flow Lite pre entrenado.

Selección de Imagen: Al iniciar la aplicación, podemos hacer clic en el botón "Elegir" para acceder a la galería de imágenes de tu dispositivo. Allí, seleccionas una imagen que deseas realizar la prueba de clasificación.

Figura 24

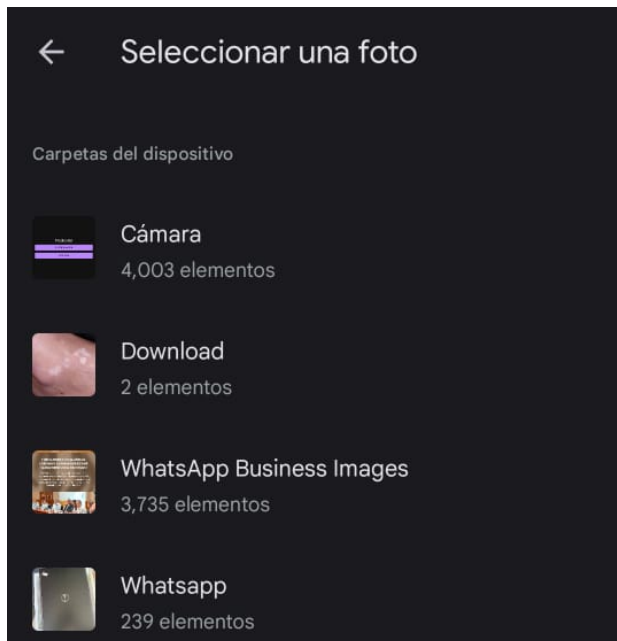
Interfaz Inicial APP



Nota: Interfaz para la selección de la Imagen a clasificar.

Figura 25

Selección Foto a Clasificar

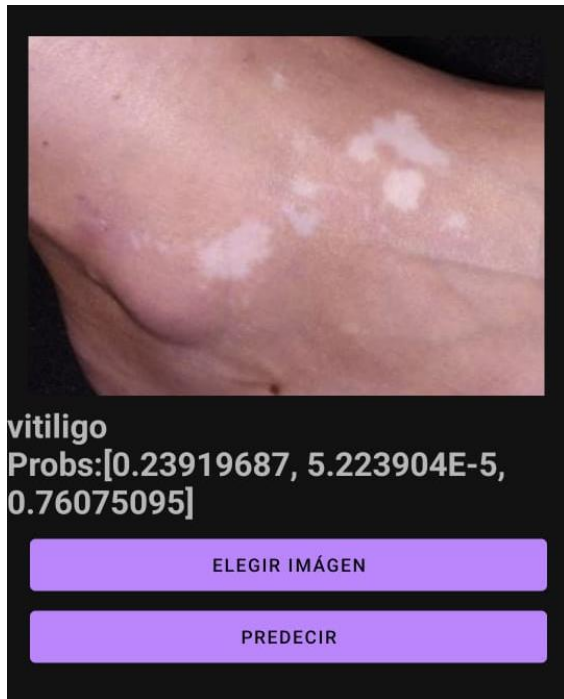


Nota: Selección de la Imagen a Clasificar en este caso iniciaremos por Vitfligo.

Antes de realizar la predicción, la imagen se redimensiona a las dimensiones requeridas por el modelo utilizando la función `getResizedBitmap()`. Esto asegura que la imagen se ajuste al tamaño de entrada del modelo.

Figura 26

Resultados caso vitiligo



Nota: La aplicación determina la clase predicha y las probabilidades asociadas en este caso correspondiente a un 23%.

Figura 27

Resultados caso psoriasis



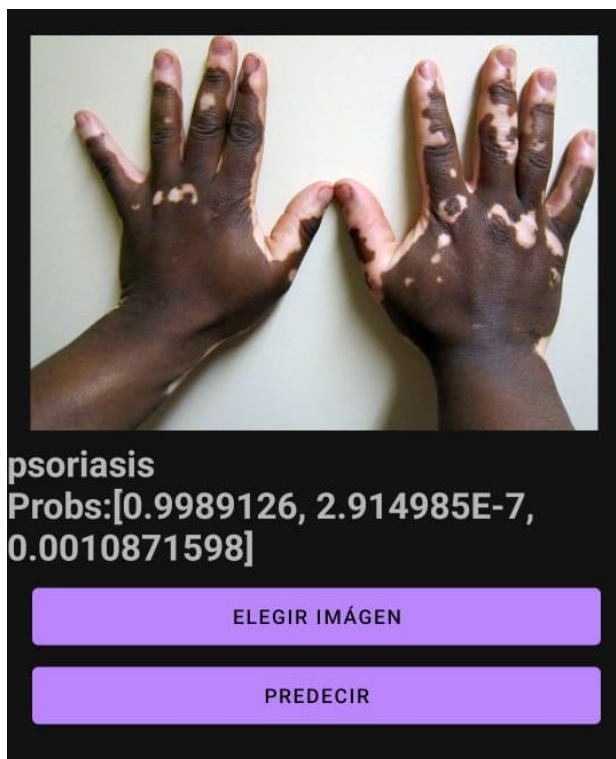
Nota: La aplicación determina la clase predicha y las probabilidades asociadas en este caso correspondiente a un 99%.

Porque varía entre un 23% de acierto en vitíligo contra un 99% de acierto en psoriasis esto corresponde a varios factores en la imagen y como el modelo lo está aprendiendo en el caso de vitíligo la variación de colores en la piel es leve igualmente lo clasifica de manera correcta en el caso de psoriasis la imagen es más precisa, así como los detalles que corresponden a esta enfermedad.

Probaremos con una imagen para el vitíligo con el que realizamos las pruebas

Figura 28

Resultados caso vitíligo imagen más detalla



Nota: En este casi tenemos un 99% de acierto ya que se nota las variaciones del tono de piel.

Los resultados obtenidos de la clasificación de enfermedades de la piel utilizando el modelo de red neuronal convolucional (CNN) son bastante alentadores. La matriz de clasificación multiclase muestra que el modelo es capaz de identificar y clasificar las clases "psoriasis", "sin enfermedad" y "vitíligo" con una precisión general del 93%.

En detalle, el modelo logra una precisión del 78% para la clase "psoriasis", indicando que el 78% de las predicciones para esta clase son correctas. Para la clase "sinenfermedad", la precisión es del 99%, lo que significa que el modelo tiene un alto nivel de confianza en sus predicciones para esta categoría. Sin embargo, la clase "vitiligo" presenta un rendimiento ligeramente inferior, con una precisión del 100% pero un recall del 52%. Esto sugiere que el modelo tiene dificultades en identificar correctamente todos los casos de "vitiligo".

La evaluación del rendimiento también revela que el modelo es sólido en términos de exactitud general (accuracy), alcanzando un 93%. Las métricas como el recall y la puntuación F1 varían según la clase, indicando una cierta desigualdad en el rendimiento entre las categorías.

En resumen, el modelo de CNN logra una precisión y rendimiento general satisfactorio en la clasificación de enfermedades de la piel. Aunque la clase "vitiligo" podría beneficiarse de mejoras adicionales, en conjunto, el modelo demuestra su capacidad para diagnosticar con precisión diversas enfermedades cutáneas en base a las imágenes proporcionadas.

Porque se puede dar la baja predicción para la clase "vitiligo" en comparación con las otras clases esto es por las características visuales que distinguen el "vitiligo" pueden ser más sutiles o difíciles de captar para el modelo en comparación con las otras enfermedades. Si el modelo no puede capturar adecuadamente estas características, su rendimiento en la clasificación de "vitiligo" será menor.

Conclusiones y Recomendaciones

Conclusiones

En base a los resultados obtenidos con una población de más de 1500 imágenes, el modelo logra un porcentaje de precisión de 90%, debido al porcentaje de error que se encuentra a la detección de la enfermedad "vitiligo", con el desarrollo del modelo de CNN logra una precisión y rendimiento general satisfactorio en la clasificación de enfermedades de la piel. Aunque la clase "vitiligo" podría beneficiarse de mejoras adicionales, en conjunto, el modelo demuestra su capacidad para diagnosticar con precisión diversas enfermedades cutáneas en base a las imágenes proporcionadas.

En general, el modelo muestra un rendimiento satisfactorio en la clasificación de las enfermedades de la piel. La precisión general indica que el modelo es capaz de realizar diagnósticos precisos en la mayoría de los casos.

El modelo tiene dificultades para identificar correctamente todos los casos de "vitiligo". Esto puede deberse a la naturaleza sutil o difícil de captar de las características de esta enfermedad.

El lenguaje de programación Python es un lenguaje de alto nivel, ya que, posee una amplia gama de bibliotecas de datos y fuentes con algoritmos establecidos, permitiendo módulos menos extensos por el uso de Inteligencia Artificial entre ellos.

El uso de la metodología ágil, en este caso SCRUM, nos permitió un desarrollo sólido, y eficiente entre los desarrollares y los usuarios que probaron la aplicación.

Recomendaciones

Se aconseja recopilar más imágenes y datos para la clase "vitiligo" y cualquier otra clase con un número limitado de muestras. Esto ayudará a equilibrar el conjunto de datos y proporcionará al modelo una mejor comprensión de las características únicas de cada enfermedad.

Experimentar con diferentes configuraciones de hiperparámetros, como tasas de aprendizaje, número de capas y tamaños de lotes, puede mejorar la capacidad del modelo para aprender patrones complejos y generalizar mejor.

El rendimiento del modelo debe monitorearse continuamente y refinarse con nuevos datos y mejoras técnicas a medida que estén disponibles esto apoyado de expertos médicos, ya que nos ayudaran a validar las predicciones del modelo con pruebas reales para asegurarse de que las predicciones sean consistentes con el diagnóstico humano.

Generar un Dataset más amplio que permita validar el funcionamiento del sistema. Del mismo modo se debe aumentar la población de imágenes de entrenamiento para obtener mayor precisión en las detecciones de las enfermedades.

Trabajos futuros

Una mejora de la clasificación continuar ajustar y optimizar el modelo actual, experimentando con diferentes arquitecturas de CNN, hiperparámetros y técnicas de regularización para lograr un rendimiento aún mejor en la clasificación de todas las clases.

Centrarse en mejorar el rendimiento en clases específicas que presentan desafíos, como "vitiligo". Investigar métodos específicos para captar mejor las características sutiles de estas enfermedades.

Desarrollar una interfaz de usuario más amigable que permita a los médicos cargar imágenes y recibir predicciones del modelo, además de proporcionar información adicional sobre las enfermedades.

Explorar la posibilidad de utilizar modelos pre-entrenados y técnicas de transferencia de aprendizaje para aprovechar el conocimiento previo de modelos exitosos en problemas similares.

Bibliografía

- Basogain Olabe, X. (2015). REDES NEURONALES ARTIFICIALES. En X. Basogain Olabe, *REDES NEURONALES ARTIFICIALES Y SUS APLICACIONES*. España.
- Cárdenas Vera, M. F., Llerena Pizarro, O. R., & Calle, E. (2012). Automatización de un sistema de centrado de componentes utilizando visión artificial. Cuenca, Ecuador. Obtenido de <https://dspace.ups.edu.ec/bitstream/123456789/1109/14/UPS-CT002210.pdf>
- Das, S. (2020). Psoriasis. Obtenido de <https://www.msmanuals.com/es-es/hogar/trastornos-de-la-piel/psoriasis-y-trastornos-de-descamaci%C3%B3n/psoriasis>
- Hidalgo Cajo, I. M., Yasaca Pucuna, S., Hidalgo Cajo, B. G., Hidalgo Cajo, D., & Latorre Benalcázar, N. B. (2020). Estudio comparativo de los algoritmos backpropagation (bp) y multiple linear regression (mlr) a través del análisis estadístico de datos aplicado a redes neuronales artificiales. Ecuador. Obtenido de <https://revista.redipe.org/index.php/1/article/view/939>
- Hidalgo Troya, A., Alvarado Pérez, J., Timarán Pereira, S., Caicedo Zambrano, S., & Hernández Arteaga, I. (2018). *Descubrimiento de patrones de desempeño académico con árboles de decisión en las competencias genéricas de la formación profesional*. Obtenido de <https://ediciones.ucc.edu.co/index.php/ucc/catalog/book/36>
- LeCun, Y., Bengio, Y., & Geoffrey, H. (2015). *Deep learning*. Obtenido de Nature: <https://www.nature.com/articles/nature14539>
- Mohammad, J., Ebrahim, N.-E., Nader, K., Reza Soroushmehr, S. M., Shadrokh, S., & Kayvan, N. (2016). Extraction of Skin Lesions from Non-Dermoscopic Images Using Deep Learning. Arxiv. Obtenido de <https://arxiv.org/abs/1609.02374>
- Molé, M., & Coringrato, M. (2019). *Actualización sobre vitiligo*. Obtenido de <http://www.dermatolarg.org.ar/index.php/dermatolarg/article/view/1874>
- Tamayo Otero, R. A., Prieto Cuesta, C. D., Villegas Llangary, G. M., & Menéndez Reyes, B. E. (2022). Factores de Riesgo y desencadenantes de la Psoriasis: estudio de

revisión. Guayaquil, Ecuador. Obtenido de <https://jah-journal.com/index.php/jah/article/view/87/184>