



**Universidad Nacional Mayor de San Marcos**

**Universidad del Perú. Decana de América**

**Facultad de Ingeniería de Sistemas e Informática**

**Escuela Profesional de Ingeniería de Sistemas**

**Aplicación de la inteligencia artificial en la  
dermatología para la detección temprana del cancer de  
piel**

**TESIS**

**Para optar el Título Profesional de Ingeniero de Sistemas**

**AUTOR**

**Jesus Alfonso CUADROS CARRASCO**

**ASESOR**

**Víctor Hugo BUSTAMANTE OLIVERA**

**Lima, Perú**

**2024**



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

## Referencia bibliográfica

---

Cuadros, J. (2024). *Aplicación de la inteligencia artificial en la dermatología para la detección temprana del cancer de piel*. [Tesis de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Sistemas]. Repositorio institucional Cybertesis UNMSM.

---

### Metadatos complementarios autor/ asesor

|                                  |                                                                                           |
|----------------------------------|-------------------------------------------------------------------------------------------|
| <b>Datos de autor</b>            |                                                                                           |
| Nombres y apellidos              | Jesus Alfonso Cuadros Carrasco                                                            |
| Tipo de documento de identidad   | DNI                                                                                       |
| Número de documento de identidad | 71122396                                                                                  |
| URL de ORCID                     | <a href="https://orcid.org/0009-0006-7653-3251">https://orcid.org/0009-0006-7653-3251</a> |
| <b>Datos de asesor</b>           |                                                                                           |
| Nombres y apellidos              | Victor Hugo Bustamante Olivera                                                            |
| Tipo de documento de identidad   | DNI                                                                                       |
| Número de documento de identidad | 25655590                                                                                  |
| URL de ORCID                     | <a href="https://orcid.org/0000-0001-8805-9629">https://orcid.org/0000-0001-8805-9629</a> |
| <b>Datos del jurado</b>          |                                                                                           |
| <b>Presidente del jurado</b>     |                                                                                           |
| Nombres y apellidos              | Jorge Luis Chávez Soto                                                                    |
| Tipo de documento                | DNI                                                                                       |
| Número de documento de identidad | 08675814                                                                                  |
| <b>Miembro del jurado 1</b>      |                                                                                           |
| Nombres y apellidos              | John Ledgard Trujillo Trejo                                                               |
| Tipo de documento                | DNI                                                                                       |
| Número de documento de identidad | 06187585                                                                                  |
| <b>Datos de investigación</b>    |                                                                                           |
| Línea de investigación           | Inteligencia Artificial                                                                   |

|                                                        |                                                                                                     |
|--------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| Grupo de investigación                                 | N/A                                                                                                 |
| Agencia de financiamiento                              | Sin fines de Lucro                                                                                  |
| Ubicación geográfica de la investigación               | Lima, Lima, Puente Piedra                                                                           |
| Año o rango de años en que se realizó la investigación | 2024                                                                                                |
| URL de disciplinas OCDE                                | <a href="https://purl.org/pe-repo/ocde/ford#2.02.04">https://purl.org/pe-repo/ocde/ford#2.02.04</a> |



**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**  
**FACULTAD DE INGENIERIA DE SISTEMAS E INFORMATICA**  
**Escuela Profesional de Ingeniería de Sistemas**

**Acta Virtual de Sustentación de Tesis**

Siendo las 10:02 horas del día 19 de diciembre del año 2024, se reunieron virtualmente los docentes designados como miembros de Jurado de Tesis, presidido por el Lic. Jorge Luis Chávez Soto Lic. John Ledgard Trujillo Trejo (Miembro) y el Lic. Víctor Hugo Bustamante Olivera Vásquez (Miembro Asesor), usando la plataforma Meet ( <https://meet.google.com/spj-nraa-yri>), para la sustentación Virtual de la tesis Intitulada: **“APLICACIÓN DE LA INTELIGENCIA ARTIFICIAL EN LA DERMATOLOGÍA PARA LA DETECCIÓN TEMPRANA DEL CANCER DE PIEL”**, del Bachiller: **Cuadros Carrasco Jesus Alfonso**; para obtener el Título Profesional de Ingeniero de Sistemas.

Acto seguido de la exposición de la Tesis, el Presidente invitó al Bachiller a responder las preguntas formuladas por los Miembros del Jurado.

El Bachiller, en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez las preguntas formuladas por los señores miembros del Jurado.

Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, el bachiller obtuvo la nota de 19 ( diecinueve )

A continuación, el Presidente del Jurado Lic. Jorge Luis Chávez Soto, declara al Bachiller **Ingeniero de Sistemas**.

Siendo 10:49 horas, se levantó la sesión.

Lic. Jorge Luis Chávez Soto  
Presidente

Lic. John Ledgard Trujillo Trejo  
Miembro

Lic. Víctor Hugo Bustamante Olivera  
Miembro Asesor



**Universidad Nacional Mayor de San Marcos**

Universidad del Perú. Decana de América

**Vicerrectorado de Investigación y Posgrado**



## **CERTIFICADO DE SIMILITUD**

Yo **VICTOR HUGO BUSTAMANTE OLIVERA** en mi condición de **asesor** acreditado con la Resolución Directoral N° 000023-2024-EPISI-FISI/UNMSM de la **tesis/monografía/informe** de investigación/trabajo académico, cuyo título es: **APLICACIÓN DE LA INTELIGENCIA ARTIFICIAL EN LA DERMATOLOGÍA PARA LA DETECCIÓN TEMPRANA DEL CANCER DE PIEL**, presentado por el **Bachiller/magíster/egresado/licenciado/estudiante Cuadros Carrasco Jesus Alfonso** para optar el grado/título/especialidad de **INGENIERO DE SISTEMAS**, CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante el software de similitud textual, el documento evaluado cuenta con el porcentaje de 5% de similitud, nivel **PERMITIDO** para continuar con los trámites correspondientes y para su publicación en el repositorio institucional.

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del grado/ título/ especialidad correspondiente.

Firma del Asesor \_\_\_\_\_



DNI: 25655590

Nombres y apellidos del asesor: Victor Hugo Bustamante Olivera

## **DEDICATORIA**

El presente trabajo va dedicado a todas las personas que por importunos de la vida no fueron tratadas de la manera correcta por un centro de salud o que en su defecto los especialistas no pudieron hacerse abasto para poder atenderles.

También dedico este trabajo a las autoridades, docentes y personas que alguna vez fueron parte de mi crecimiento profesional y dan como culmine este trabajo donde se busca hacer un bien a la sociedad.

## **AGRADECIMIENTO**

Agradezco primeramente a Dios por darme la oportunidad de haber seguido en pie pese a tantas adversidades que se presentaron tanto internas como externas a mí.

De la misma manera agradezco a mis familiares y seres queridos que confiaron en mí para hasta el día de hoy. Que a pesar del largo trajín siguen poniendo sus esperanzas en mí y este trabajo es el reflejo de todo lo que un día cultivaron en mi vida.

## RESUMEN

Debido a la alta tasa de cáncer de la piel el presente trabajo busca combatir la escasez de profesionales en dermatología frente a su alta demanda mediante la creación de una inteligencia artificial que ayuda a su detección temprana. Se basa en el Machine Learning por medio de las iteraciones de épocas y redes neuronales convolucionales, las cuales brindan un diagnóstico benigno anticipado, que permite ayudar a muchas personas. Se consideran diversos parámetros como limitar el ámbito de estudio a dos tipos principales de cáncer y utilizamos un buen dataset proveniente de Kaggle dado que era necesario una buena cantidad de imágenes para poder preprocesarlas haciendo frente a la amplia variedad de casos que se presentan día a día.

El producto final de la tesis es una herramienta óptima para el diagnóstico del cáncer tanto desde un punto de vista social como económico. Para su desarrollo se utilizó el lenguaje Python y la librería PyTorch. En el apartado estadístico se usó Tableau, RapidMiner y Knime previo al análisis. Los resultados finales y el dataset fueron analizados en Kaggle.

Desde el punto de vista tecnológico y medico se pudo conseguir un precedente para múltiples estudios relacionados al cáncer de la piel.

**PALABRAS CLAVE:** Redes Neuronales, cáncer de piel, dataset, inteligencia artificial, preprocesamiento de imágenes

## **ABSTRACT**

Due to the high rate of skin cancer, this work seeks to combat the shortage of professionals in dermatology in the face of its high demand by creating artificial intelligence that helps in its early detection. It is based on Machine Learning through epoch iterations and convolutional neural networks, which provide an early benign diagnosis, which helps many people. Various parameters are considered, such as limiting the scope of study to two main types of cancer and we used a good dataset from Kaggle since a good number of images were necessary to be able to preprocess them to deal with the wide variety of cases that arise every day.

The final product of the thesis is an optimal tool for the diagnosis of cancer from both a social and economic point of view. For its development, the Python language and the PyTorch library were used. In the statistical section, Tableau, RapidMiner and Knime were used prior to the analysis. The results and the dataset were analyzed in Kaggle.

From a technological and medical point of view, it was possible to achieve a precedent for multiple studies related to skin cancer.

**KEY WORDS:** Neural Networks, skin cancer, dataset, artificial intelligence, image preprocessing

# INTRODUCCIÓN

Los estudios referentes a la Inteligencia Artificial van en aumento en los últimos años, de la mano con estos los problemas en salud han tomado un rol protagónico debido a lo sucedido con el COVID-19. En este escenario y frente a la máxima actividad Solar a la que estamos afrontando los casos del Cáncer de la piel están en aumento. Bajo este contexto se creará un software que utiliza la inteligencia artificial para la detección temprana de casos benignos de Nevo Displásico y Queratosis Actínica. Esto por medio de redes neuronales convolucionales que irán aprendiendo el patrón por medio de épocas, aumentando así su eficiencia a la hora de dar un diagnóstico.

Existen diferentes factores que influyen en la incidencia del cáncer de la piel los cuales se tendrán en cuenta, limitándonos a los tipos de cáncer de la piel antes mencionados. El estudio se basa en diversas documentaciones que abarcan el tema desde el punto de vista médico, referente al Cáncer de la piel, matemático, donde se explorara el modelo ideal para la inteligencia artificial y tecnológico, para el uso de las nuevas tecnologías emergentes que agilizan el desarrollo de un software que usa una caja negra donde la eficiencia ira aumentando considerablemente hasta el final del proyecto. Siendo así que se determinaran los términos necesarios para el desarrollo de todo el proyecto.

Sera menester para esta tarea el contar con una buena base de imágenes para su entrenamiento, un sistema de clasificación acorde al modelo planteado contando con puntos de vista tanto médicos como tecnológicos de manera que podamos pronosticar de forma eficiente os casos benignos. También se deben de tener en cuenta los temas éticos que conlleva el uso de la inteligencia artificial al estar tratando con posibles personas enfermas de Cáncer de piel. El uso de herramientas similares nos será de ayuda para generar este software en Python, PyTorch y Tensorflow teniendo en cuenta tanto precedentes nacionales como internacionales.

Las herramientas que se utilizarán serán Kaggle como plataforma donde se codifica el proyecto, donde se almacena el dataset y se generan los resultados. También será necesario usar RapidMiner, Tableau y Knime para las estadísticas previas al desarrollo del software y comparar como hay una mejoría al finalizar. Gracias a las librerías que contamos con Python y los datos de entrenamiento podremos ir generando diversos reportes con graficas que reflejan el avance de la inteligencia artificial. Kaggle también cuenta con librerías donde lo más destacable es el tema de preprocesamiento de imágenes y diferentes técnicas que ayudaran a eliminar la velloosidad de las muestras. Con todo eso las Redes Neuronales ya entrenadas generaran por medio de épocas un resultado optimo.

# ÍNDICE

## CAPÍTULO 1 PLANTEAMIENTO METODOLÓGICO .... 10

|                                                |    |
|------------------------------------------------|----|
| 1.1 ANTECEDENTES DEL PROBLEMA .....            | 10 |
| 1.2 DEFINICIÓN O FORMULACIÓN DEL PROBLEMA..... | 12 |
| 1.3 IMPORTANCIA DEL PROBLEMA .....             | 12 |
| 1.4 OBJETIVOS .....                            | 14 |
| 1.4.1 <i>Objetivo General</i> .....            | 14 |
| 1.4.2 <i>Objetivos Específicos</i> .....       | 14 |
| 1.5 ALCANCES Y LIMITACIONES.....               | 14 |
| 1.6 JUSTIFICACIÓN .....                        | 15 |
| 1.6.1 <i>Justificación Social</i> .....        | 15 |
| 1.6.2 <i>Justificación Técnica</i> .....       | 16 |
| 1.6.3 <i>Justificación Económica</i> .....     | 16 |

## CAPÍTULO 2 MARCO TEÓRICO..... 17

|                                                                                      |    |
|--------------------------------------------------------------------------------------|----|
| 2.1 DEFINICIONES Y CONCEPTOS .....                                                   | 17 |
| 2.1.1 <i>Artificial Intelligence (AI) o IA (inteligencia artificial):</i> .....      | 17 |
| 2.1.2 <i>Machine Learning (ML) o Aprendizaje máquina:</i> .....                      | 17 |
| 2.1.3 <i>Natural Language Processing (NLP):</i> .....                                | 17 |
| 2.1.4 <i>Big data o Macrodatos:</i> .....                                            | 17 |
| 2.1.5 <i>Artificial Neural Networks (ANN) o Redes neuronales artificiales:</i> ..... | 18 |
| 2.1.6 <i>Deep Learning (DL) o Aprendizaje profundo:</i> .....                        | 18 |
| 2.1.7 <i>Fuzzy Logic o Lógica Difusa:</i> .....                                      | 18 |
| 2.1.8 <i>Radiómica:</i> .....                                                        | 18 |
| 2.1.9 <i>Cambio Climático:</i> .....                                                 | 18 |
| 2.1.10 <i>Capa de Ozono y Radiación UV:</i> .....                                    | 18 |
| 2.1.11 <i>Nevo Displásico:</i> .....                                                 | 19 |
| 2.1.12 <i>Queratosis Actínica:</i> .....                                             | 19 |
| 2.1.13 <i>Cáncer de Piel:</i> .....                                                  | 19 |
| 2.1.14 <i>Melanoma:</i> .....                                                        | 19 |
| 2.2 FACTORES DE RIESGO DEL CÁNCER DE LA PIEL: .....                                  | 20 |
| 2.3 TIPOS DE CÁNCER DE LA PIEL: .....                                                | 22 |
| 2.4 VISIÓN ARTIFICIAL:.....                                                          | 24 |
| 2.5 DESARROLLO Y EXPANSIÓN DE LA INTELIGENCIA ARTIFICIAL: .....                      | 25 |
| 2.5.1 <i>Inteligencia Artificial aplicada a la salud:</i> .....                      | 26 |
| 2.6 CONCEPTOS DE PROCESAMIENTO Y CLASIFICACIÓN DE IMÁGENES: .....                    | 27 |
| 2.6.1 <i>Modelo matemático de una imagen:</i> .....                                  | 28 |
| 2.6.2 <i>Espacios de color:</i> .....                                                | 28 |
| 2.6.3 <i>Convolución y filtrado de imágenes:</i> .....                               | 28 |
| 2.6.4 <i>Aprendizaje y clasificación:</i> .....                                      | 29 |

## CAPÍTULO 3 ESTADO DEL ARTE ..... 32

|                                                    |    |
|----------------------------------------------------|----|
| 3.1 IMPORTANCIA DE UN BUEN DATASET .....           | 32 |
| 3.2 CIENCIA DE DATOS Y EL CUIDADO DE LA SALUD..... | 32 |

|                                                                                                                |    |
|----------------------------------------------------------------------------------------------------------------|----|
| 3.3 CLASIFICACIÓN DE LUNARES CON VISIÓN ARTIFICIAL .....                                                       | 33 |
| 3.4 LA INTELIGENCIA ARTIFICIAL EN DERMATOLOGÍA .....                                                           | 34 |
| 3.5 PRONÓSTICO EN MELANOMA .....                                                                               | 34 |
| 3.6 CÁNCER DE PIEL .....                                                                                       | 35 |
| 3.7 ÉTICA SOBRE EL USO DE LA IA EN LA DERMATOLOGÍA .....                                                       | 35 |
| 3.8 ¿LA IA AMENAZA AL TRABAJO MÉDICO DE LOS DERMATÓLOGOS?.....                                                 | 36 |
| 3.9 LA LIMPIEZA DE IMÁGENES POR MEDIO DE ALGORITMOS EN LA DERMATOLOGÍA .....                                   | 37 |
| 3.10 FOTOFINDER AI .....                                                                                       | 37 |
| 3.11 CLASIFICADOR DE IMÁGENES CON PYTHON Y TENSORFLOW.....                                                     | 38 |
| 3.12 CLASIFICACIÓN AVANZADA DE IMÁGENES CON IA / ML (CNN) .....                                                | 39 |
| 3.13 CÁNCER DE PIEL EN EL MUNDO .....                                                                          | 41 |
| 3.14 SITUACIÓN DE CÁNCER EN EL PERÚ .....                                                                      | 43 |
| 3.15 CASO REGIONAL SOBRE CÁNCER DE PIEL EN PERÚ: EXPOSICIÓN SOLAR EN COMERCIANTES AMBULANTES EN CHICLAYO ..... | 45 |

## **CAPÍTULO 4 METODOLOGÍA E IMPLEMENTACIÓN . 47**

|                                                                     |     |
|---------------------------------------------------------------------|-----|
| 4.1 HERRAMIENTAS A USAR .....                                       | 47  |
| 4.2 DATASET TEST EN KAGGLE .....                                    | 48  |
| 4.3 CLÚSTERES EN RAPIDMINER .....                                   | 54  |
| 4.4 DATASET TRAIN EN TABLEAU .....                                  | 59  |
| 4.5 CLASIFICACIÓN Y CLÚSTERES EN KNIME.....                         | 63  |
| 4.6 LIBRERÍAS USADAS DE KAGGLE .....                                | 67  |
| 4.7 ANÁLISIS DE LOS ESPACIOS VACÍOS Y VALORES PERDIDOS .....        | 71  |
| 4.8 ANÁLISIS DE LA VARIABLE OBJETIVO .....                          | 73  |
| 4.8 PREPROCESANDO LOS DATASETS.....                                 | 75  |
| 4.8 FORMATOS DE LAS IMÁGENES A TRATAR.....                          | 77  |
| 4.9 TÉCNICAS DE COLOR.....                                          | 83  |
| 4.10 REMOCIÓN DE VELLOSIDAD EN LA PIEL .....                        | 91  |
| 4.11 USAR EL PYTORCH EN EL DATASET .....                            | 93  |
| 4.12 REDES NEURONALES MEDIANTE RESNET50NETWORK .....                | 97  |
| 4.13 REDES NEURONALES MEDIANTE EFFICIENTNETWORK.....                | 105 |
| 4.13 ELECCIÓN DEL MÉTODO A UTILIZAR CON UNA MUESTRA CONTROLADA..... | 114 |
| 4.13 PRUEBA DE LA IA CON EL DATASET TEST .....                      | 119 |
| 4.13 CALCULAMOS EL ROC Y LA MATRIZ DE CONFUSIÓN .....               | 124 |

## **CONCLUSIONES Y RECOMENDACIONES ..... 126**

## **BIBLIOGRAFÍA..... 128**

## ÍNDICE DE TABLAS

|                                                                                                                              |    |
|------------------------------------------------------------------------------------------------------------------------------|----|
| <b>Tabla 1</b> Factores Asociados a la Queratosis Actínica .....                                                             | 42 |
| <b>Tabla 2</b> Conocimientos de los comerciantes ambulantes de un mercado en Lambayeque, Chiclayo -<br>setiembre, 2017 ..... | 46 |
| <b>Tabla 3</b> Detallado de las variables de la data train.csv .....                                                         | 53 |
| <b>Tabla 4</b> Tabla de centroide de k-means .....                                                                           | 56 |
| <b>Tabla 5</b> Correlaciones de los atributos .....                                                                          | 58 |

## ÍNDICE DE IMÁGENES

|                                                                                |    |
|--------------------------------------------------------------------------------|----|
| <b>Figura 1</b> Artículos de MeSH publicados entre 1988 a 2018 .....           | 10 |
| <b>Figura 2</b> Queratosis Actínica.....                                       | 22 |
| <b>Figura 3</b> Nevo Displásico.....                                           | 23 |
| <b>Figura 4</b> Aprendizaje no supervisado.....                                | 30 |
| <b>Figura 5</b> Red Neuronal artificial multicapa.....                         | 31 |
| <b>Figura 6</b> Estadística de cáncer por tipo .....                           | 43 |
| <b>Figura 7</b> Estadística de cáncer según el género.....                     | 44 |
| <b>Figura 8</b> Estadística de cáncer según la edad .....                      | 44 |
| <b>Figura 9</b> Detallado de las variables de la data train.csv .....          | 50 |
| <b>Figura 10</b> Imágenes de la data test.csv .....                            | 51 |
| <b>Figura 11</b> Imágenes de la data train.csv .....                           | 54 |
| <b>Figura 12</b> Resumen de k-means según la edad aproximada.....              | 55 |
| <b>Figura 13</b> Mapa de calor de k-means según la edad aproximada.....        | 55 |
| <b>Figura 14</b> Gráfico de centroide de k-means según la edad aproximada..... | 56 |
| <b>Figura 15</b> Estadísticas según la edad aproximada.....                    | 57 |
| <b>Figura 16</b> Estadísticas según el lugar del cuerpo afectado .....         | 57 |
| <b>Figura 17</b> Estadísticas según el sexo .....                              | 58 |
| <b>Figura 18</b> Partes del cuerpo afectadas .....                             | 59 |
| <b>Figura 19</b> Benigno o maligno.....                                        | 60 |
| <b>Figura 20</b> Diagnostico y edad .....                                      | 61 |
| <b>Figura 21</b> Edad y sexo.....                                              | 62 |
| <b>Figura 22</b> Esquema general en knime .....                                | 63 |
| <b>Figura 23</b> Matriz de confusión de Naive Bayes .....                      | 64 |
| <b>Figura 24</b> Curva de ROC de Naive Bayes.....                              | 64 |
| <b>Figura 25</b> Matriz de confusión de Árbol de decisión .....                | 65 |
| <b>Figura 26</b> Curva de ROC de Árbol de decisión .....                       | 65 |
| <b>Figura 27</b> Matriz de confusión de KNN.....                               | 66 |
| <b>Figura 28</b> Curva de ROC de KNN .....                                     | 66 |
| <b>Figura 29</b> Matriz de confusión de SVM Learner .....                      | 67 |
| <b>Figura 30</b> Curva de ROC de SVM Learner.....                              | 67 |
| <b>Figura 31</b> Mapa de valores perdidos .....                                | 72 |

|                                                                                     |     |
|-------------------------------------------------------------------------------------|-----|
| <b>Figura 32</b> Frecuencia de la variable objetivo y su distribución por edad..... | 75  |
| <b>Figura 33</b> Tamaño de 100 imágenes .....                                       | 80  |
| <b>Figura 34</b> Muestras benignas .....                                            | 82  |
| <b>Figura 35</b> Muestras malignas .....                                            | 83  |
| <b>Figura 36</b> Filtro blanco y negro.....                                         | 84  |
| <b>Figura 37</b> Filtro de desenfoque gaussiano.....                                | 86  |
| <b>Figura 38</b> Vista original de imágenes.....                                    | 87  |
| <b>Figura 39</b> Imágenes transformadas con aumento .....                           | 90  |
| <b>Figura 40</b> Muestras antes de la remoción de vello.....                        | 93  |
| <b>Figura 41</b> Muestras después de la remoción de vello .....                     | 93  |
| <b>Figura 42</b> Resultados de la salida con ResNet50Network .....                  | 101 |
| <b>Figura 43</b> Resultados de la salida con EfficientNetwork .....                 | 110 |
| <b>Figura 44</b> Evaluación de imágenes con IA .....                                | 123 |
| <b>Figura 45</b> Matriz de confusión final.....                                     | 125 |

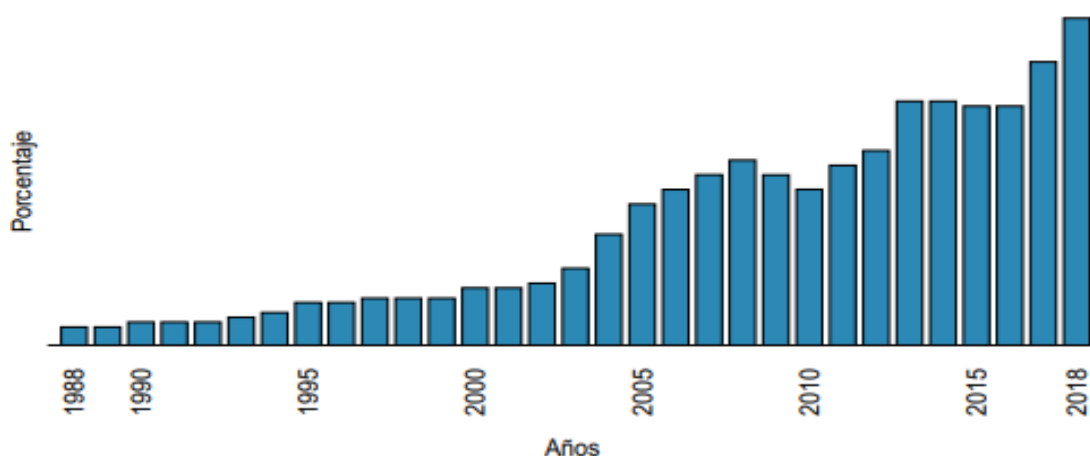
# Capítulo 1 PLANTEAMIENTO METODOLÓGICO

## 1.1 Antecedentes del problema

El uso de la inteligencia artificial ha marcado un hito en nuestras vidas y en la manera en que entendemos diversas áreas del conocimiento, al punto de poder considerarse una cuarta revolución industrial en la historia de la humanidad. Es importante mencionar que el área de las enfermedades dermatológicas es bastante vasta; intentar abarcar todas las posibles variantes sería ilusorio. Por lo tanto, es prudente delimitar el alcance de esta investigación para que su aplicación sea más fructífera y satisfactoria.

Como ilustra (Rodríguez, 2019) mediante la siguiente imagen, el avance de la inteligencia artificial se refleja en los artículos publicados bajo el término MeSH (Medical Subject Headings).

**Figura 1** Artículos de MeSH publicados entre 1988 a 2018



Fuente: Medigraphic MeSH Artificial intelligence

Es relevante resaltar que los especialistas en estas áreas suelen tener un margen de error humano. Sin embargo, la inteligencia artificial también puede presentar un margen de error, aunque este puede ser reducido con un mayor entrenamiento a través del Machine

Learning. En el ejemplo presentado en el artículo, se demuestra que, al entrenar a la inteligencia artificial con al menos 100,000 imágenes, su eficacia y eficiencia superan a las de los especialistas, lo que permite ayudar de manera más efectiva a los pacientes que requieren este tratamiento.

Al evaluar los aspectos más básicos relacionados con el cáncer de piel, fue necesario recurrir a una clínica con años de experiencia en el tema (Clinic, 2020). Conocer la sintomatología general de la enfermedad y los signos de alerta fue el primer paso en este proceso. Entre los signos destacados se encuentran bultos, lesiones, úlceras y melanomas, que son causados por células escamosas, basales y melanocitos. Factores como la herencia, la radiación solar y el tipo de piel influyen en la tendencia a desarrollar cáncer de piel; sin embargo, es importante destacar que la radiación es casi el eje central de todos estos factores. También se debe hacer hincapié en los métodos de prevención, los cuales están estrechamente relacionados con la detección temprana mencionada en el título de la tesis.

El estudio de detección se basa en imágenes que servirán como soporte para la inteligencia artificial, por lo que el conocimiento proporcionado por la (Clinic, 2020) resulta fundamental. En este caso, será necesario utilizar el reconocimiento de imágenes para distinguir entre diferentes condiciones, dado que existen múltiples enfermedades de la piel que no pueden ser abordadas en su totalidad, como se mencionará en los alcances. Asimismo, se deben considerar los diferentes tipos de piel y sus tonalidades, ya que estas son variables cruciales para determinar si un paciente padece o no cáncer de piel, así como para proporcionar un diagnóstico adecuado.

En cuanto al aspecto económico y práctico del tema, (Sebastian, 2020) busca mostrar la rentabilidad diagnóstica de los esquemas de seguimiento aplicados a los pacientes con melanoma en diferentes niveles de riesgo, tomando como base el American Joint Committee on Cancer (AJCC). Conocer el costo y la eficiencia de este proceso, así como su precisión en la evaluación de la expresión genética y sus variantes, es de suma importancia.

El avance tecnológico es innegable, y con él, el área de ciencias de la salud se está transformando; el uso de la ciencia de datos es, sin duda, uno de los principales responsables de este cambio. Ya sea mediante sistemas expertos, inteligencia artificial o

redes neuronales, las decisiones clínicas se han visto potenciadas, como afirman (Muñoz Rodríguez, Moreno Espinosa, & Torres Garzón, 2020).

El uso de modelos matemáticos y algoritmos de visión artificial resulta esencial para discriminar características relevantes en imágenes de lesiones cutáneas, facilitando el diagnóstico temprano. En particular, el procesamiento de imágenes emplea la segmentación y otros métodos analíticos para resaltar estructuras y bordes en lunares y manchas, apoyándose en técnicas que mejoran la detección de cambios en el color, asimetría y bordes irregulares. Esto es útil para distinguir melanomas de otras alteraciones cutáneas (Márquez Díaz, 2020, págs. 751-766).

Por último, el tema de algoritmos, Machine Learning e inteligencia artificial es abordado de manera más completa por Benavent Núñez et al. (2020), quienes concluyen que el avance en la inteligencia artificial aplicada a la dermatología tiene un potencial significativo que puede ser explotado en los próximos años. Este campo no solo abarca una esfera específica, sino que trasciende a otras áreas y cuenta con un respaldo sólido en demanda. Las investigaciones en este ámbito seguirán avanzando, y su desarrollo representa un aporte significativo para la sociedad futura.

## **1.2 Definición o formulación del problema**

**Interrogativo:** ¿Cómo influye la creación de una IA en la detección temprana del cáncer de la piel frente a la escasez de profesionales de dermatología?

**Descriptivo:** La escasez de profesionales en dermatología propicia la creación de una inteligencia artificial para la detección temprana del cáncer de piel.

## **1.3 Importancia del problema**

Respecto al tipo de investigación, podemos catalogarla como aplicada. Esto se debe a que nos basamos en conocimientos puros o básicos proporcionados por dermatólogos, así

como en los distintos casos de pacientes, analizados a través de imágenes, con el fin de aplicar todo esto a la detección temprana del cáncer de piel en pacientes con diversos síntomas detectables por la inteligencia artificial.

Como se verá a lo largo de la investigación, el problema del cáncer de piel ha ido en aumento con el paso de los años, desarrollando nuevas variantes que son nocivas a su manera. A nivel social, podemos observar cómo este problema ha escalado posiciones hasta convertirse en un tema cada vez más recurrente (Rodríguez, 2019). Por ejemplo, puede desfigurar la piel, los huesos y los músculos de una persona en el caso de la queratosis actínica, o incluso causar la muerte a través de un melanoma originado en un nevo displásico.

Además, debemos considerar el avance científico en campos de la informática, como se mencionó en los antecedentes, que ayudan a contrarrestar los problemas que aquejan a la humanidad. En esta investigación, estos avances se relacionan directamente con el área de la dermatología. Su importancia radica principalmente en la preservación de la salud de las personas a futuro, al disminuir las bajas causadas por el cáncer de piel. Al igual que en muchos otros casos, si se detecta a tiempo, se puede salvar una vida o evitar una operación complicada.

En conclusión, el estudio de las enfermedades de la piel, y en particular del cáncer, representa un problema persistente en nuestra sociedad, que se espera que aumente a medida que problemas como el calentamiento global se agraven. No podemos hacer la vista gorda ante este gigante que, en los últimos tiempos, ha cobrado protagonismo, especialmente por la emisión de radiación solar y la intervención humana. Además, es importante señalar que los profesionales de esta área no pueden cubrir la infinidad de casos que se presentan, por lo que es viable proporcionarles una herramienta que les ayude a satisfacer la creciente demanda en este campo. Por lo tanto, podemos enfatizar que esta

investigación tiene como objetivo apoyar en la salvación de muchas víctimas de esta devastadora enfermedad.

## **1.4 Objetivos**

### **1.4.1 Objetivo General**

- Desarrollar un software aplicando inteligencia artificial en la dermatología para la detección temprana del cáncer de piel

### **1.4.2 Objetivos Específicos**

- Investigar la sintomatología, prevención y causas del cáncer de piel, específicamente del Nevo Displásico y la Queratosis Actínica.
- Reducir la tasa de detección de falsos positivos por medio del preprocesamiento de imágenes.
- Implementar un modelo de Machine Learning que permita la clasificación eficiente de imágenes dermatológicas.
- Probar de la eficiencia del software para diagnosticar de forma precisa el cáncer de piel benigno.

## **1.5 Alcances y Limitaciones**

### **Alcances:**

Se desarrollará una inteligencia artificial para la detección temprana del cáncer de piel debido a varios factores relevantes que deben tenerse en cuenta en este tipo de investigación:

- Variedad de enfermedades dermatológicas: Existe una gran cantidad de enfermedades de la piel, muchas de las cuales presentan apariencias similares o sintomatología parecida, lo que dificulta su diagnóstico correcto.

- Diversidad en características físicas: Los tonos de piel, el color de ojos y el cabello, contemplados por la escala Fitzpatrick, son elementos importantes en el análisis de imágenes, ya que pueden influir en la precisión del diagnóstico.
- Enfoque específico: Este proyecto se centrará únicamente en el nevo displásico y la queratosis actínica, dos variantes del cáncer de piel que se desean abarcar.

### **Limitaciones:**

Dentro de las limitaciones que pueden obstaculizar el correcto avance del proyecto o restringir la cobertura de los aspectos o áreas del conocimiento, encontramos:

- Requerimientos temporales: El tiempo necesario para desarrollar la inteligencia artificial es considerable, ya que se necesita una gran cantidad de imágenes para entrenar el modelo, incluso para una sola enfermedad.
- Complejidad del campo médico: Abordar un tema médico desde la especialidad de informática presenta desafíos significativos, dado que se requiere un entendimiento profundo de ambas disciplinas.
- Dificultades de acceso a casos clínicos: Puede ser complicado acceder a ciertos casos de pacientes que padecen enfermedades de la piel, los cuales serían útiles para la investigación y el entrenamiento del modelo.

## **1.6 Justificación**

La presente investigación y creación de una inteligencia artificial se justifica en diversas áreas: social, técnica y económica.

### **1.6.1 Justificación Social**

La justificación social se fundamenta en la creciente demanda de dermatólogos, atribuida a la exposición a las radiaciones UV de tipo A y B producidas por el sol. La presencia de quemaduras solares se asocia con un aumento en la incidencia del cáncer de piel, pudiendo duplicar o cuatriplicar el riesgo en

ciertos casos. Detectar una enfermedad en su estado temprano es crucial para evitar su progresión y minimizar los daños, ya que, por ejemplo, la queratosis actínica puede desfigurar huesos si se extiende mediante tumores ulcerados. Por otro lado, el nevo displásico es más mortal y se propaga rápidamente, pudiendo resultar fatal a través de melanomas, cuya evaluación se realiza utilizando la regla ABCDE. En última instancia, esta herramienta contribuirá a preservar la vida de las personas, garantizando su bienestar.

### **1.6.2 Justificación Técnica**

Desde el punto de vista técnico, la investigación se apoya en el desarrollo de un modelo matemático que permita filtrar casos de cáncer de piel mediante imágenes. Esto implicará la aplicación de conocimientos en inteligencia artificial y machine learning. La investigación está justificada por la contribución que representa al campo de la ingeniería, proporcionando una herramienta que satisfaga la creciente necesidad de profesionales en dermatología. Al ser un apoyo para los dermatólogos, esta inteligencia artificial no irá en contra de la ética profesional, sino que potenciará su trabajo al permitirles atender a un mayor número de pacientes.

### **1.6.3 Justificación Económica**

Desde la perspectiva económica, se deben considerar los ahorros en costos relacionados con la atención dermatológica. Utilizando esta inteligencia artificial, los dermatólogos podrán atender a más pacientes en el mismo período de tiempo, lo que se traduce en menos horas pagadas y más pacientes atendidos. Esto, en términos económicos, significa un aumento en los ingresos frente a menores costos operativos para hospitales y clínicas. Asimismo, contar con un servicio más eficaz y eficiente puede mejorar la imagen de las instituciones de salud, atrayendo a un público mayor. En consecuencia, se genera un escenario donde todos ganan: se monetiza más y aumenta el número de personas curadas a tiempo del cáncer de piel.

## Capítulo 2 MARCO TEÓRICO

### 2.1 Definiciones y conceptos

A lo largo de la tesis se irán abarcando diferentes palabras de forma regular, por lo que esclareceremos las terminologías de cada uno de ellos:

#### 2.1.1 Artificial Intelligence (AI) o IA (inteligencia artificial):

La inteligencia artificial (IA) es "el estudio de los agentes inteligentes, sistemas capaces de tomar decisiones racionales basadas en información percibida y sus objetivos" (Poole, 2020, pág. 15).

#### 2.1.2 Machine Learning (ML) o Aprendizaje máquina:

El aprendizaje de máquina se refiere a "un tipo de inteligencia artificial en el que las computadoras aprenden de los datos sin ser explícitamente programadas para realizar una tarea específica" (Kelleher, 2019, pág. 5).

#### 2.1.3 Natural Language Processing (NLP):

El procesamiento de lenguaje natural (PLN) es "la capacidad de una máquina para comprender y manipular el lenguaje humano, basada en técnicas de aprendizaje automático" (Eisenstein, 2019, pág. 23).

#### 2.1.4 Big data o Macrodatos:

Los macrodatos son "grandes volúmenes de datos complejos que requieren técnicas avanzadas de procesamiento y análisis para ser gestionados y extraer valor" (Marr, 2020, pág. 12).

### **2.1.5 Artificial Neural Networks (ANN) o Redes neuronales artificiales:**

Las redes neuronales artificiales son "modelos computacionales inspirados en la estructura del cerebro humano, consistentes en nodos interconectados que procesan información en capas" (Aggarwal, 2021, pág. 30).

### **2.1.6 Deep Learning (DL) o Aprendizaje profundo:**

El aprendizaje profundo es "una subcategoría del aprendizaje de máquina que utiliza arquitecturas de redes neuronales profundas para realizar tareas complejas de análisis de datos" (Li, 2021, pág. 45).

### **2.1.7 Fuzzy Logic o Lógica Difusa:**

La lógica difusa es "un enfoque matemático que permite manejar la incertidumbre y el grado de verdad en los datos, en lugar de la lógica booleana estricta" (Zimmermann, 2021, pág. 60).

### **2.1.8 Radiómica:**

La radiómica es "un campo de investigación que extrae grandes cantidades de características cuantitativas de imágenes médicas para la personalización de tratamientos médicos" (Lambin, 2020, pág. 793).

### **2.1.9 Cambio Climático:**

El cambio climático implica "alteraciones significativas en los patrones climáticos de la Tierra, causadas principalmente por actividades humanas, como la emisión de gases de efecto invernadero" (Field, 2022, pág. 457). Estas alteraciones afectan los ecosistemas y la salud pública y generan condiciones climáticas extremas que incrementan los riesgos para la humanidad.

### **2.1.10 Capa de Ozono y Radiación UV:**

La capa de ozono actúa como un "escudo natural que bloquea gran parte de la radiación ultravioleta (UV) proveniente del sol. Su disminución, causada por

sustancias como los CFCs, ha incrementado los niveles de radiación UV en la superficie terrestre, lo cual aumenta el riesgo de cáncer de piel y otros problemas de salud" (McKenzie, 2023, pág. 125).

#### **2.1.11 Nevo Displásico:**

Un nevo displásico es "un tipo de lunar con cambios celulares anormales que puede parecerse al melanoma. Aunque en su mayoría son benignos, estos nevos se consideran posibles precursores del melanoma, por lo que requieren vigilancia médica" (Weinstock, 2022, pág. 237).

#### **2.1.12 Queratosis Actínica:**

La queratosis actínica es "una lesión cutánea premaligna causada por exposición prolongada al sol, que se presenta como áreas ásperas y escamosas en la piel. Aunque generalmente son benignas, pueden evolucionar hacia carcinoma de células escamosas" (Marks, 2021, pág. 296).

#### **2.1.13 Cáncer de Piel:**

El cáncer de piel es "una enfermedad causada principalmente por exposición a radiación UV, que afecta tanto la epidermis como las células subyacentes. Incluye tipos comunes como el carcinoma de células basales, el carcinoma de células escamosas y el melanoma" (Green, 2023, pág. 1048).

#### **2.1.14 Melanoma:**

El melanoma es "el tipo de cáncer de piel más peligroso, que se origina en los melanocitos, células que producen el pigmento de la piel. La exposición excesiva a rayos UV es un factor de riesgo importante, y su prevención incluye protección solar y monitoreo regular" (Gandini, 2023, pág. 2502).

## **2.2 Factores de riesgo del cáncer de la piel:**

En los últimos años, la incidencia de cáncer de piel ha aumentado en todos los países del mundo, siendo Estados Unidos el país con la mayor frecuencia y mortalidad por esta enfermedad. El cáncer de piel puede definirse como el crecimiento excesivo, desorganizado e invasivo de las células propias de la piel, principalmente como consecuencia de la exposición solar (Narang, 2023). Este fenómeno se debe a mutaciones en el ADN que afectan a las células escamosas, células basales y melanocitos, tal como lo menciona (Clinic, 2020).

Los rayos ultravioletas (UV) han aumentado en intensidad en los últimos años debido al calentamiento global y a la reducción de la capa de ozono. Sin esta capa protectora, los rayos UV del tipo A y B impactan con mayor frecuencia la piel humana, lo que provoca mutaciones en el ADN y, como resultado, la génesis del cáncer de piel. Estas mutaciones dificultan que el cuerpo reconozca el ADN dañado, lo que favorece el desarrollo del cáncer.

Mayo Clinic (2020) proporciona información detallada sobre los factores de riesgo del cáncer de piel, entre los cuales se destacan los siguientes:

- Piel clara: En este caso podemos valernos de la escala fitzpatrick para determinar qué tan propenso es uno de padecer cáncer de piel según factores como el color de piel, cabello y ojos. Sin embargo, factores como la melanina son determinantes en lo que respecta a la protección frente a la radiación UV. Concluyendo así que el tono de piel claro es un factor de riesgo a tener en consideración.
- Antecedentes de quemaduras por el sol: Independientemente de la edad en la cual se produzcan pueden llegar no solo a duplicar la incidencia de cáncer de la piel en caso de darse en una ocasión, sino también cuatriplicarse en caso de ser cinco quemaduras según Narang (2023).
- Exposición excesiva al sol: Ya habiendo mencionado la radiación en aumento en los últimos años es prudente tener en cuenta que la piel necesita de vitamina D proveniente del sol. Aunque podemos decir que todo en exceso es dañino, sobre todo si esta exposición se hace en horas pico del día donde el sol es más

peligroso. Se puede menguar con el uso de protección solar y una adecuada vestimenta, pero lo recomendable es no arriesgarse en primer lugar.

- Climas soleados o en grades altitudes: Volvemos al mismo punto de la radiación donde zonas como ecuador tienen mayor incidencia de calor al estar más alejado de los polos y por la misma distribución de la tierra. Además, que sin contar el calor del lugar es más critica la altura porque se recibe una cantidad mortífera de radiación que en algunos casos puede ser perjudicial.
- Lunares: Aquí se entra a tallar en uno de los ejes principales de la investigación, porque estos lunares van relacionados con el Nevo Displásico y su presencia ciertamente puede ser una señal de alerta. Ya que dependiendo de su desarrollo puede causar la misma muerte.
- Lesiones cutáneas precancerosas: Aquí tocamos otro aspecto importante de la investigación referente a la queratosis actínica, la cual no es tan mortal como la pasada. Pero, el detectar estas lesiones puede ayudar a que no perforen el hueso y dejen marcas que solo pueden ser corregidas con cirugías.
- Antecedentes familiares de cáncer de piel: El riesgo genético siempre debe tenerse en cuenta a la hora de saber si es más o menos probable que una persona desarrolle una enfermedad, más tratándose del cáncer
- Antecedentes personales de cáncer de piel: En este apartado se puede ver las recaídas de una persona que ya padeció la enfermedad por algunos de los factores ya mencionados o algunos que seguiremos mencionando.
- Sistema inmunitario debilitado: No solamente aplica para el cáncer, puesto que el tener las defensas bajas propicia todo tipo de enfermedades. Cosas como la alimentación y medicamentos también se ven inmiscuidas en este apartado. Enfermedades como el VIH/SIDA, así como las que necesitan fármacos inmunosupresores post operatorios.
- Exposición a la radiación: Este punto se tocó casi de ultimo por el principal causante e involucrado con puntos pasados debido a la mutación de ADN que provoca últimamente en las personas. Solo hay que recalcar que los rayos del tipo C son benignos en contraste con los tipos A y B que se ven actualmente.

- La exposición a ciertas sustancias: Podemos mencionar al arsénico como uno de los tantos detonantes que ocasionan del cáncer de piel, aunque de la misma manera es bueno ver las consecuencias de las sustancias a las que exponemos la piel.

### **2.3 Tipos de cáncer de la piel:**

Dado que abarcar todos los tipos de cáncer de piel seria extender demasiado la investigación, pero a fin de suplir el mayor número de casos se eligió los dos principales tipos conocidos pero que pueden diferenciar fácilmente el uno del otro, los cuales son:

- Queratosis Actínica: Esta es la menos mortal de las dos, pero no por ello la podemos menospreciar. Como ya se mencionó anteriormente esta surge de lesiones cutáneas precancerosas las cuales forman unos pseudo tumores que de tratarse puede evitarse que avance al punto de desfigurar el hueso, el musculo y tejido de la persona. Esta se presenta en forma de carcinomas de células basales que son similares a los tumores y carcinomas escamo celulares que se son similares a las ulceras y se llenan de sangre como en la siguiente imagen.

***Figura 2 Queratosis Actínica***



Fuente: Sinapsis EMP

- **Nevo Displásico:** Esta es más mortal que la anterior porque se forma por medio de melanomas que pueden expandirse de manera acelerada propiciando la muerte del paciente. El melanoma en si es un tumor agresivo que se caracteriza por empezar como un lunar de forma extraña, el cual se distingue de los lunares normales por medio de la regla ABCDE:
  - **A (Asimetría):** Se refiere a que podemos trazar una línea cualquiera que pase por el lunar donde los dos espacios formados son diferentes o asimétricos.
  - **B (Borde):** El lunar en cuestión no cuenta con bordes notables, debido a que se mezcla con la piel teniendo una forma extraña.
  - **C (Color):** Vemos variaciones de tonalidad en el lunar, siendo que en algunas zonas es más oscura que en otras.
  - **D (Diámetro):** Es sabido que los lunares son pequeños y esta característica los mide en un aproximado de 6mm donde al superar este tamaño entra a ser considerado un posible melanoma.
  - **E (Evolución):** Entendemos por evolución al cambio del lunar a lo largo del tiempo pasando de tener unas características normales a unas que se asemejan a un melanoma.

Por último, mostrar una imagen donde se ve la regla ABCDE en un melanoma:

***Figura 3 Nevo Displásico***



Fuente: Mayo Clinic Melanoma

## **2.4 Visión artificial:**

Muñoz Rodríguez, Moreno Espinosa y Torres Garzón (2020) destacan que la percepción visual ha sido un campo de profunda investigación en la ciencia cognitiva, ya que permite explorar cómo la mente humana convierte la información sensorial en percepción consciente. Según estos autores, la percepción visual es fundamental porque facilita la integración de otras percepciones sensoriales en representaciones visuales complejas. Este fenómeno respalda la idea de que las representaciones mentales en imágenes enriquecen el pensamiento y no pueden reducirse únicamente a expresiones verbales.

Los sistemas de visión artificial actuales pueden generar descripciones detalladas del entorno que los rodea mediante el procesamiento y reconstrucción de imágenes, lo cual se relaciona estrechamente con la percepción computacional. Esto significa que la máquina no solo es capaz de "ver" su entorno, sino también de interactuar con él de manera coherente. En el campo de la robótica, la percepción activa es un componente esencial, ya que implica un sistema de visión que funcione en tiempo real, con suficiente capacidad de procesamiento para ser eficaz en aplicaciones que requieren un análisis continuo y en línea. Esta percepción activa también se puede aplicar en otras áreas, como la animación por computadora, donde se necesita un modelo que integre movimiento y coordinación visual.

A continuación, se presentan ejemplos de aplicaciones de la visión artificial en diferentes áreas:

- Medicina: análisis de semen, control de calidad en procesos de selección y clasificación, y análisis morfológico de células.
- Industria: medición dimensional de piezas, detección de defectos en circuitos integrados, supervisión de manipulación de materiales.
- Seguridad y Defensa: reconocimiento facial, vigilancia para detectar intrusos, guiado remoto de misiles, y control del tráfico aéreo y marítimo.
- Agricultura y Ecología: pronóstico de cultivos, estudio de plagas, y análisis del equilibrio ecológico.
- Meteorología y Geología: cartografía climática, pronóstico del tiempo, y exploración de depósitos minerales.

- Demografía y Planificación Urbana: inventario y planificación de ciudades, monitoreo del crecimiento poblacional.
- Robótica: automatización de tareas peligrosas y visualización en terrenos hostiles.
- Interacción Hombre-Máquina: uso de gestos para la interacción, reconocimiento de patrones, y lectura de códigos de barras y OCR.
- Finanzas y Computación Gráfica: visualización de datos de mercado y análisis predictivo.

(Muñoz Rodríguez, Moreno Espinosa, & Torres Garzón, 2020).

## **2.5 Desarrollo y expansión de la Inteligencia Artificial:**

En los últimos diez años, los conceptos de inteligencia artificial (IA) y aprendizaje automático han experimentado una expansión significativa, en gran parte gracias a la revolución de los macrodatos. Este avance ha impulsado un crecimiento en la cantidad de publicaciones científicas y ha extendido el impacto de estas tecnologías en diversos sectores, especialmente en el ámbito de la salud. Sin embargo, el desarrollo de la IA se remonta a varias décadas. Es importante comprender su evolución para manejar de forma adecuada conceptos fundamentales en este campo.

La IA tiene sus orígenes en 1956, cuando John McCarthy, en la Universidad de Dartmouth, propuso esta disciplina inspirándose en el modelo computacional de Alan Turing, quien investigó cómo las máquinas podían aprender. Desde sus inicios, el objetivo de la IA ha sido imitar aspectos de la inteligencia humana. Uno de los primeros ejemplos de IA fue el programa de juego de damas desarrollado por Arthur Samuel para el modelo 701 de IBM, que permitía a la máquina mejorar su rendimiento en el juego a partir de sus experiencias previas.

Durante las décadas de 1960 y 1970, la investigación en IA se centró en áreas como la heurística, una técnica para resolver problemas complejos al clasificar opciones con base en la información disponible en un momento dado. En este periodo también comenzó el desarrollo del procesamiento del lenguaje natural (PLN), utilizando reglas estáticas. Más

tarde, en los años 80, con la aparición del aprendizaje automático y las redes neuronales, el PLN experimentó un avance notable.

Sin embargo, las expectativas iniciales de la IA no se cumplieron, lo que llevó a una reducción del interés y la financiación en el área a principios de los años 90, periodo conocido como el "invierno de la IA". Esto resultó en un estancamiento de los avances debido a la desilusión con los resultados obtenidos.

A comienzos del siglo XXI, los nuevos avances en tecnología de la computación, como el aumento en la capacidad de almacenamiento de datos y el procesamiento de alta velocidad a bajo costo, junto con la revolución de los macrodatos, llevaron a un resurgimiento del interés en la IA y en el aprendizaje automático (ML). Desde entonces, la IA ha comenzado a tener un impacto significativo en sectores como el transporte, la educación, la seguridad, el entretenimiento y la salud. Proyectos como los vehículos autónomos, la traducción en tiempo real y las recomendaciones personalizadas son solo algunos ejemplos de aplicaciones que utilizan técnicas de reconocimiento de lenguaje, visión por computadora, procesamiento del lenguaje natural y predicción a través de redes neuronales (Benavent Núñez, Colomer Mascaró, Quecedo Gutiérrez, Gol-Montserrat, & Del Llano Señarís, 2020).

### **2.5.1 Inteligencia Artificial aplicada a la salud:**

En el ámbito de la salud, la IA ha surgido como una tecnología prometedora para mejorar el sistema sanitario. Sin embargo, uno de los mayores desafíos para su implementación es la obtención y el procesamiento de datos, ya que estos son especialmente delicados en este sector. Afortunadamente, la adopción masiva de la Historia Clínica Electrónica (HCE) ha facilitado la obtención de datos útiles, y la proliferación de sensores, aplicaciones móviles y dispositivos conectados en hospitales ha contribuido a incrementar significativamente la cantidad de datos disponibles.

Una de las aplicaciones más prometedoras de la IA en la salud es su integración en los sistemas de apoyo a la toma de decisiones clínicas (CDSS). Desde mediados del siglo XX, se han desarrollado sistemas para colaborar en

tiempo real con los médicos en la toma de decisiones. Inicialmente, estos sistemas se basaban en reglas predefinidas derivadas de la experiencia de profesionales o de la literatura científica, ofreciendo apoyo en el diagnóstico y tratamiento mediante un razonamiento estructurado.

Aunque estos primeros sistemas basados en reglas lograron éxitos en algunos proyectos piloto de diagnóstico y tratamiento, su uso en la práctica clínica general fue limitado, ya que resultaban costosos de desarrollar y requerían actualizaciones constantes a medida que avanzaba el conocimiento médico.

Los recientes avances en IA y aprendizaje automático han permitido el desarrollo de algoritmos más complejos que superan las limitaciones de los sistemas basados en reglas estáticas. La HCE ha facilitado la integración de estos nuevos sistemas de apoyo a la toma de decisiones, permitiendo detectar patrones en los datos y analizar relaciones complejas entre ellos. Esto ha dado lugar al desarrollo de proyectos piloto que han demostrado mejoras en los resultados de salud y en la eficiencia diagnóstica y terapéutica. Los profesionales de la salud juegan un rol clave en estos desarrollos, ya que es esencial una comprensión profunda de las características relevantes de las enfermedades para optimizar el uso de la IA en la práctica clínica (Benavent Núñez, Colomer Mascaró, Quecedo Gutiérrez, Gol-Montserrat, & Del Llano Señarís, 2020).

## **2.6 Conceptos de procesamiento y clasificación de Imágenes:**

Para describir conceptos de procesamiento y clasificación de imágenes en relación con la detección temprana de anomalías en la piel, el trabajo de Márquez Díaz (2020) ofrece un enfoque detallado, basado en modelos matemáticos y técnicas de visión artificial para el análisis de melanomas y queratosis actínica. Este estudio abarca el procesamiento de imágenes en varias etapas, comenzando con la segmentación de objetos relevantes para aislar áreas de interés, en este caso, lunares y manchas en la piel. Dichas zonas se analizan aplicando umbrales para determinar bordes y características relevantes

### 2.6.1 Modelo matemático de una imagen:

Márquez Díaz propone un modelo donde una imagen se representa como una función en escala de grises o RGB, donde cada píxel se puede asociar con coordenadas específicas. Este modelo es clave para aislar y estudiar las variaciones en tonalidad y morfología, aspectos esenciales en la detección de irregularidades en muestras dermatológicas

### 2.6.2 Espacios de color:

Para mejorar la segmentación de imágenes en este estudio, se emplean distintas conversiones entre espacios de color, como el RGB y la escala de grises, lo cual permite un análisis más detallado de las características pigmentarias y morfológicas de las áreas estudiadas. Estas conversiones son especialmente útiles para la detección de bordes y variaciones en el brillo de las imágenes.

- Espacio de color RGB: Este espacio de color representa el color como una combinación lineal de tres componentes: R (rojo), G (verde) y B (azul). Cada uno de estos componentes o canales está codificado en 8 bits (1 byte), permitiendo valores que van de 0 a 255. En conjunto, cada píxel contiene 24 bits (3 bytes) de información.
- Espacio de color HSV: En este espacio de color, el color se describe mediante tres características: H (tono), S (saturación) y V (valor). Este enfoque facilita el análisis y la manipulación de colores según sus propiedades perceptuales, lo que resulta útil para ciertas tareas de segmentación.

### 2.6.3 Convolución y filtrado de imágenes:

- Convolución: Es una operación matemática que combina dos imágenes o funciones de un solo componente  $f(x, y)$  y  $h(x, y)$  para reproducir una función  $g(x, y) = f(x, y) * h(x, y)$ . La convulsión se define como:

$$g(u, v) = \sum_{x=0}^m \sum_{y=0}^n f(u-x, v-y) * h(x, y)$$

En el procesamiento de imágenes, se utiliza para resaltar patrones y eliminar el ruido, facilitando la identificación de contornos en áreas de interés. Para mejorar la precisión en la detección de anomalías, se aplican filtros específicos como los de Sobel, Roberts y Prewitt, que ayudan a identificar los bordes y las formas en la imagen (Pérez Ortiz de Landaluce, 2021).

#### **2.6.4 Aprendizaje y clasificación:**

El aprendizaje automático permite la creación de modelos que pueden detectar patrones y clasificar datos basados en ejemplos de entrenamiento. Este enfoque es esencial en tareas como la clasificación de imágenes (Pérez Ortiz de Landaluce, 2021). En este estudio, se emplean dos tipos principales de aprendizaje: supervisado y no supervisado.

- **Aprendizaje no supervisado:** Este tipo de aprendizaje no requiere datos etiquetados, permitiendo que el sistema identifique patrones y estructuras internas sin intervención humana. En la clasificación de imágenes, el aprendizaje no supervisado es útil para descubrir similitudes sin necesidad de categorizar previamente cada ejemplo. (Pérez Ortiz de Landaluce, 2021, Figura 4).
- **Aprendizaje supervisado:** En este enfoque, los datos de entrenamiento están etiquetados, lo que permite al modelo aprender a partir de ejemplos con resultados conocidos. Durante el entrenamiento, el sistema ajusta sus parámetros para minimizar el error de predicción, optimizando así su rendimiento en tareas de clasificación (Pérez Ortiz de Landaluce, 2021).

**Figura 4** Aprendizaje no supervisado



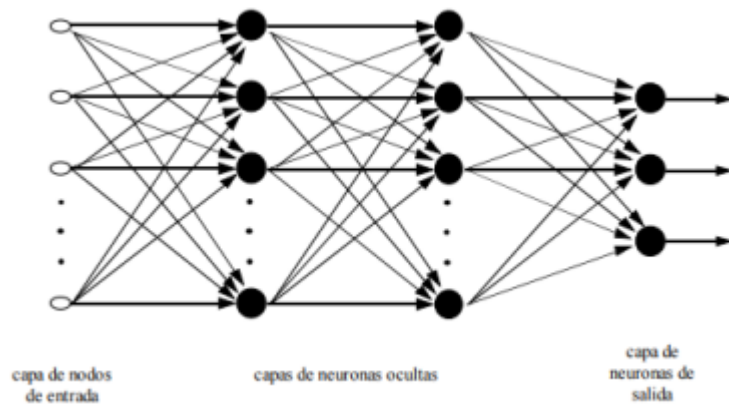
Fuente: Clasificación de imágenes mediante algoritmos de Deep Learning: Mascarillas de COVID-19.

- Categorización: Cada clasificador utilizado en este estudio evalúa un vector de características extraído de la imagen y asigna un valor de -1 o 1, indicando la categoría a la que pertenece la muestra. La capacidad de predicción de cada clasificador depende de su topología y de una función de costo que optimiza el modelo para reducir los errores en la categorización.
  - SVM (Máquina de vectores de soporte): Las SVM son modelos supervisados comúnmente usados en la clasificación de imágenes. Estas separan las clases mediante un hiperplano óptimo, maximizando la distancia entre las categorías y minimizando el error de clasificación. Cuando los datos no son linealmente separables, las SVM utilizan funciones de Kernel, proyectando los datos en un espacio de mayor dimensión en el cual se vuelven separables (Pérez Ortiz de Landaluce, 2021).

Para datos no linealmente separables, se emplea la técnica del Kernel, que proyecta los datos de un espacio de menor dimensión (por ejemplo,  $R^2$ ) a uno de mayor dimensión (por ejemplo,  $R^3$ ), lo cual permite aplicar un hiperplano de separación para mejorar la categorización.

- Red neuronal: Las redes neuronales, particularmente las convolucionales, son fundamentales para la clasificación de imágenes. Estas redes imitan la estructura del cerebro humano mediante múltiples capas de neuronas interconectadas, permitiendo el aprendizaje de características visuales en distintos niveles de abstracción, desde bordes simples hasta formas complejas (Pérez Ortiz de Landaluce, 2021).

**Figura 5** Red Neuronal artificial multicapa



Fuente: Clasificación de imágenes mediante algoritmos de Deep Learning: Mascarillas de COVID-19.

## **Capítulo 3 ESTADO DEL ARTE**

### **3.1 Importancia de un buen dataset**

En primera instancia (Benavent Núñez et al., 2020) realizan un filtro de las bases de datos de Medline y WOS obteniendo 229 resultados. Para posteriormente pasar a tener 120 con la eliminación de duplicados y acabar finalmente con 71 tras una lectura exhaustiva.

Dentro de todos estos trabajos seleccionados podemos destacar a dos que pertenecen al área médica de dermatología, más específicamente hablando sobre el tema de diagnóstico. Y de los mismos podemos destacar al trabajo realizado por (Kharazmi P, et al, 2018). Dentro de este mismo trabajo se habla sobre las estructuras vasculares en torno a la detección y localización de estas mediante el uso de imágenes. Siendo esto último de especial importancia ya que la presente investigación se basa justamente en el mismo principio de recolección de imágenes para emitir un resultado. Para la misma investigación solo se tuvo en cuenta un total de 400 imágenes, pero una cantidad de parches que roza los 10000 resultados.

Así que teniendo en cuenta la bibliografía ya tratada podemos llegar a algunas conclusiones respecto a esto. Remarcar lo importante que es el hecho de tener una mayor entrada de imágenes para una mayor precisión a la hora de realizar diagnósticos clínicos. Por lo que para la presente investigación se tendrá en cuenta esto a fin de brindar un servicio mucho más eficaz. También es lícito mencionar que no solo se realizan diagnósticos en el apartado médico sino para la predicción y el tratamiento. Basados en esto podemos ver que las fronteras de la investigación pueden ampliarse en trabajos futuros siguiendo esa guía.

### **3.2 Ciencia de datos y el cuidado de la salud**

El aporte de (Muñoz Rodríguez et al., 2020) está en introducir el término de salud móvil por parte de la OMS que consiste en aplicar el uso de la tecnología con el fin de diagnosticar las enfermedades con mayor precisión y eficacia. Enfocados en enfermedades no transmisibles pero si con chance de mortalidad.

Teniendo claro esto y la intervención cada vez mayor de los dispositivos tecnológicos y que en el apartado de decisiones clínicas se requiere de un tacto más firme y objetivo. Podemos concluir que la aplicación de la ciencia de datos y la inteligencia artificial son cada día una opción más viable en el apartado médico y que la presente investigación busca aumentar aún más el abanico de trabajos referentes a esta área. Además de ver como la misma OMS ya tiene en mayor consideración estos avances con el paso de los años. Y por lo mismo estamos en la época donde es mejor investigar y estudiar este tipo de temas.

### **3.3 Clasificación de lunares con visión artificial**

Mediante diferentes herramientas (Reales Castro, 2017) nos presenta un desarrollo que abarca una de las partes centrales de la investigación. Ya que al tratar con el melanomas, los cuales son un tipo de lunares provenientes del Nevo displásico, un tipo de cáncer de piel, nos es útil la información contenida en esta tesis.

Empecemos con el hecho del preprocesamiento que es evaluado con herramientas de graficación en 3 dimensiones. Pudiendo así hallar los desniveles producto de los vellos de las personas. Donde cabe aclarar que su funcionamiento también está basado en fórmulas matemáticas. Una vez capturada la imagen de los vellos se pasa a la segmentación para poder detectar el lunar. Habiendo realizado el proceso de manera satisfactoria pasamos a la atenuación de sombras mediante un espectro de colores.

Conseguidos los datos anteriormente mencionados podemos extraer las características del lunar. Empezando por la asimetría del mismo que se enlaza a lo ya tratado con la regla ABCDE. Basada en diferentes parámetros que se entrelazan con los canales de la oscuridad de la piel, la información del color, así como la textura y sus cambios. Luego se ve el tema de los bordes mediante canales y fórmulas matemáticas. Para después ver el color basado en la escala RGB así como diferentes nomenclaturas que convergen en diferencias estructurales a modo de características.

Luego de esto realiza la clasificación prueba y resultados obtenidos. De todo esto podemos sacar una clara guía de como enlazar el tema de la entrada de imágenes y la

posterior salida que resulta en el caso de los lunares. Quedando esto como un precedente de la línea metodológica que se debe seguir para alcanzar el éxito.

### **3.4 La inteligencia artificial en dermatología**

Este artículo fue la principal inspiración para realizar este proyecto de investigación, esto de la mano de (Rodríguez, 2019). Donde se expuso la creación de una IA que recibió una entrada de cerca de 100000 imágenes de casos de pacientes con enfermedades dermatológicas. Además, muestra por medio de una gráfica la evolución de los MeSH desde 1988 hasta 2018. Lo que nos indica que los artículos relacionados a IA no solo han aumentado, sino que van de la mano con el apartado médico. Y en lo que respecta a la dermatología también se tiene la incidencia producto de radiación.

Por lo que este artículo sirvió como una guía para forjar la idea del proyecto de investigación, así como cimentar las bases del procedimiento que debe seguir. Así también mostrar trabajos pasados en esta área y cómo empezaron en tan solo 100 imágenes para ir aumentando la cantidad de entradas. Esto último se traduce en una mayor precisión y una ayuda para los profesionales de dermatología con una alta carga laboral.

### **3.5 Pronóstico en melanoma**

Lo más rescatable de esta tesis doctoral es el hecho de involucrar de buena manera el apartado económico que será necesario a posteriori. Pues una vez se realice el proyecto y se consiga una inteligencia artificial capaz de dar diagnósticos clínicos por medio de imágenes tras un previo entrenamiento. Será necesario poder monetizar de alguna manera, por lo que con la experiencia de (Sebastian, 2020) podemos tener una idea clara de cómo se realiza esto enfocándose al apartado de los melanomas.

Pero que su funcionamiento tampoco está muy alejado a lo que sería la queratosis actínica la cual también se evaluará producto del siguiente informe. Entrando en materia la evaluación económica es un tema crucial pues es la manera en cómo muchos especialistas terminan monetizando por medio de sus saberes. Y es menester conocer experiencias pasadas a fin de saber que margen de ganancia se obtiene, cual es el precio que se debe

cobrar por el uso de esta herramienta o si es mejor ser patrocinado por alguna empresa clínica que le interese la idea y el producto.

### **3.6 Cáncer de piel**

Lo evaluado dentro de la investigación con fines educativos por (Narang, 2023) va más ligado al ámbito médico. Conociendo de primera mano la gravedad y recurrencia de la enfermedad a tratar. Pero sobre todo definir conceptos clave, delimitar el alcance al tener tantas variaciones. Conocer la sintomatología y causas de la enfermedad, así como el que la propicia. Para poder así evaluar de una mejor manera los casos que se trataran.

Dentro de las herramientas que se proporcionan están la escala Fitzpatrick y la regla ABCDE. Siendo ambos pilares en lo que tratar estas variantes de la enfermedad se trata. Además de tener en cuenta las recomendaciones respectivas que nos sirven para saber redactar un buen diagnóstico clínico que sepa satisfacer las necesidades del cliente dados los resultados obtenidos.

Ya que la idea es reunir conocimientos tanto médicos como tecnológicos a fin de hacerlos converger en uno solo y que esto nos ayude a emular lo realizado por los dermatólogos. Sirviendo por lo tanto estos conocimientos médicos para muchas instancias del trabajo que se realizarán.

### **3.7 Ética sobre el uso de la IA en la dermatología**

Dentro de las a destacar en esta investigación se encuentra la mención del DDSS (Diagnostic Decision Support Systems) como uno de los métodos que le deparan a la dermatología tras la intervención actual de la IA, según afirma (Sánchez-Saldaña, 2019). Siendo este punto importante porque es a donde terminan apuntando la investigación, a eso le sumamos los estudios alrededor del melanoma que van directamente ligados al nevo displásico.

También es menester mencionar como la intervención de estas tecnologías choca con el tema de la ética. Este problema surge al pensar que las herramientas remplazaran al hombre cuando el tema de la presión y confiabilidad supere a los especialistas, que dicho

sea de paso ya se está dando en el área de estudio de esta investigación a lo referido al cáncer de la piel. Por lo mismo es bueno recalcar que no se busca remplazar al hombre sino ser una ayuda frente a la gran demanda de los últimos tiempos. Aunque a esto sumarle la intervención del autor que menciona como un eje clave las habilidades blandas del doctor como algo irrepetible por la IA. Ya que temas como la cercanía y calidez que se da en el trato es algo que no se puede emular dada la frialdad de la misma tecnología con la que se está trabajando.

De todo lo tratado podemos sacar que el futuro de estas ciencias en conjunto, las IA y la dermatología, tendrán gran cabida en el siglo que se avecina. Por lo que el incurrir en esta investigación será de más ayuda a la hora de tratar este tipo de enfermedades de manera prudente, precisa, pero respetando el aspecto ético y los beneficios que solo pueden ser ofrecidos por especialistas en esta área. Buscando así automatizar la detección temprana del cáncer de la piel, como ayuda idónea para los especialistas.

### **3.8 ¿La IA amenaza al trabajo médico de los dermatólogos?**

Los aportes dados por (Martorell, y otros, 2021) podemos destacar su punto de vista y aportes más ligados al ámbito medico incluso mencionando que los ingenieros no le dan la importancia respectiva al tema, al menos no de la manera que cabría esperar de un especialista. Pudiendo ver así la necesidad de tener a algún especialista en esta área con el fin de brindar un servicio que no solo supla las expectativas informáticas sino médicas.

Aclararan también el tema de la ética sobre hasta dónde puede llegar viéndolo desde un punto de vista médico habla más de una simbiosis, que de por si sería el punto óptimo donde no se pierda la intervención humana, pero que no se desperdicie la tecnología brindada por la IA o que se piense que esta suplirá a los especialistas.

También se toca el tema de la introducción de esta tecnología al campo médico. Remarcando primero que debe darse de manera pausada, porque si bien la tecnología está avanzando a pasos agigantados el poder adaptarse a las mismas siempre es el dilema en estos casos. Todo esto se da debido a que se busca la simbiosis de especialistas y tecnología, pero también debemos mencionar que se buscó incluso dar un paso más allá de

lo investigado. No solo se trata de predecir quien tiene o no cáncer, sino tratarlo y adecuar dicha tecnología a la robótica con el fin de ser parte de las prácticas de los futuros dermatólogos.

### **3.9 La limpieza de imágenes por medio de algoritmos en la dermatología**

En primera instancia (Hoogstra, 2019) habla de la enfermedad del cáncer de la piel dentro de las que termina evaluando, seguidamente abarca el cómo muchas veces este tipo de enfermedades pasa desapercibido en su etapa más temprana.

Sin embargo, la parte más resaltante de la investigación médica es el abarcar los algoritmos automáticos para lesiones cutáneas. Esto es importante resaltarlo porque a lo largo de la investigación se le dio más énfasis al tema de los melanomas, pero con esta investigación podemos ver lo propio con los daños causados por la Queratosis Actínica. Se empieza describiendo la lesión para después realizar un mapa que involucre a sus características mediante redes neuronales y luego culminar con un aprendizaje profundo. Remarcando ya en última instancia la ya tocada regla ABCD.

De todo esto podemos rescatar los pasos que involucran a las lesiones cutáneas a la investigación, pudiendo realizar un paralelismo con lo visto en los melanomas que tienen su propia manera y pasos de tratar. Viendo así una clara diferenciación que se dejó en manifiesto desde el comienzo de la investigación.

### **3.10 FotoFinder AI**

Este es el más claro ejemplo de hasta donde se puede llegar la investigación y todo de la mano de (Haenssle, 2021) y todo su equipo a la hora de desarrollar esta IA. Con una amplia cantidad de imágenes y un largo historial que les respalda sirve para entender muchos puntos a la hora de realizar la investigación.

Se puede hacer un claro distinguir entre lo logrado ya por 1998 y los éxitos cosechados ya en 2021 por esta IA. Se tiene como un pilar de toda la investigación al Deep Learning, así como múltiples investigaciones de la mano de (Haenssle, 2021) y su equipo. Gracias a la herramienta Body Dermoscopy se puede conseguir imágenes de las lesiones

provenientes del cuerpo de los pacientes. Gracias a esta tecnología se puede tener una amplia cantidad de entradas que ayudaron a entrenar a esta IA para que ya hoy opere con mucha efectividad y eficacia.

Más allá de su contribución a la dermatología y lo relacionado a los vellos de la piel y el cabello podemos destacar la opción de acceder de alguna manera a los beneficios de esta IA y estudiar un poco el funcionamiento de esta para emular lo que en primer lugar le dio ese éxito y relacionarlo con la presente investigación. Siendo así que haremos un especial énfasis a las áreas de IA y cáncer de la piel tratadas por este especialista y su equipo. Por lo que de parte de la investigación se podría contribuir en lo referente a la Queratosis Actínica que es menos estudiada que los melanomas.

### **3.11 Clasificador de imágenes con Python y Tensorflow**

La elección de esta aplicación va de la mano con tener ya muy clara la manera en la que se va a desarrollar el presente trabajo de investigación. Partiendo de que Google Colab nos ofrece un ambiente de programación en Python gratuito y sin instalar nada previamente. Esto diendo de la mano con Tensorflow nos permitirá realizar la caja negra que detecte si la persona en cuestión padece o no de cáncer de la piel. Todo el proceso parte de tener una base de datos con imágenes de la enfermedad, que debe recabarse o en su defecto buscar entre las muchas bases de datos que ofrece la plataforma de manera gratuita.

Dentro de la aplicación mencionada (Ringa, Tu primer clasificador de imágenes con Python y Tensorflow, 2021) utiliza la base de datos 'fashion\_mnist' que cuenta con 60000 datos de entrenamiento y 10000 datos de pruebas. De manera análoga se trabajara solo que teniendo una entrada aproximada de 100000 imágenes de zonas que evidencian cáncer de la piel en su etapa temprana y una cantidad de datos de prueba acorde a la necesidad del proyecto. Al tratar con imágenes se debe de normalizar la información debido a que esta se encuentra trabajando con 255 valores propio de la escala de grises, añadiendo que de trabajarse con colores deben de utilizarse 3 canales propios del RGB que se verá en otro artículo a profundidad. Gracias a la herramienta plot podemos ver las imágenes que se están utilizando. Y gracias a keras podemos dar pie a la generación de redes neuronales.

Es relevante mencionar que dentro de las redes neuronales nuestra entrada son las imágenes, las capas ocultas serán generadas por la IA en cuestión mediante el entrenamiento. Que en resumidas cuentas se traduce en múltiples ecuaciones con pesos y márgenes de error, pero estas pecan de analizar solo imágenes similares a las de la búsqueda e independiente de la posición de las mismas. Siendo así necesario el posterior uso de redes neuronales convolucionales.

Al momento de procesar la información lo hacemos mediante un compilador que usa a Adam con optimizador y a `SparseCategoricalCrossentropy` como variable de sesgo. Los modelos se pueden entrenar mejor por medio de bloques, cuya cantidad depende de la persona, en este caso (Ringa, Tu primer clasificador de imágenes con Python y Tensorflow, 2021) utilizó 32 con fines intuitivos. Ya para finalizar uno puede mediante código en python crear una función que nos diga si que la IA predice o no con exactitud lo que se le pide, teniendo en el caso mostrado una precisión alrededor del 88%.

De todo esto podemos concluir que el uso de redes neuronales es muy eficaz a la hora de realizar procesos de detección de imágenes, pero que si queremos tener una mayor precisión es necesario utilizar las redes neuronales convolucionales.

### **3.12 Clasificación avanzada de imágenes con IA / ML (CNN)**

Aquí encontramos una comparativo de distintos modelos de detección de imágenes desde el más simple hasta el más complejo y eficiente. Ya habiendo abarcado el modelo más básico pasaremos a ver lo que diferencia a las redes neuronales convolucionales. Partamos del hecho que se basan en la detección de características y para esto usan dos o más capas de convolución y agrupación. En nuestro caso particular se podrían agrupar los distintos tonos y circunferencias de las heridas, así como el cambio brusco que tienen frente a la piel. Además, estas funcionan semejantes al ojo humano mediante neuronas simples en gran cantidad agrupadas por neuronas compuestas todo esto de la mano de David H. Hubel y Torsten Wiesel.

En la aplicación (Ringa, Redes Neuronales Convolucionales - Clasificación avanzada de imágenes con IA / ML (CNN), 2021) nos explica que para determinar los ejes

que conforman los contornos de las imágenes hace falta asignarle un valor numérico a cada píxel, siendo así que la computadora detecta la variación brusca entre un píxel y otro. Teniendo así que trabajar con grupos de 3x3 que tienen un píxel central. Donde los pesos de cada uno de los 9 valores irán variando según el aprendizaje de la red neuronal por machine learning. Esto es de por sí al trabajar con imágenes en blanco y negro ya que al recurrir a imágenes a color se debe realizar el proceso 3 veces debido a que el número de canales asciende a este número para luego juntarse y generar la misma cantidad de imágenes que en la escala de grises. Donde explotaremos los ejes como líneas que separan la herida o el lunar de la piel, además de previamente eliminar los vellos que tiene la misma para un análisis más preciso.

Mediante el proceso de agrupación (Ringa, Redes Neuronales Convolucionales - Clasificación avanzada de imágenes con IA / ML (CNN), 2021) nos explica que las imágenes ya analizadas tienden a decrecer en tamaño debido a que los píxeles se van tomando en grupos según desee la persona, que determinan la escala en la que se reducirá la imagen original, pero conservando los ejes más importantes y relegando lo menos necesario. Esto se traduce en poder determinar la evolución que padece el nevo displásico, ya que son lunares que se van acrecentando con el paso del tiempo o según el caso del paciente. Y de manera análoga en la queratosis actínica podemos ver la evolución de heridas que se generan en la piel y que la IA las siga detectando como tal.

Finalmente podemos acotar la inclusión de dos variables que hacen más eficaz el trabajo ya realizado como lo son el aumento de datos y el dropout. Este primero nos sirve para con unas entradas ya fijas el entrenamiento involucra también deformar las entradas de tal manera que se obtengan distintos puntos de vista de este. Esto nos es útil para que independiente de la posición de la imagen o el tamaño de esta se puede determinar con certeza si se trata de la enfermedad, ya que los casos siempre son diversos y las heridas no siempre están en la misma posición o del mismo tamaño. Mientras que el dropout nos ayuda a poder darle el máximo uso a las redes neuronales al limitar a aquellas que no estén trabajando adecuadamente o cuyo valor no sea relevante en el resultado, agilizando así el proceso de análisis de imágenes. Este valor de dropout puede ser añadido por nosotros según (Ringa, Redes Neuronales Convolucionales - Clasificación avanzada de imágenes

con IA / ML (CNN), 2021) pero se recomienda uno estándar de 0.5 para mejorar el resultado.

Podemos concluir así que la adición de las redes convolucionales, el aumento de datos y el dropout son las claves para poder determinar si una persona padece o no de cáncer de la piel de tipo nevo displásico o queratosis actínica con un margen de error mucho menor y la posibilidad de adaptarse a una mayor variedad de casos más realistas.

### **3.13 Cáncer de piel en el mundo**

Según (Skin, 2022) existen en promedio 13 millones de casos diagnosticados de cáncer de la piel en el mundo. El problema es tal que 1 de cada 3 cánceres diagnosticados terminan siendo de la piel con una mortandad aproximada de 65000 personas.

Se recalca la incidencia en jóvenes y por efecto de los rayos ultravioleta; Sin embargo, se enfatiza que con una educación adecuada de la enfermedad esta puede ser combatida de manera temprana. Dichas afirmaciones van de la mano con lo expuesto en el presente trabajo de investigación. De todo lo anterior podemos sacar que no estamos tratando con una enfermedad cualquiera; pero es manejable en la medida que se conozca cómo funciona. Se puede extrapolar también información vital a la hora de generar una base de datos que será usada en la predicción temprana de este cáncer por medio de redes neuronales en Python.

También podemos encontrar información relevante sobre los dos tipos de cáncer de la piel que serán abordados. En primer lugar, podemos hablar de la Queratosis Actínica es el tipo de cáncer de la piel más propenso a desarrollarse de la mera exposición solar; aunque este solo en un 5% al 10% de los casos termina agravándose como cáncer. En los factores que se analizan según (Skin, 2022) podemos ver los siguientes:

**Tabla 1** Factores Asociados a la Queratosis Actínica

| Característica | Descripción                                                                                                                                                                                                                                        |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Textura        | Superficie que puede variar de plana a levemente elevada, con apariencia escamosa, áspera, o incluso presentar una protuberancia en forma de cuerno.                                                                                               |
| Tamaño         | Las dimensiones varían desde un punto diminuto hasta una pulgada de diámetro                                                                                                                                                                       |
| Color          | Puede ser de tonos rojos, tostados, rosados, color piel, marrón o plateado.                                                                                                                                                                        |
| Ubicación      | Frecuentemente afecta zonas expuestas al sol, como la cara, labios, orejas, cuero cabelludo, hombros, cuello, manos y antebrazos. La queratosis actínica en los labios, conocida como queilitis actínica, afecta principalmente el labio inferior. |

Fuente: Adaptado de Skin Cancer Foundation (2022).

En el caso de los melanomas se afirma que son los más mortales y que la formación de estos va de la mano con la melanina. Siendo así este se divide en 4 distintos tipos según (Skin, 2022):

- Melanoma de extensión superficial: El tipo más común
- Lentigo maligno: Se da mayormente en gente mayor.
- Melanoma lentiginoso acral: Se desarrolla en personas de color.
- Melanoma nodular: El más letal de todos con un 10% a 15% de los casos.

También se señala que en algunos casos es difícil de detectar debido a su naturaleza cambiantes según lo detalla la regla ABCDE, en el apartado de Evolución, siendo que no basta con solo detectarlos por el color o tamaño, sino también con cualquier sospecha de alteraciones en la piel.

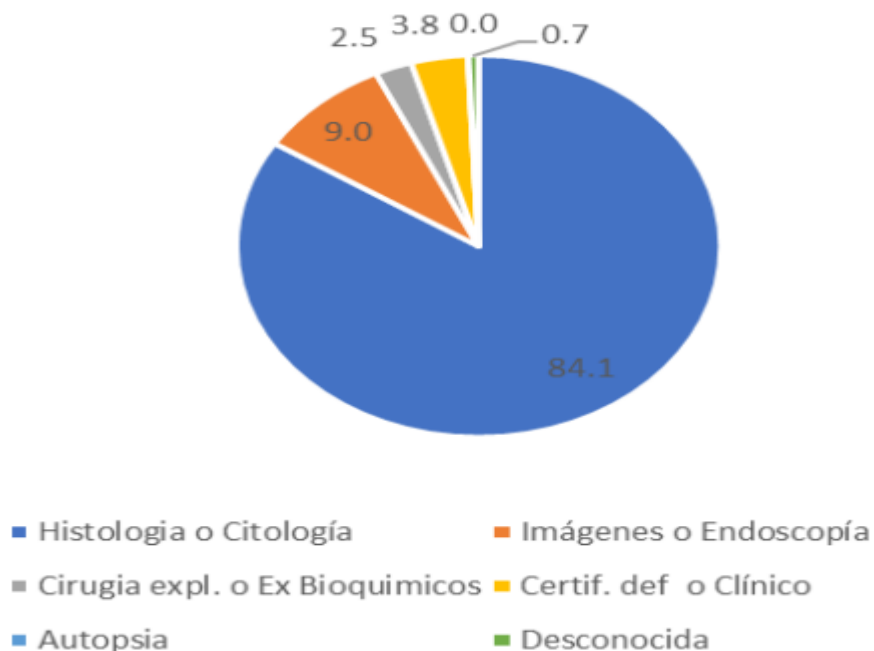
Los primeros datos recabados nos sirven a la hora de analizar las imágenes que serán procesadas, pues estas se basan en características, las cuales se detallan la tabla 1. Mientras que los datos obtenidos del melanoma nos hablan de una mayor variedad de estos datos clave a la hora de determinar una salida correcta, donde por temas de complejidad en primera instancia se evaluarán a todos los melanomas como un grupo en conjunto.

### 3.14 Situación de cáncer en el Perú

Según (Revilla T., 2021) en el año 2020 hubo un total de 19.3 millones de casos de cáncer en el Perú con una tasa de muerte de 1.4 millones y una tendencia que incrementara en 32% llegando a los 5 millones por año. Sin embargo, en este año no se abarca los temas relacionados al cáncer de la piel, solo se recalca que el cáncer en general está ligado a un sedentarismo y a una falta de actividad física.

Ya en el año 2019 si encontramos una alta incidencia de cáncer de la piel en las estadísticas presentadas por el Ministerio de salud como se puede ver en la siguiente gráfica:

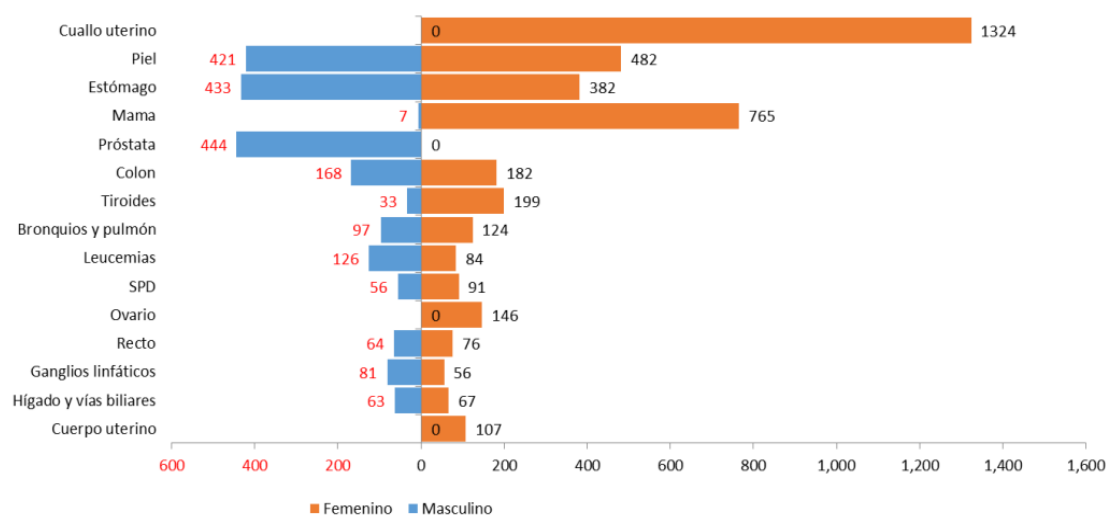
**Figura 6** Estadística de cáncer por tipo



Fuente: Ministerio de Salud

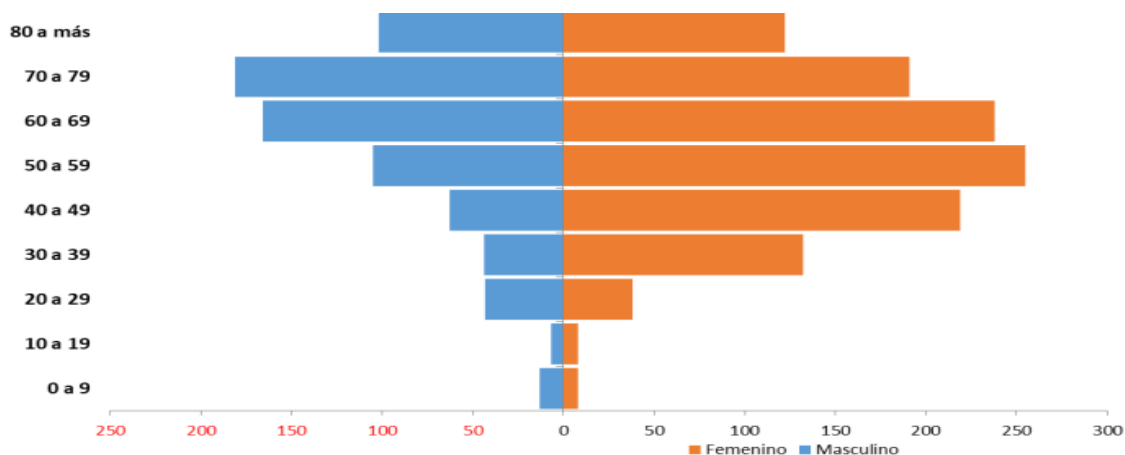
De la misma podemos ver que los problemas de histología y citología son predominantes con respecto a los demás tipos de cáncer. En la siguiente gráfica también podemos observar como este cáncer es de los 3 más recurrentes en el caso de los hombres y en el último gráfico vemos como el cáncer es más incidente en personas mayores, pero que los nuevos casos se dan en jóvenes debido a sus estilos de vida y exposición a los rayos UV.

**Figura 7 Estadística de cáncer según el género**



Fuente: Ministerio de Salud

**Figura 8 Estadística de cáncer según la edad**



Fuente: Ministerio de Salud

### **3.15 Caso Regional sobre Cáncer de Piel en Perú: Exposición Solar en Comerciantes Ambulantes en Chiclayo**

Los conocimientos y prácticas de exposición solar en comerciantes ambulantes de un mercado en Chiclayo aportan un análisis detallado del impacto de la radiación solar en la salud, así como de los factores que determinan la adopción de medidas preventivas en una población laboralmente expuesta. Entre los 291 comerciantes encuestados, se identificaron riesgos significativos en relación con enfermedades y daños causados por la exposición solar prolongada, destacando una prevalencia del 90,03% de conocimiento sobre el riesgo de cáncer de piel, seguido de un 96,22% sobre el riesgo de sarna, y un 97,94% sobre el riesgo de celulitis (Tabla 2). En cuanto a los efectos del envejecimiento cutáneo, el 53,6% reconoció la asociación entre la exposición solar y la aparición de manchas y arrugas, y el 21,6% identificó la relación con picazón y sarpullido según el estudio de (Sempértgui-Ruiz, 2020).

Respecto a los daños oculares, el 56,01% de los comerciantes estaba informado sobre el riesgo de desarrollar cataratas debido a la exposición solar, mientras que el 34,36% reconoció el riesgo de miopía y el 7,2% de sangrado en los ojos. Además, el 80,41% identificó correctamente el horario de mayor radiación ultravioleta entre las 10 a.m. y las 3 p.m., lo que refleja un conocimiento adecuado en esta área (Tabla 2). Sin embargo, persisten ciertas creencias erróneas: el 16,49% pensaba que en un día nublado no existe riesgo de quemaduras solares, y el 16,83% no reconocía los peligros de la exposición solar en invierno.

En términos de características sociodemográficas, los comerciantes de foto tipo IV y III fueron los más frecuentes (48,8% y 24,7%, respectivamente), lo cual indica una predisposición moderada a alta para sufrir daños por exposición solar en condiciones de radiación intensa. En cuanto a los hábitos de foto protección, solo el 24,7% de los encuestados utilizaba bloqueador solar, a menudo debido a recomendaciones familiares (63,9%).

El análisis de estos resultados muestra que, a pesar de ciertos niveles de conocimiento sobre los riesgos asociados a la exposición solar, la aplicación de prácticas preventivas sigue siendo limitada. Estos hallazgos subrayan la importancia de políticas de salud pública orientadas a incrementar tanto el acceso a productos de protección solar como la educación en foto protección para reducir la incidencia de cáncer de piel y otras condiciones relacionadas con la exposición prolongada en contextos laborales informales.

**Tabla 2** Conocimientos de los comerciantes ambulantes de un mercado en Lambayeque, Chiclayo - setiembre, 2017

|                                                 |                                                             | Correcta | %      | Incorrecta | %      |
|-------------------------------------------------|-------------------------------------------------------------|----------|--------|------------|--------|
| Enfermedad por exposición solar prolongada      | Estrías                                                     | 12       | 4,12%  | 279        | 95,88% |
|                                                 | Cáncer de piel                                              | 262      | 90,03% | 17         | 9,07%  |
|                                                 | Sarna                                                       | 11       | 3,78%  | 268        | 96,22% |
|                                                 | Celulitis                                                   | 6        | 2,06%  | 273        | 97,94% |
| Característica del envejecimiento de la piel    | Verrugas, lunares                                           | 39       | 13,4%  | 240        | 86,6%  |
|                                                 | Acné, ronchas                                               | 33       | 11,3%  | 246        | 88,7%  |
|                                                 | Manchas, arrugas                                            | 156      | 53,6%  | 135        | 46,3%  |
|                                                 | Picazón, sarpullido                                         | 63       | 21,6%  | 228        | 78,4%  |
| Daños a los ojos por exposición solar           | Estrabismo                                                  | 7        | 2,4%   | 284        | 97,6%  |
|                                                 | Sangrado en los ojos                                        | 21       | 7,2%   | 270        | 92,8%  |
|                                                 | Miopía                                                      | 100      | 34,36% | 191        | 65,64% |
|                                                 | Cataratas                                                   | 163      | 56,01% | 128        | 43,99% |
| Horario donde los rayos del sol son más fuertes | Entre las 10 am y 3pm                                       | 234      | 80,41% | 57         | 19,59% |
|                                                 | Antes de las 7 am                                           | 6        | 2,06%  | 285        | 97,94% |
|                                                 | Entre las 3pm y 6pm                                         | 27       | 9,27%  | 264        | 90,73% |
|                                                 | Entre las 7am y 10 am                                       | 24       | 8,24%  | 267        | 91,76% |
| Lo cierto con respecto a los rayos del sol      | En un día nublado no te puedes quemar                       | 48       | 16,49% | 243        | 83,51% |
|                                                 | La gente de piel blanca es más sensible a los rayos del sol | 154      | 52,92% | 137        | 47,08% |
|                                                 | Si estoy bajo el sol y no siento calor, no me quemaré       | 40       | 13,74% | 251        | 86,26% |
|                                                 | El sol en el invierno no es peligroso                       | 49       | 16,83% | 242        | 83,17% |

Fuente: Obtenido de Conocimientos y hábitos de exposición solar en comerciantes ambulantes de un mercado de Chiclayo, Perú (Sempértegui-Ruiz et al., 2020).

## Capítulo 4 METODOLOGÍA E IMPLEMENTACIÓN

En este capítulo veremos los pasos a seguir para realizar la detección de los melanomas y la queratosis actínica en casos reales por medio de un dataset público al cual expondremos a una IA que maneja un entramiento mediante machine learning y redes neuronales convolucionales. La idea es saber identificar entre los distintos tipos de enfermedades de la piel y el cáncer que se pueda presentar en este. De modo que sepamos si es de tipo maligno o benigno además de generar un diagnóstico en el proceso mediante el uso de múltiples herramientas.

### 4.1 Herramientas a usar

#### 4.1.1 Kaggle:

Según (Sculley, 2022) podemos ver los aspectos más importantes de Kaggle como herramienta para la codificación y obtención de la data del proyecto de investigación. Dentro de la misma plataforma se realizan varios concursos de diferentes indoles, en nuestro caso nos importa una esponsorizada hecha por (ISIC, 2020) los cuales nos brindaron la data a utilizar.

Dentro de la plataforma podemos ver además muchos tipos de códigos que se corren en la misma plataforma, los cuales sirven para aprender y formar un camino a seguir. Es un avance progresivo que se va obtenido producto de ver muchos códigos de muchos autores. Es sobretodo eficiente a la hora de conocer funciones en los códigos que pueden ser similares. Sin embargo, siento que lo más destacado es el tema de las librerías a usar.

En el caso de esta página podemos fácilmente utilizar la data desde la misma plataforma, cosa que ayuda debido al tamaño de las imágenes y al preprocesamiento necesario. También debemos señalar que nos provee de muchos gráficos que se generan por medio de código de Python de manera análoga a lo que muestra Google Colab.

Con todo esto podemos concluir que el uso de esta herramienta será crucial en el desarrollo de la tesis y para obtener los resultados finales.

#### 4.1.2 RapidMiner:

Al dirigirnos a la página de RapidMiner podemos ver un interfaz amigable que nos permite descargar la herramienta para los fines de la investigación. Quien se encarga de la misma es el CEO, (Lee, 2022), gracias a él podemos evaluar los clústeres en la data de “test.csv”.

RapidMiner es una herramienta versátil que nos permite analizar una data en formatos de tablas como xlsx o csv. Los cuales se evalúan mediante conectores de forma manual o de manera automática como lo presentamos en siguientes puntos.

En la misma podemos obtener valores estadísticos que pueden ser interpretados, se puede tener una precisión de la variable a evaluar mediante los demás campos sabiendo el peso que tiene cada uno de estos en la misma. De la misma forma se puede analizar cada campo por separado lo que nos ayuda a tener un primer vistazo de la data a evaluar.

#### 4.1.2 Tableau:

Esta herramienta es útil para el manejo de grandes volúmenes de datos como lo es el caso de la tesis. (Stolte, Hanrahan, & Chabot, 2022) cofundadores de la empresa nos muestran que la herramienta es útil para ver de manera gráfica las datas mediante gráficos que son bastante interactivos y que se pueden identificar de manera inmediata con el tema de tamaño, colores y formas.

Dentro de lo que se está viendo analizamos los campos que tiene el dataset “train.csv” comparando los campos de forma estratégica para manejar a gran escala cómo se comportan previo a la codificación, siendo también viable en el caso se depuren los datos.

Podemos así concluir que el uso de una herramienta como esta es clave para poder mostrar dentro de una exposición, así como explicar de forma rápida de que trata cada conjunto de variables.

#### 4.1.2 Knime:

La última herramienta por utilizar es aquella que se usa de manera semejante a las redes neuronales, conectando nodos a fin de generar todo un proceso y entramado. De la mano de (Berthold, 2022) vemos un potencial tremendo y muy preciso al evaluar conjuntos de datos grande.

Ya hubo un uso de esta en donde nos percatamos del uso de la matriz de confusión y la curva de ROC que son claves para el inicio y final de la tesis teniendo un contraste claro de aquello a lo que se llegó luego del entrenamiento y codificación.

Gracias a esta podemos simular lo que será en un futuro de manera rápida, además podemos comparar los datos recogidos en las conclusiones. Podemos finalizar así con estas herramientas viendo cómo se complementan y tienen usos similares que nos sirven para poder esquematizar, visualizar y entender mejor la data que estamos evaluando y su impacto frente a las redes neuronales a las que serán expuestas.

## 4.2 Dataset test en Kaggle

Conociendo la influencia del cáncer de la piel y la cantidad de muertes que genera, podemos añadir que un 75% de estas muertes se deben a los melanomas, pues como se

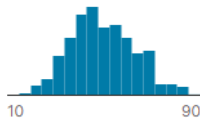
mostró anteriormente es el más letal de todos. Sin la ayuda de algoritmos de detección de imágenes contextuales que de la mano con una IA ayuden a los dermatólogos, muchos pacientes pueden tener un diagnóstico erróneo. Es por ello por lo que la organización Society for Imaging Informatics in Medicine (SIIM) nos ayudó en la búsqueda de imágenes médicas promoviendo la investigación e innovación mediante un concurso público que ayudo a incrementar la cantidad de data a lo largo del tiempo. De la mano con la International Skin Imaging Collaboration (ISIC) tenemos la mayor colección disponible públicamente de imágenes dermatoscopias de lesiones cutáneas con control de calidad. (ISIC, 2020)

Empecemos mencionando los datos de prueba que son un total de 10982, los cuales tienen las siguientes cabeceras:

- Image\_name: Nombre de la imagen con el formato ISIC
- Patient\_id: Código del paciente con el formato IP
- Sex: Sexo del paciente, siendo un 57% hombres y un 43% mujeres
- Age\_approx: Edad aproximada del paciente en un intervalo de 10 a 90 años
- Anatom\_site\_general\_challenge: Parte del cuerpo de donde se extrajo la imagen a evaluar.

**Figura 9** Detallado de las variables de la data train.csv

< test.csv (490.92 kB)

| Detail                 | Compact       |     | Column |     |                                                                                    |                       |                 |
|------------------------|---------------|-----|--------|-----|------------------------------------------------------------------------------------|-----------------------|-----------------|
| ▲ image_name           | ▲ patient_id  |     | ▲ sex  |     | # age_approx                                                                       | ▲ anatom_site_gene... |                 |
| 10982<br>unique values | IP_3579794    | 2%  | male   | 57% |  | torso                 | 53%             |
|                        | IP_6877747    | 1%  | female | 43% |                                                                                    | lower extremity       | 23%             |
|                        | Other (10634) | 97% |        |     |                                                                                    | Other (2634)          | 24%             |
|                        |               |     |        |     |                                                                                    |                       |                 |
| ISIC_0052060           | IP_3579794    |     | male   |     | 70.0                                                                               |                       |                 |
| ISIC_0052349           | IP_7782715    |     | male   |     | 40.0                                                                               |                       | lower extremity |
| ISIC_0058510           | IP_7960270    |     | female |     | 55.0                                                                               |                       | torso           |
| ISIC_0073313           | IP_6375035    |     | female |     | 50.0                                                                               |                       | torso           |
| ISIC_0073502           | IP_0589375    |     | female |     | 45.0                                                                               |                       | lower extremity |
| ISIC_0074618           | IP_4438996    |     | male   |     | 50.0                                                                               |                       | lower extremity |
| ISIC_0076801           | IP_3110660    |     | male   |     | 45.0                                                                               |                       | upper extremity |
| ISIC_0077586           | IP_8578442    |     | male   |     | 50.0                                                                               |                       | lower extremity |
| ISIC_0082004           | IP_0817448    |     | female |     | 45.0                                                                               |                       | torso           |
| ISIC_0082785           | IP_3634181    |     | male   |     | 65.0                                                                               |                       | lower extremity |
| ISIC_0085332           | IP_7899071    |     | male   |     | 40.0                                                                               |                       | torso           |
| ISIC_0087243           | IP_7746230    |     | male   |     | 60.0                                                                               |                       | torso           |
| ISIC_0088167           | IP_0581821    |     | female |     | 30.0                                                                               |                       | upper extremity |
| ISIC_0089356           | IP_8564532    |     | female |     | 65.0                                                                               |                       | torso           |
| ISIC_0090283           | IP_9657527    |     | male   |     | 55.0                                                                               |                       | head/neck       |
| ISIC_0092481           | IP_4288676    |     | female |     | 55.0                                                                               |                       | lower extremity |

Fuente: Society for Imaging Informatics in Medicine (SIIM) e International Skin Imaging Collaboration (ISIC)

A continuación, se pueden observar algunas de las imágenes de prueba de la mano de Society for Imaging Informatics in Medicine (SIIM) e International Skin Imaging Collaboration (ISIC), que como se puede apreciar tienen una buena variedad los cuales nos servirán de base para el posterior entrenamiento de la data.

**Figura 10** Imágenes de la data test.csv



Fuente: *Society for Imaging Informatics in Medicine (SIIM) e International Skin Imaging Collaboration (ISIC)*

Luego de tener de base poco más de 10,000 imágenes podemos empezar a trabajar mediante el uso de redes neuronales convolucionales con 33126 imágenes, las cuales

cuentan con nuevas cabeceras que incluyen el tema del diagnóstico y el tipo de cáncer de piel, las cuales veremos a continuación:

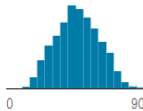
- Image\_name: Nombre de la imagen con el formato ISIC
- Patient\_id: Código del paciente con el formato IP
- Sex: Sexo del paciente, siendo un 57% hombres y un 43% mujeres
- Age\_approx: Edad aproximada del paciente en un intervalo de 10 a 90 años
- Anatom\_site\_general\_challenge: Parte del cuerpo de donde se extrajo la imagen a evaluar.
- Diagnosis: Nos da un diagnóstico general de la imagen pudiendo saber si es un melanoma o nevo displásico, por defecto ya un 16% se consideran de este último, mientras que un 82% aun es desconocido, esperando a ser evaluado.
- Benign\_malignant: Este nos indica si una imagen muestra un cáncer de piel de tipo maligno o benigno, que por efectos de la investigación se espera hallar casos en su estado benigno donde aún puede ser tratado por especialistas en dermatología.
- Target: Es la variable que determina si el resultado fue favorable o por el contrario no lo fue, además de ser la variable que nos permite evaluar la data en general bajo varios programas manteniendo un formato de 0 y 1.

También se adjunta las imágenes que van a ser entrenadas que como pasaba en el otro caso son bastante variadas. Pero si hay algo que destacar es la presencia de vello en más de una de ellas las cuales pueden obstaculizar el estudio por lo que pueden ser removidas mediante un algoritmo, así como tener la consideración pertinente en la tabla general.

**Tabla 3** Detallado de las variables de la data train.csv

< train.csv (2.06 MB)

Detail Compact Column

| ▲ image_name                  | ▲ patient_id                 | ▲ sex                                   | # age_approx                                                                      | ▲ anatom_site_gene...                                | ▲ diagnosis                                | ▲ benign_malignant         | # target                                                                            |
|-------------------------------|------------------------------|-----------------------------------------|-----------------------------------------------------------------------------------|------------------------------------------------------|--------------------------------------------|----------------------------|-------------------------------------------------------------------------------------|
| <b>33126</b><br>unique values | <b>2056</b><br>unique values | male 52%<br>female 48%<br>Other (65) 0% |  | torso 51%<br>lower extremity 25%<br>Other (7864) 24% | unknown 82%<br>nevus 16%<br>Other (809) 2% | benign 98%<br>malignant 2% |  |
| ISIC_2637011                  | IP_7279968                   | male                                    | 45.0                                                                              | head/neck                                            | unknown                                    | benign                     | 0                                                                                   |
| ISIC_0015719                  | IP_3075186                   | female                                  | 45.0                                                                              | upper extremity                                      | unknown                                    | benign                     | 0                                                                                   |
| ISIC_0052212                  | IP_2842074                   | female                                  | 50.0                                                                              | lower extremity                                      | nevus                                      | benign                     | 0                                                                                   |
| ISIC_0060279                  | IP_6890425                   | female                                  | 45.0                                                                              | head/neck                                            | unknown                                    | benign                     | 0                                                                                   |
| ISIC_0074268                  | IP_8723313                   | female                                  | 55.0                                                                              | upper extremity                                      | unknown                                    | benign                     | 0                                                                                   |
| ISIC_0074311                  | IP_2950485                   | female                                  | 40.0                                                                              | lower extremity                                      | unknown                                    | benign                     | 0                                                                                   |
| ISIC_0074542                  | IP_4698288                   | male                                    | 25.0                                                                              | lower extremity                                      | unknown                                    | benign                     | 0                                                                                   |
| ISIC_0075663                  | IP_6017204                   | female                                  | 35.0                                                                              | torso                                                | unknown                                    | benign                     | 0                                                                                   |
| ISIC_0075914                  | IP_7622888                   | male                                    | 30.0                                                                              | torso                                                | unknown                                    | benign                     | 0                                                                                   |
| ISIC_0076262                  | IP_5075533                   | female                                  | 50.0                                                                              | lower extremity                                      | unknown                                    | benign                     | 0                                                                                   |
| ISIC_0076545                  | IP_9802602                   | male                                    | 55.0                                                                              | upper extremity                                      | unknown                                    | benign                     | 0                                                                                   |
| ISIC_0076742                  | IP_2318163                   | male                                    | 75.0                                                                              | upper extremity                                      | unknown                                    | benign                     | 0                                                                                   |
| ISIC_0076995                  | IP_2235340                   | female                                  | 55.0                                                                              | torso                                                | nevus                                      | benign                     | 0                                                                                   |
| ISIC_0077472                  | IP_3691360                   | female                                  | 40.0                                                                              | torso                                                | unknown                                    | benign                     | 0                                                                                   |
| ISIC_0077735                  | IP_1109756                   | male                                    | 70.0                                                                              | torso                                                | unknown                                    | benign                     | 0                                                                                   |
| ISIC_0078703                  | IP_7279968                   | male                                    | 45.0                                                                              | torso                                                | unknown                                    | benign                     | 0                                                                                   |
| ISIC_0078712                  | IP_2189124                   | male                                    | 40.0                                                                              | lower extremity                                      | unknown                                    | benign                     | 0                                                                                   |
| ISIC_0079038                  | IP_5295061                   | male                                    | 70.0                                                                              | torso                                                | unknown                                    | benign                     | 0                                                                                   |
| ISIC_0080512                  | IP_1870306                   | male                                    | 75.0                                                                              | torso                                                | unknown                                    | benign                     | 0                                                                                   |
| ISIC_0080752                  | IP_2613684                   | male                                    | 50.0                                                                              | torso                                                | unknown                                    | benign                     | 0                                                                                   |
| ISIC_0080817                  | IP_7318404                   | male                                    | 50.0                                                                              | lower extremity                                      | unknown                                    | benign                     | 0                                                                                   |

Fuente: Society for Imaging Informatics in Medicine (SIIM) e International Skin Imaging Collaboration (ISIC)

**Figura 11** Imágenes de la data train.csv



Fuente: *Society for Imaging Informatics in Medicine (SIIM) e International Skin Imaging Collaboration (ISIC)*

### 4.3 Clústeres en RapidMiner

En este apartado se hará uso de la herramienta RapidMiner, así como de la data “test.csv” para poder determinar la distribución de los grupos que ya han sido analizados

previamente, siendo estos un total de 10982. Veremos los distintos resultados que arroja el programa e interpretaremos lo que significan.

En primer lugar, veremos que la variable designada a la hora de realizar los clústeres fue la de “edad aproximada” debido a su factor numérico. Por otro lado, las otras variables o eran cadenas, caracteres o enteros de 0 y 1. A continuación se ve un resumen gracias al k-means:

**Figura 12** Resumen de k-means según la edad aproximada

### k-Means - Summary

Number of Clusters: 2

#### Cluster 0

5,676

age\_approx is on average 28.63% larger

#### Cluster 1

5,306

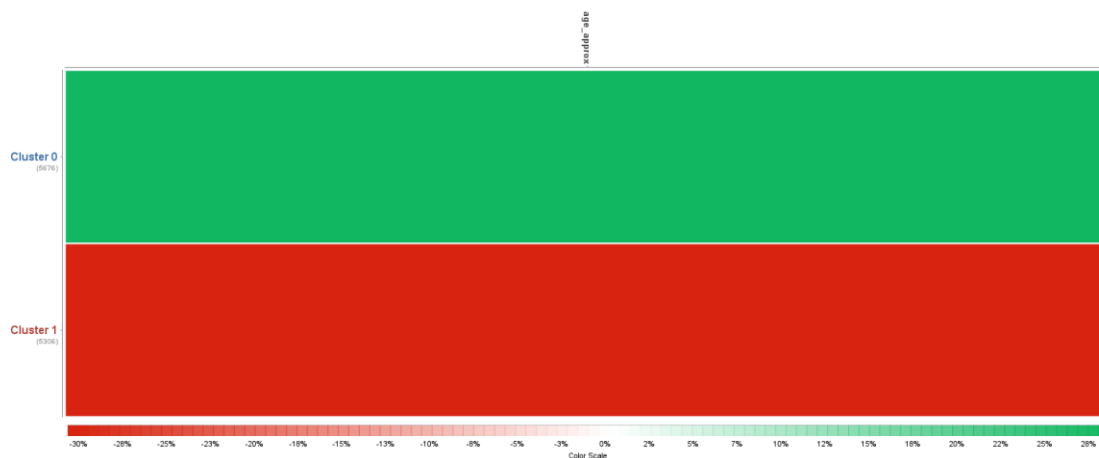
age\_approx is on average 30.63% smaller

Fuente: Elaboración propia.

Posterior a ello podemos ver un mapa de calor que completando la información de calor se ven los intervalos en porcentaje de las edades. Siendo en el clúster 0 de 28.63% más grandes siendo representado por el color verde. Mientras que el clúster 1 tiene 30.63% representado por el color rojo.

**Figura 13** Mapa de calor de k-means según la edad aproximada

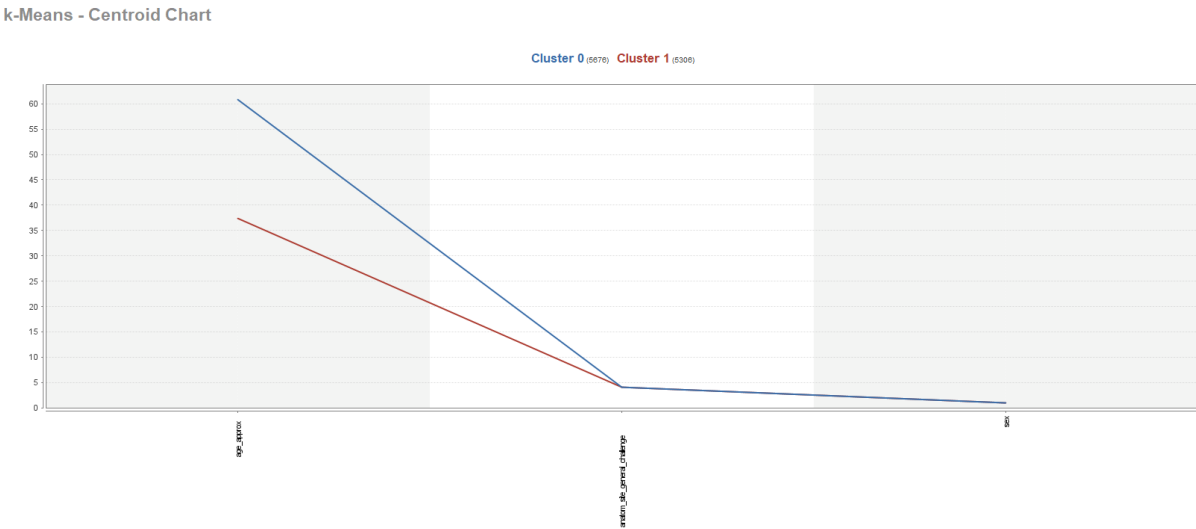
k-Means - Heat Map



Fuente: Elaboración propia.

En el caso del centroide se puede apreciar de mejor manera la distribución de las edades mediante una gráfica, siendo la línea de color azul el clúster 0 y la de color rojo, el clúster 1. Viendo como la primero tiene un pico mayor a los 60 años y la segunda ronda entre los 40 y 35 años.

**Figura 14** Gráfico de centroide de k-means según la edad aproximada



Fuente: Elaboración propia.

Justamente lo antes mencionados se ve más específico en esta tabla, donde la edad aproximada del clúster 0 es de 61 años y la del clúster 1 de 37 años. En ambos casos se redondea debido a que estamos hablando de edades, pero en caso de que se deseen tener en cuenta meses y días los valores se encuentran en la tabla. En el caso de las partes del cuerpo y del sexo no se encuentra nada relevante para el análisis.

**Tabla 4** Tabla de centroide de k-means

k-Means - Centroid Table

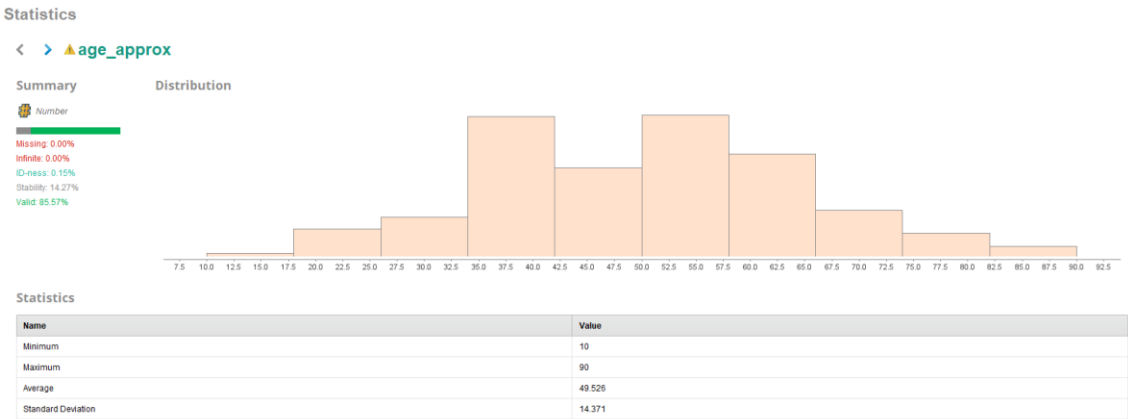
| Cluster   | age_approx | anatom_site_general_challenge | sex |
|-----------|------------|-------------------------------|-----|
| Cluster 0 | 60.842     | 4                             | 1   |
| Cluster 1 | 37.420     | 4                             | 1   |

Fuente: Elaboración propia.

Pasando al apartado estadístico arrojado por el programa vemos la distribución de las variables empezando por la edad aproximada. Viendo aquí que los picos de edad están en los rangos de 35 y 43 años junto a los 50 y 58 años aproximadamente. Siendo que la media de edad final es de 50 años. Donde la mínima edad es de 10 años, la máxima de 90

años sirviendo para que el análisis final ocupe un amplio público objetivo. Y finalizamos con la desviación estándar de 14.371.

Figura 15 Estadísticas según la edad aproximada



Fuente: Elaboración propia.

Vemos que en las partes afectadas hay diferentes resultados siendo que el torso es quien se lleva la mayor cantidad de casos con un total de 5847 ocupando así el 55%. En el caso de las extremidades superiores tenemos que son un total de 2501 ocupando el 23.53% del total. En el caso de las extremidades inferiores tenemos que son 1573 siendo así el 14.80% del total. Por otra parte, el cuello y la cabeza son apenas 576 con el 5.42% de casos totales. Por otro lado, las palmas y suelas son 108 casi quedándose al final con 1.02%. Para terminar, se ve que en la zona oral y genital hay un total de 26 casos con un 0.24%, añadiendo que se perdieron un 3.20% de los casos.

Figura 16 Estadísticas según el lugar del cuerpo afectado



Fuente: Elaboración propia.

Analizando la variable del sexo solamente tenemos dos poblaciones diferenciadas siendo que son más los casos de hombres estudiados con un 56.96% del total, lo que representa 6255 casos. Mientras que las mujeres no se quedan atrás con 4727 casos siendo el 43.04% de los casos totales.

Figura 17 Estadísticas según el sexo



Fuente: Elaboración propia.

Casi para concluir vemos las correlaciones entre variables donde se tomaron todas las variables junto a un análisis cruzado de las mismas que nos puede servir para un análisis posterior y para ver todo a grandes rasgos.

Tabla 5 Correlaciones de los atributos

Correlations

| Attributes                                      | age_approx | anatom... | anatom... | anatom_sit... | anatom... | anatom... | anatom... | sex = male |
|-------------------------------------------------|------------|-----------|-----------|---------------|-----------|-----------|-----------|------------|
| age_approx                                      | 1          | 0.057     | 0.003     | 0.179         | -0.043    | -0.044    | -0.054    | 0.109      |
| anatom_site_general_challenge = head/neck       | 0.057      | 1         | -0.128    | -0.043        | -0.023    | -0.251    | -0.096    | 0.012      |
| anatom_site_general_challenge = lower extremity | 0.003      | -0.128    | 1         | -0.099        | -0.054    | -0.579    | -0.222    | -0.077     |
| anatom_site_general_challenge = MISSING         | 0.179      | -0.043    | -0.099    | 1             | -0.018    | -0.194    | -0.074    | 0.109      |
| anatom_site_general_challenge = palms/soles     | -0.043     | -0.023    | -0.054    | -0.018        | 1         | -0.106    | -0.041    | 0.047      |
| anatom_site_general_challenge = torso           | -0.044     | -0.251    | -0.579    | -0.194        | -0.106    | 1         | -0.436    | 0.042      |
| anatom_site_general_challenge = upper extremity | -0.054     | -0.096    | -0.222    | -0.074        | -0.041    | -0.436    | 1         | -0.045     |
| sex = male                                      | 0.109      | 0.012     | -0.077    | 0.109         | 0.047     | 0.042     | -0.045    | 1          |

Fuente: Elaboración propia.

Finalizamos esta primera parte del uso del RapidMiner ya que para la segunda se requiere una evaluación previa mediante el código para poder adaptar la data de “train.csv”.

#### 4.4 Dataset train en Tableau

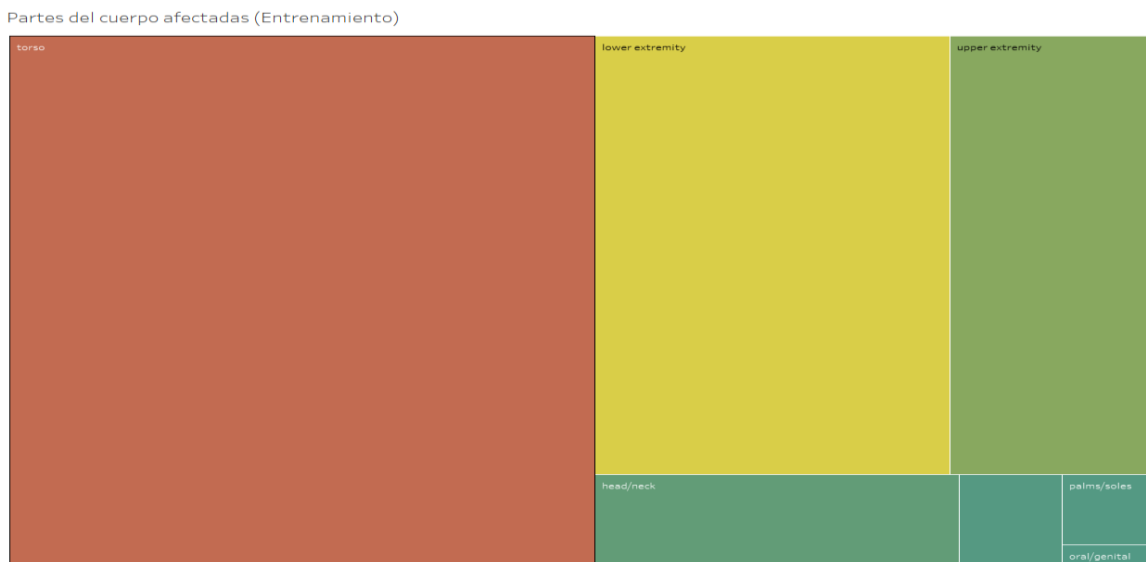
En primer lugar, debemos de tratar este segundo dataset de forma distinta, empezando por el hecho de que debemos tratarlo antes de utilizarlo de lo contrario generara resultados inadecuados.

Como prueba de esto se analizó la data mediante la herramienta Knime donde lo esperado estaba lejos de ser lo que se busca. Esto se debe a que en el apartado del diagnóstico la mayoría de los casos son desconocidos.

Primeramente, analizaremos mediante la herramienta Tableau las variables y sus cantidades mediante diagramas que nos proporciona la herramienta:

- Partes del cuerpo afectadas: En el caso de las partes del cuerpo que tenemos disponibles podemos ver como el torso sigue siendo la que mayor cantidad de personas tiene con un total de 16845, por otro lado, las extremidades inferiores le siguen con 8417, luego las extremidades superiores con 4983, cabeza y cuello con 1855, palmas con 375 y la vía oral con 124.

***Figura 18 Partes del cuerpo afectadas***

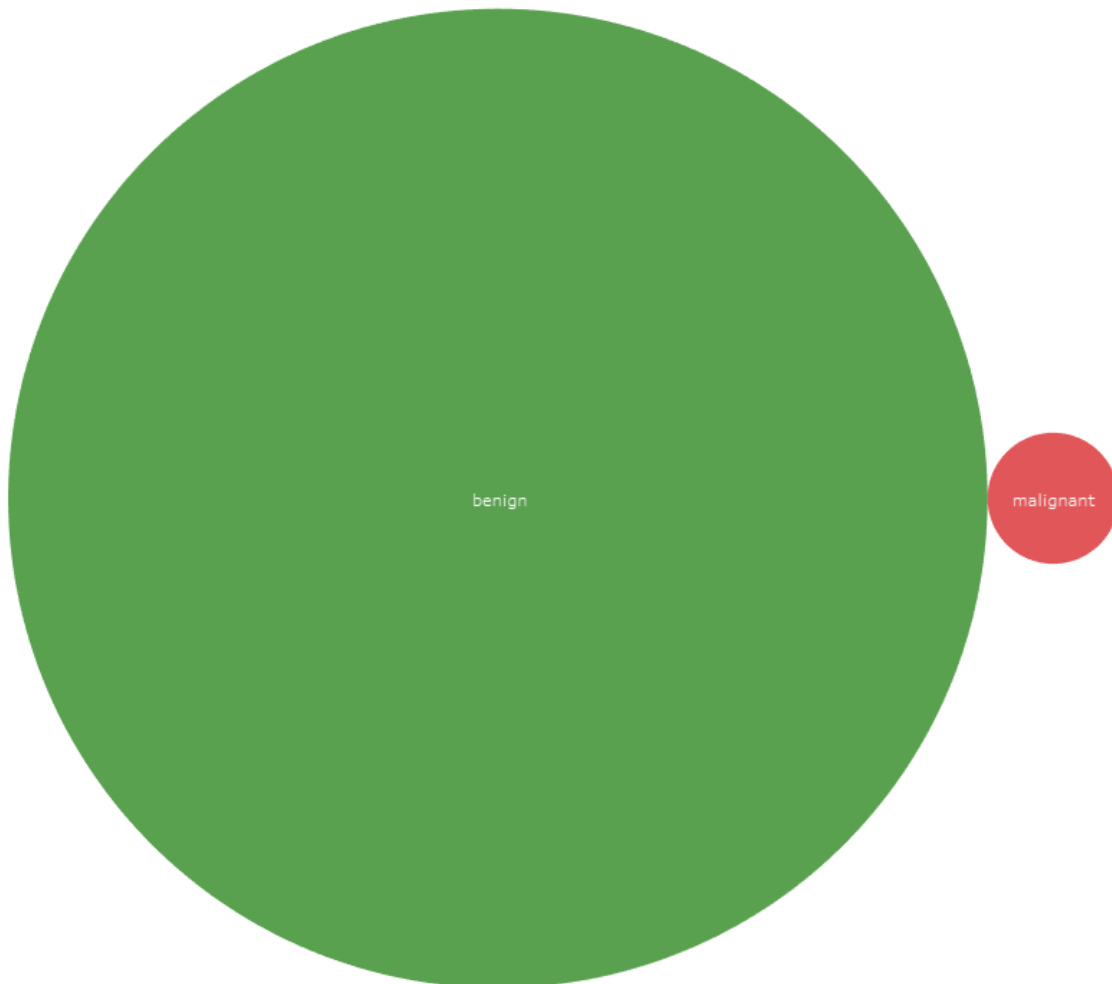


Fuente: Elaboración propia.

- Benigno y maligno: Es este caso vemos una mayoría por parte de los casos benignos siendo 32542 y los malignos siendo 584.

***Figura 19 Benigno o maligno***

Benigno o maligno



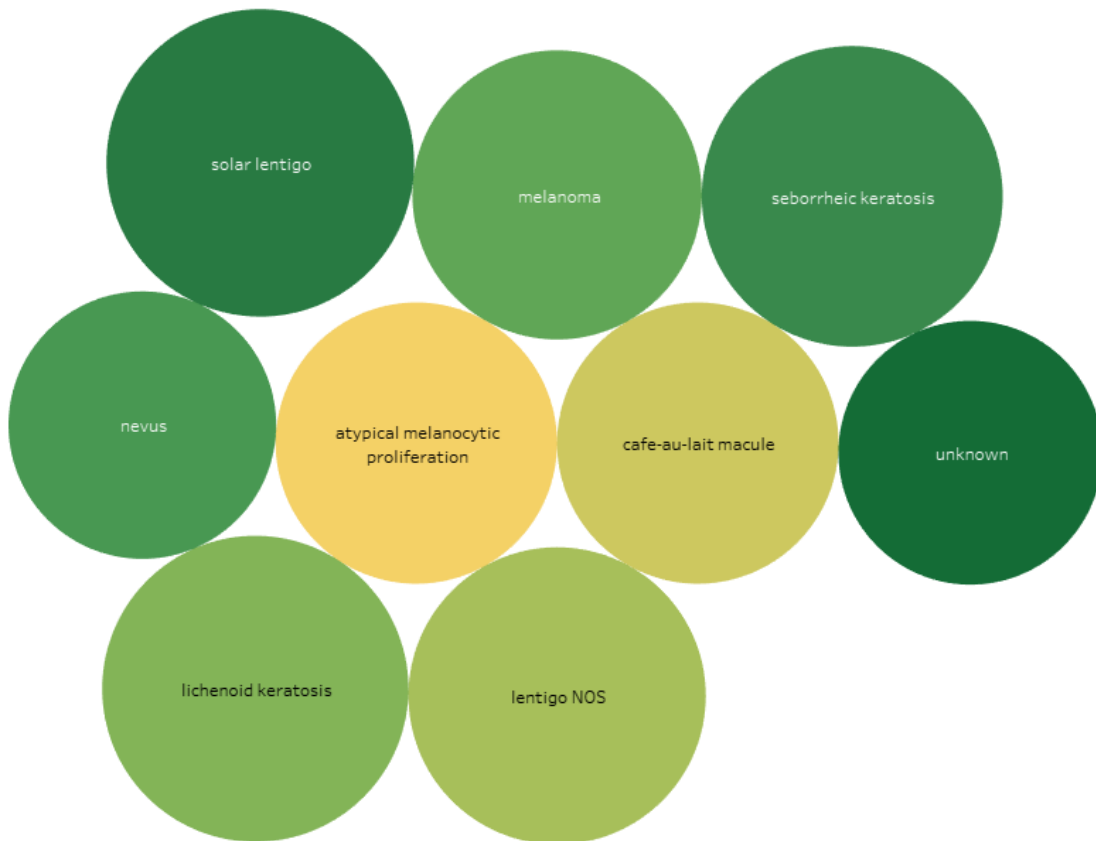
Fuente: Elaboración propia.

- Diagnóstico y edad: En este caso vemos los distintos tipos de cáncer evaluados y la edad promedio de las personas que lo padecen. Aquí se ve como los desconocidos

tienen una edad promedio de 48.38. Por otro lado, los nevos displásicos tienen un 49.75 y el melanoma con 58.13.

**Figura 20** Diagnostico y edad

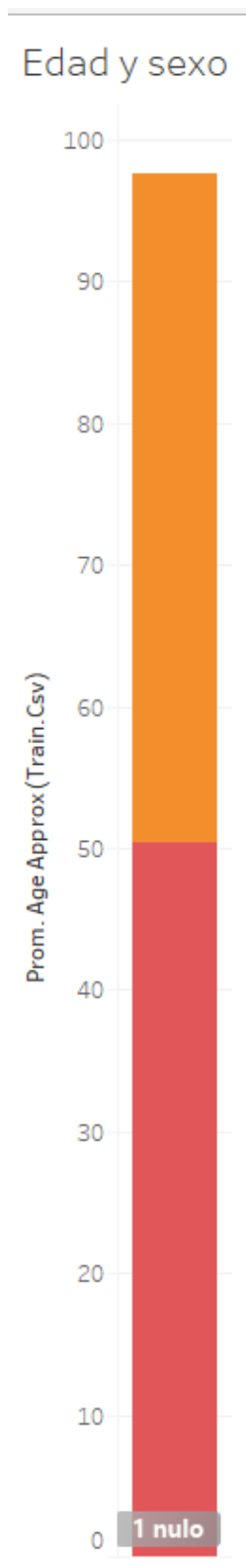
Diagnostico y edad



Fuente: Elaboración propia.

- Sexo y edad: Aquí la población analizada según el sexo es de 47.237 para las mujeres y de 50.398 para los hombres.

**Figura 21** Edad y sexo

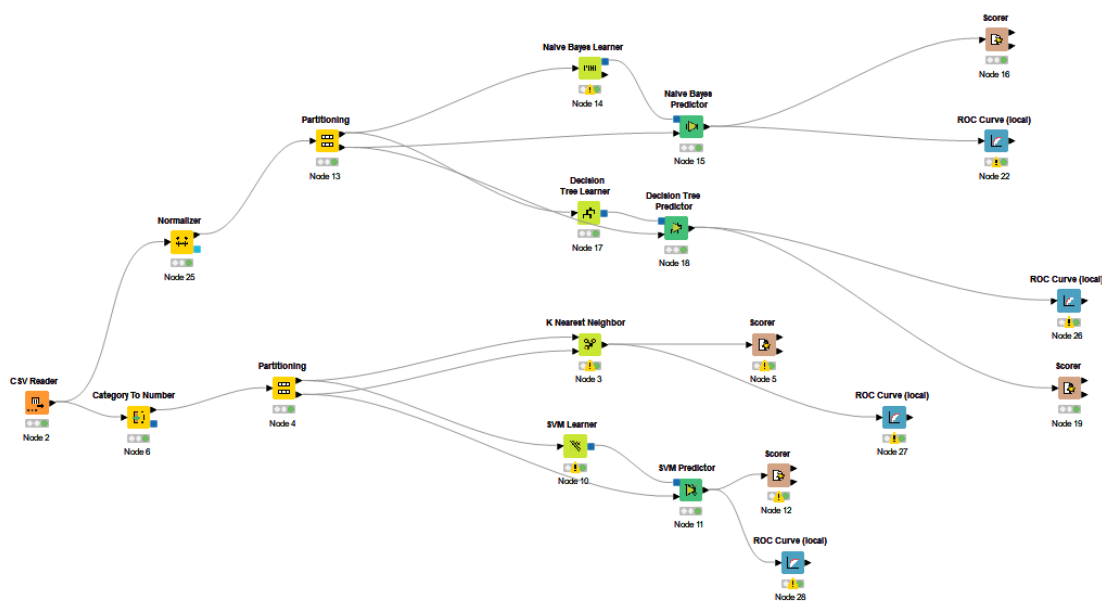


Fuente: Elaboración propia.

## 4.5 Clasificación y clústeres en Knime

Posterior a ello analizamos la data mediante diversos algoritmos de clasificación y clústeres con el fin de tener una evaluación completa de la data, entendiendo que aún necesita de un filtro previo.

*Figura 22 Esquema general en Knime*



Fuente: Elaboración propia.

Dentro de la primera evaluación a la data pudimos ver como mediante una normalización hemos obtenido los resultados en los algoritmos en naive bayes y árbol de decisión. Siendo los resultados los siguientes:

**Figura 23** Matriz de confusión de Naive Bayes

|                                  |         |          |       |                |             |                |               |                |               |
|----------------------------------|---------|----------|-------|----------------|-------------|----------------|---------------|----------------|---------------|
| Confusion Matrix - 3:16 - Scorer |         |          |       |                |             |                |               |                |               |
| File                             | Hilite  |          |       |                |             |                |               |                |               |
| diagnosis \...                   | unknown | melanoma | nevus | seborrheic ... | lentigo NOS | lichenoid k... | solar lentigo | cafe-au-lai... | atypical m... |
| unknown                          | 8110    | 0        | 0     | 0              | 0           | 0              | 0             | 0              | 0             |
| melanoma                         | 0       | 169      | 0     | 0              | 0           | 0              | 0             | 0              | 0             |
| nevus                            | 1594    | 0        | 0     | 0              | 0           | 0              | 0             | 0              | 0             |
| seborrheic k...                  | 40      | 0        | 0     | 0              | 0           | 0              | 0             | 0              | 0             |
| lentigo NOS                      | 15      | 0        | 0     | 0              | 0           | 0              | 0             | 0              | 0             |
| lichenoid ker...                 | 7       | 0        | 0     | 0              | 0           | 0              | 0             | 0              | 0             |
| solar lentigo                    | 2       | 0        | 0     | 0              | 0           | 0              | 0             | 0              | 0             |
| cafe-au-lait ...                 | 0       | 0        | 0     | 0              | 0           | 0              | 0             | 0              | 0             |
| atypical mel...                  | 1       | 0        | 0     | 0              | 0           | 0              | 0             | 0              | 0             |

Correct classified: 8,279

Accuracy: 83.307%

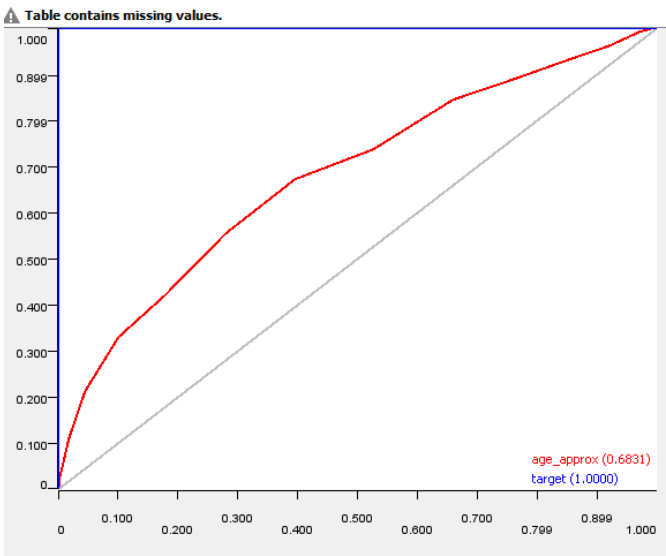
Cohen's kappa (κ): 0.155%

Wrong classified: 1,659

Error: 16.693%

Fuente: Elaboración propia.

**Figura 24** Curva de ROC de Naive Bayes



Fuente: Elaboración propia.

**Figura 25** Matriz de confusión de Árbol de decisión

Confusion Matrix - 3:19 - Scorer

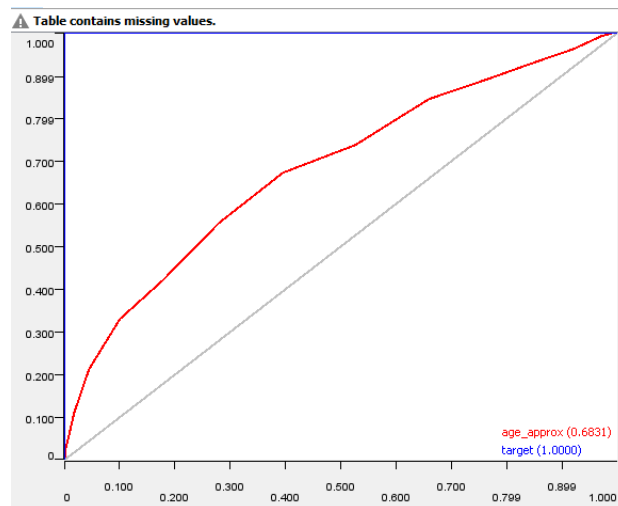
| diagnosis \...   | unknown | nevus | melanoma | seborrheic ... | lentigo NOS | lichenoid k... | solar lentigo | cafe-au-lai... | atypical m... |
|------------------|---------|-------|----------|----------------|-------------|----------------|---------------|----------------|---------------|
| unknown          | 8099    | 11    | 0        | 0              | 0           | 0              | 0             | 0              | 0             |
| nevus            | 1584    | 10    | 0        | 0              | 0           | 0              | 0             | 0              | 0             |
| melanoma         | 0       | 0     | 169      | 0              | 0           | 0              | 0             | 0              | 0             |
| seborrheic k...  | 39      | 1     | 0        | 0              | 0           | 0              | 0             | 0              | 0             |
| lentigo NOS      | 15      | 0     | 0        | 0              | 0           | 0              | 0             | 0              | 0             |
| lichenoid ker... | 7       | 0     | 0        | 0              | 0           | 0              | 0             | 0              | 0             |
| solar lentigo    | 2       | 0     | 0        | 0              | 0           | 0              | 0             | 0              | 0             |
| cafe-au-lait ... | 0       | 0     | 0        | 0              | 0           | 0              | 0             | 0              | 0             |
| atypical mel...  | 1       | 0     | 0        | 0              | 0           | 0              | 0             | 0              | 0             |

Correct classified: 8,278  
Accuracy: 83.296%  
Cohen's kappa ( $\kappa$ ): 0.161%

Wrong classified: 1,660  
Error: 16.704%

Fuente: Elaboración propia.

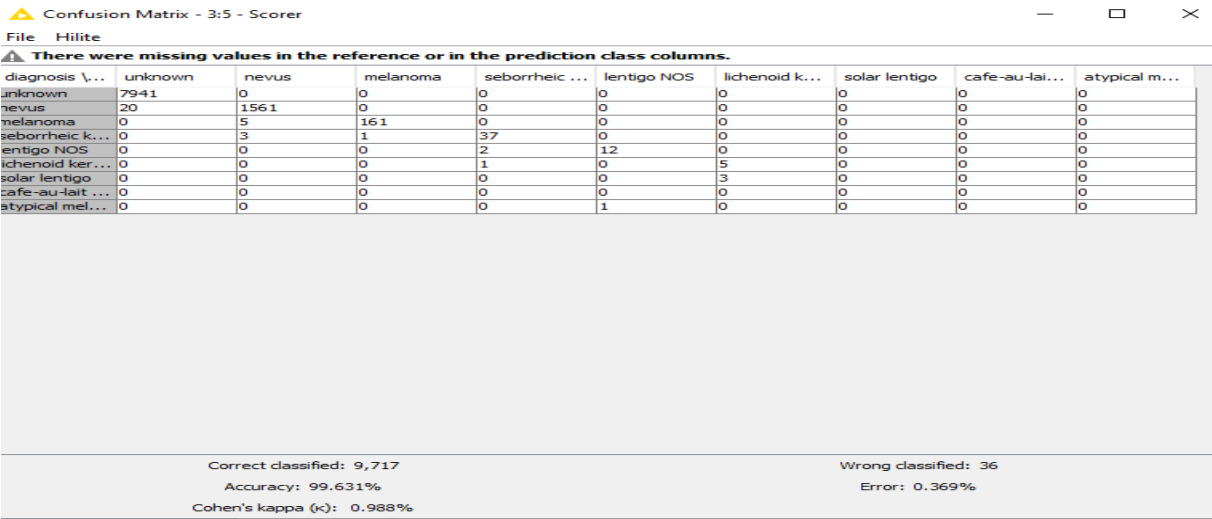
**Figura 26** Curva de ROC de Árbol de decisión



Fuente: Elaboración propia.

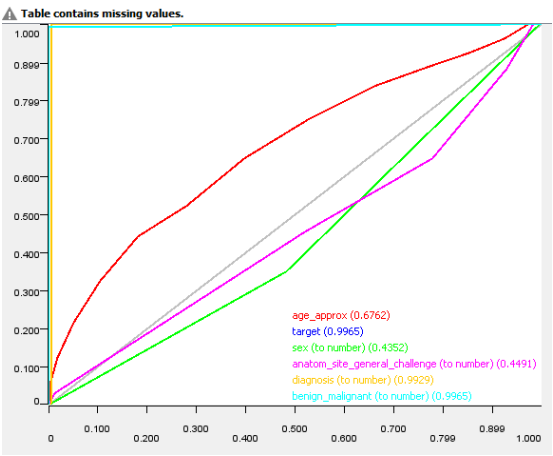
En el caso de los algoritmos de KNN y SVMLearn se hizo una conversión previa de las variables a números con el fin de evaluar correctamente los clústeres. Posterior a ello si hizo la partición, predicción, análisis y resultados que se verán a continuación:

**Figura 27** Matriz de confusión de KNN



Fuente: Elaboración propia.

**Figura 28** Curva de ROC de KNN



Fuente: Elaboración propia.

**Figura 29** Matriz de confusión de SVM Learner

Confusion Matrix - 3:12 - Scorer

File Hilite

There were missing values in the reference or in the prediction class columns.

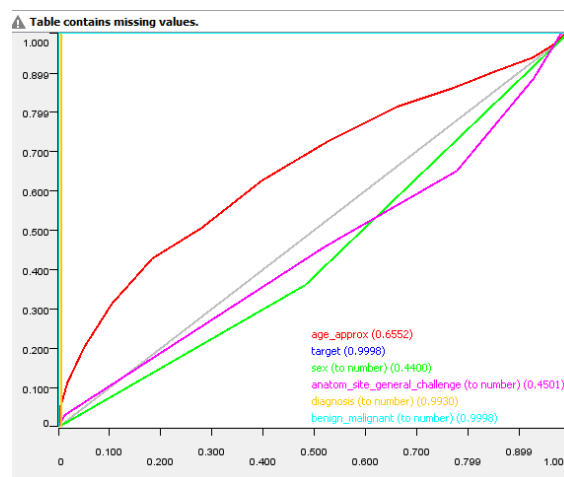
| diagnosis \...   | unknown | nevus | melanoma | seborrheic ... | lentigo NOS | lichenoid k... | solar lentigo | cafe-au-lai... | atypical m... |
|------------------|---------|-------|----------|----------------|-------------|----------------|---------------|----------------|---------------|
| unknown          | 7941    | 0     | 0        | 0              | 0           | 0              | 0             | 0              | 0             |
| nevus            | 0       | 1581  | 0        | 0              | 0           | 0              | 0             | 0              | 0             |
| melanoma         | 0       | 0     | 166      | 0              | 0           | 0              | 0             | 0              | 0             |
| seborrheic k...  | 0       | 41    | 0        | 0              | 0           | 0              | 0             | 0              | 0             |
| lentigo NOS      | 0       | 14    | 0        | 0              | 0           | 0              | 0             | 0              | 0             |
| lichenoid ker... | 0       | 6     | 0        | 0              | 0           | 0              | 0             | 0              | 0             |
| solar lentigo    | 0       | 3     | 0        | 0              | 0           | 0              | 0             | 0              | 0             |
| cafe-au-lait ... | 0       | 0     | 0        | 0              | 0           | 0              | 0             | 0              | 0             |
| atypical mel...  | 0       | 1     | 0        | 0              | 0           | 0              | 0             | 0              | 0             |

Correct classified: 9,688  
Accuracy: 99.334%  
Cohen's kappa ( $\kappa$ ): 0.978%

Wrong classified: 65  
Error: 0.666%

Fuente: Elaboración propia.

**Figura 30** Curva de ROC de SVM Learner



Fuente: Elaboración propia.

## 4.6 Librerías usadas de Kaggle

A simple vista, el código se organiza en cinco grandes bloques separados por comentarios que estructuran sus funcionalidades de manera clara y lógica:

1. **Importación de Módulos Fundamentales:** El primer bloque se enfoca en las importaciones esenciales para el sistema, cubriendo aspectos generales como el manejo de archivos y directorios (`os` y `os.path`), gestión de memoria (`gc`), y el seguimiento de tiempo (`time` y `datetime`). Estos módulos son la base para la organización y optimización del procesamiento.
2. **Manejo de Imágenes y Visualización Básica:** Este bloque integra librerías para el preprocesamiento y visualización de datos, tales como `cv2` (OpenCV) y `PIL` (Python Imaging Library) para manipulación de imágenes y conversión entre espacios de color, útil en el análisis visual. Además, emplea `pydicom` para trabajar con imágenes en formato DICOM, que son frecuentes en el ámbito médico. En cuanto a visualización, se utilizan `matplotlib`, `seaborn`, y `missingno` para mostrar gráficos, matrices, e identificar valores faltantes en el conjunto de datos, facilitando la inspección visual y estadística.
3. **Evaluación Estadística con Scikit-Learn:** En este bloque, las herramientas de Scikit-Learn (`sklearn`) permiten la división de datos en pliegues, la codificación de etiquetas, y la evaluación del rendimiento del modelo mediante métricas como `accuracy_score`, `roc_auc_score`, y `confusion_matrix`. Esto garantiza un análisis estadístico riguroso y una validación cruzada efectiva en el proceso de clasificación.
4. **Desarrollo y Entrenamiento de Modelos con PyTorch:** PyTorch es la herramienta principal para construir y entrenar modelos de redes neuronales en este código. Con esta librería, se manejan los conjuntos de datos y las estructuras de red como `nn` para la definición de capas, y `DataLoader` para la carga eficiente de datos. Se utilizan modelos pre-entrenados como `ResNet50`, lo cual agiliza la construcción de modelos y mejora el rendimiento al aplicar técnicas de transferencia de aprendizaje.
5. **Preprocesamiento Avanzado de Imágenes:** En el último bloque, se usan técnicas de aumentación y preprocesamiento de imágenes mediante `albumentations`, con funciones como normalización de color, ajuste de brillo, y aplicación de filtros. Estas técnicas optimizan la calidad de las imágenes y minimizan elementos distractores, como la vellosoidad en imágenes médicas, lo que es crucial para mejorar la precisión de diagnóstico. Además, se incluye la configuración para ignorar

advertencias mediante `warnings.filterwarnings("ignore")`, asegurando una ejecución limpia y continua.

Este enfoque modular y metódico permite que el código maneje de manera eficiente un flujo de trabajo complejo, desde la carga y procesamiento de datos hasta la visualización y modelado, optimizando el rendimiento en cada etapa.

```
pip install efficientnet-pytorch
pip install torchvision
pip install pydicom
pip install torch torchvision torchaudio
pip install -U albumentations
# -----
# Importación de módulos para manejo de archivos y directorios
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
import os.path # Para operaciones relacionadas con rutas de archivos
import gc # Para la recolección de basura y manejo de memoria
import time # Manejo de tiempo
import datetime # Para trabajar con fechas y horas
# -----
# Librerías para manipulación de imágenes y espacios de color
import cv2 # OpenCV para procesamiento de imágenes
from PIL import Image # Convertir entre espacios de color, de RGB a YCbCr
# -----
# Importación de librerías básicas para análisis de datos y visualización
import pandas as pd # Manipulación de datos en formato DataFrame
import numpy as np # Operaciones numéricas y manejo de matrices
import seaborn as sns # Visualización avanzada de datos
import missingno as msno # Para la visualización de datos faltantes
import matplotlib
import matplotlib.pyplot as plt # Visualización de gráficos y datos
import matplotlib.image as mpimg # Comprobación de imágenes
visualmente
import random # Generación aleatoria
# Barra de progreso para facilitar el seguimiento en procesos largos
from tqdm.notebook import tqdm
# -----
# Librerías para procesamiento de imágenes en formatos especializados
import pydicom # Trabajar con imágenes médicas en formato DICOM
```

```

from skimage.transform import resize # Redimensionar imágenes
# -----
# Módulos de Scikit-Learn para preprocesamiento y evaluación de
modelos
from sklearn.model_selection import StratifiedKFold, GroupKFold #
División de datos en pliegues
from sklearn.metrics import accuracy_score, roc_auc_score,
confusion_matrix # Métricas de evaluación
from sklearn.preprocessing import LabelEncoder, OneHotEncoder,
StandardScaler # Codificación de etiquetas
# -----
# Importación de PyTorch para modelado de redes neuronales
import torch
import torchvision.transforms as transforms # Para las
transformaciones
import torchvision
import torch.nn as nn # Definición de arquitecturas de redes
neuronales
import torch.nn.functional as F # Funciones de activación y pérdida
import torchvision.models as models # Modelos para las redes
neuronales
import torch.optim as optim # Para ver la optimización del modelo
from torchvision import models, transforms # Modelos para el dataset y
transformaciones
from torch import FloatTensor, LongTensor # Tipos de tensores
específicos
from torch.utils.data import Dataset, DataLoader, Subset,
TensorDataset # Creación y manejo de conjuntos de datos
from torch.optim.lr_scheduler import ReduceLROnPlateau # Reducción de
tasa de aprendizaje
from sklearn.model_selection import KFold # Agregado para importar
KFold
import gc
# -----
# Albumentations: Librería avanzada para aumentación de imágenes
from albumentations import (
    ToFloat, Normalize, VerticalFlip, HorizontalFlip, Compose, Resize,
    RandomBrightnessContrast, HueSaturationValue, Blur, GaussNoise,
    Rotate,
    RandomResizedCrop, ShiftScaleRotate
)
from albumentations.pytorch import ToTensorV2 # Conversión a tensores
de PyTorch
# -----
# Modelos de redes neuronales

```

```

from efficientnet_pytorch import EfficientNet # Importación del
modelo EfficientNet
from torchvision.models import resnet34, resnet50, efficientnet_b0 #
Modelos ResNet pre-entrenados
# -----
# Configuración para ignorar advertencias no críticas
import warnings
warnings.filterwarnings("ignore") # Ignorar advertencias para una
ejecución limpia

```

#### 4.7 Análisis de los espacios vacíos y valores perdidos

Se analizaron los conjuntos de datos de entrenamiento y prueba para identificar las variables con valores faltantes. Utilizando gráficos de matrices de valores ausentes, se observó que las principales variables afectadas fueron "sex", "age\_approx" y "anatom\_site\_general\_challenge". En el conjunto de datos de entrenamiento ("Train"), la variable "sex" tiene 65 valores faltantes, lo que representa el 0.20% del total, "age\_approx" presenta 68 valores ausentes (0.21%), y "anatom\_site\_general\_challenge" cuenta con 527 valores faltantes (1.59%). En el conjunto de prueba ("Test"), "anatom\_site\_general\_challenge" muestra 351 valores ausentes, equivalentes al 3.20% del total. Estos resultados sugieren que la mayor proporción de datos ausentes se encuentra en la variable de anatomía en ambos conjuntos, lo cual es relevante para el análisis y procesamiento de datos futuros.

```

# Definir el directorio raíz de los datos
directory = '/kaggle/input/siim-isic-melanoma-classification'

# Cargar los datos en DataFrames
train = pd.read_csv(f'{directory}/train.csv')
test = pd.read_csv(f'{directory}/test.csv')

# Configuración de la figura con dos subplots en una fila
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(16, 6))

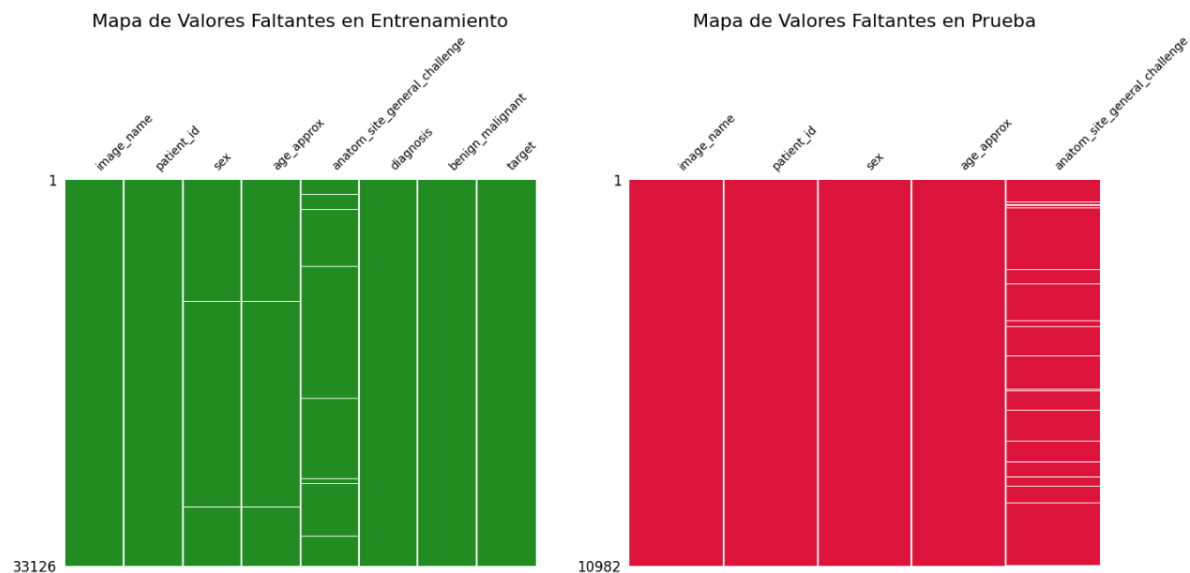
# Visualización de los valores faltantes en el conjunto de
entrenamiento (verde)
msno.matrix(train, ax=ax1, color=(34/255, 139/255, 34/255),
fontsize=10, sparkline=False) # Tono verde
ax1.set_title('Mapa de Valores Faltantes en Entrenamiento',
fontsize=16)

```

```
# Visualización de los valores faltantes en el conjunto de prueba
(rojo)
msno.matrix(test, ax=ax2, color=(220/255, 20/255, 60/255),
fontSize=10, sparkline=False) # Tono rojo
ax2.set_title('Mapa de Valores Faltantes en Prueba', fontsize=16)

# Mostrar el gráfico
plt.show()
```

**Figura 31** Mapa de valores perdidos



Fuente: Elaboración propia.

```
# Calcular valores faltantes en cada columna del conjunto de
entrenamiento
missing_train = train.isnull().sum()
total_train = len(train)
missing_train_percentage = (missing_train / total_train) * 100

# Calcular valores faltantes en cada columna del conjunto de prueba
missing_test = test.isnull().sum()
total_test = len(test)
missing_test_percentage = (missing_test / total_test) * 100

# Filtrar solo las columnas con valores faltantes
missing_train_summary = missing_train[missing_train > 0]
```

```

missing_train_percentage_summary =
missing_train_percentage[missing_train > 0]

missing_test_summary = missing_test[missing_test > 0]
missing_test_percentage_summary = missing_test_percentage[missing_test
> 0]

# Mostrar el resumen de valores faltantes en el conjunto de
entrenamiento
print("Dataset 'Train':")
for column, missing, percent in zip(missing_train_summary.index,
missing_train_summary, missing_train_percentage_summary):
    print(f"• {column}: {missing} valores perdidos ({percent:.2f}% del
total)")

# Mostrar el resumen de valores faltantes en el conjunto de prueba
print("\nDataset 'Test':")
for column, missing, percent in zip(missing_test_summary.index,
missing_test_summary, missing_test_percentage_summary):
    print(f"• {column}: {missing} valores perdidos ({percent:.2f}% del
total)")

```

Dataset 'Train':

- sex: 65 valores perdidos (0.20% del total)
- age\_approx: 68 valores perdidos (0.21% del total)
- anatom\_site\_general\_challenge: 527 valores perdidos (1.59% del total)

Dataset 'Test':

- anatom\_site\_general\_challenge: 351 valores perdidos (3.20% del total)

#### 4.8 Análisis de la variable objetivo

En el análisis de la variable objetivo que define el diagnóstico final, se realizó un tratamiento exhaustivo de los datos, comenzando con una visualización que permite observar la distribución de los casos benignos y malignos en el conjunto de datos. Se generó un gráfico de barras donde se muestra la frecuencia de la variable objetivo, revelando que la mayoría de los casos corresponden a diagnósticos benignos (32,542 casos), mientras que los casos malignos son considerablemente menos frecuentes (584 casos).

Además, se realizó un análisis de la distribución de edades para los diferentes tipos de diagnóstico. Para ello, se utilizó un gráfico de densidad (KDE) que compara las distribuciones de edad de los casos benignos y malignos. En este gráfico, la línea más gruesa, correspondiente a los casos benignos, muestra una tendencia más pronunciada en el rango de edades entre 40 y 60 años, mientras que la línea más delgada, correspondiente a los casos malignos, muestra una tendencia en el rango de 60 a 80 años. Estos resultados sugieren que la mayor parte de los diagnósticos benignos se concentran en un grupo de edad relativamente más joven, mientras que los casos malignos tienden a ser más frecuentes en pacientes de mayor edad.

El análisis gráfico realizado resalta la importancia de abordar con precaución los casos benignos, que predominan en los datos, y de considerar la distribución de edad como un factor clave para la evaluación de diagnósticos en proyectos relacionados con la detección temprana de cáncer de piel.

```
# Reemplazar valores infinitos con NaN en el DataFrame 'train'
train.replace([np.inf, -np.inf], np.nan, inplace=True)

# Definir una paleta de colores en tonos morados
colors_purple = ["#7B68EE", "#6A5ACD", "#8A2BE2", "#9370DB"]

# Crear la figura y los ejes
f, (ax1, ax2) = plt.subplots(1, 2, figsize=(16, 6))

# Gráfico de barras para la variable objetivo
a = sns.countplot(data=train, x='benign_malignant',
palette=colors_purple[2:4], ax=ax1)

# Gráficos de distribución de edad (modificado para usar 'age_approx')
sns.kdeplot(data=train[train['target'] == 0], x='age_approx', ax=ax2,
color=colors_purple[2],
linewidth=4, label='Benigno')
sns.kdeplot(data=train[train['target'] == 1], x='age_approx', ax=ax2,
color=colors_purple[3],
linewidth=4, label='Maligno')

# Añadir anotaciones al gráfico de barras
for p in a.patches:
    a.annotate(format(p.get_height(), ', '),
(p.get_x() + p.get_width() / 2., p.get_height()),
```

```

        ha='center', va='center',
        xytext=(0, 4), textcoords='offset points')

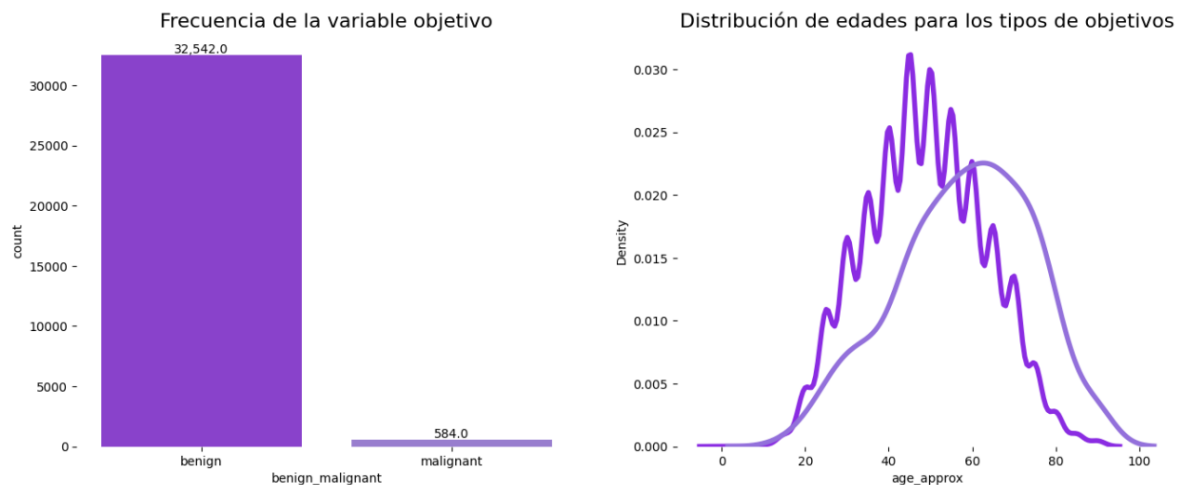
# Ajustar títulos y formato
ax1.set_title('Frecuencia de la variable objetivo', fontsize=16)
ax2.set_title('Distribución de edades para los tipos de objetivos',
fontsize=16)

# Eliminar los bordes de los gráficos
sns.despine(left=True, bottom=True)

# Mostrar la figura
plt.show()

```

**Figura 32** Frecuencia de la variable objetivo y su distribución por edad



Fuente: Elaboración propia.

## 4.8 Preprocesando los datasets

Este análisis de datos para clasificar melanoma comenzó organizando las rutas de las imágenes en formatos DICOM y JPEG. A cada conjunto de datos (entrenamiento y prueba) se le asignaron rutas específicas, utilizando el nombre único de cada imagen para generar un camino preciso que fue incorporado en los DataFrames originales.

Luego se aplicó una codificación numérica a las variables categóricas para facilitar las siguientes etapas de procesamiento. En el conjunto de entrenamiento, se seleccionaron las columnas diagnosis, sex y anatom\_site\_general\_challenge para esta transformación,

mientras que en el conjunto de prueba solo se codificaron sex y anatom\_site\_general\_challenge. Para esto, se utilizó LabelEncoder de la biblioteca sklearn.preprocessing, lo que permitió convertir las categorías textuales a un formato numérico uniforme en cada columna.

Además, en el DataFrame de entrenamiento se verificó si la columna benign\_malignant estaba presente y, de ser así, se eliminó para evitar duplicación en la información. Finalmente, los DataFrames transformados de entrenamiento y prueba se almacenaron en archivos CSV con los nombres train\_clean.csv y test\_clean.csv, con el propósito de tener conjuntos de datos organizados para la fase de modelado.

Este proceso, que incluyó la organización de rutas y la codificación de variables categóricas, crea una base de datos optimizada y preparada para las etapas de modelado y predicción en el análisis de imágenes de melanoma.

```
# === Definir rutas para archivos DICOM y JPEG utilizando 'image_name' ===  
  
# Crear y añadir rutas DICOM al DataFrame de entrenamiento  
train['path_dicom'] = directory + '/train/' + train['image_name'] +  
'.dcm'  
  
# Crear y añadir rutas DICOM al DataFrame de prueba  
test['path_dicom'] = directory + '/test/' + test['image_name'] +  
'.dcm'  
  
# Crear y añadir rutas JPEG al DataFrame de entrenamiento  
train['path_jpeg'] = directory + '/jpeg/train/' + train['image_name']  
+ '.jpg'  
  
# Crear y añadir rutas JPEG al DataFrame de prueba  
test['path_jpeg'] = directory + '/jpeg/test/' + test['image_name'] +  
'.jpg'  
  
# === Codificación de variables categóricas en TRAIN ===  
# Definir las columnas categóricas a codificar  
categorical_columns = ['sex', 'anatom_site_general_challenge',  
'diagnosis']  
label_encoder = LabelEncoder()  
  
# Codificar cada columna categórica y asignar los valores transformados  
for column in categorical_columns:  
    train[column] = label_encoder.fit_transform(train[column])
```

```

# Eliminar la columna 'benign_malignant' si está presente en el
DataFrame
if 'benign_malignant' in train.columns:
    train.drop(['benign_malignant'], axis=1, inplace=True)

# === Codificación de variables categóricas en TEST ===
# Definir columnas categóricas en el conjunto de prueba
test_categorical_columns = ['sex', 'anatom_site_general_challenge']
for column in test_categorical_columns:
    test[column] = label_encoder.fit_transform(test[column])

# Guardar los DataFrames limpios
train.to_csv('train_clean.csv', index=False)
test.to_csv('test_clean.csv', index=False)

```

#### 4.8 Formatos de las imágenes a tratar

El análisis comienza evaluando la cantidad de imágenes disponibles en los conjuntos de datos, diferenciadas en formatos DICOM y JPEG para los conjuntos de entrenamiento y prueba. Según los resultados, hay 33,126 imágenes disponibles en ambos formatos para el conjunto de entrenamiento, mientras que el conjunto de prueba contiene 10,982 imágenes en cada formato. Este paso resulta crucial para garantizar que los datos estén completos y organizados antes de proceder con el análisis y el procesamiento.

Posteriormente, se considera trabajar con una muestra de 100 imágenes para examinar sus características y distribuciones. Este procedimiento busca identificar las particularidades de las imágenes y la razón de la selección de los formatos DICOM y JPEG, dado su uso frecuente en contextos médicos debido a su calidad y compatibilidad con herramientas de análisis.

El enfoque también incluye la preparación de las imágenes mediante la estandarización de su tamaño, un paso necesario para facilitar su análisis. Además, se contempla la visualización de muestras representativas de casos benignos y malignos, seleccionando 10 imágenes de cada categoría, lo cual permite observar las diferencias visuales entre ambas clases y preparar los datos para los procesos de modelado y clasificación.

```

# === Definir rutas de los directorios ===
# Especificar las ubicaciones donde se encuentran los archivos de
imágenes DICOM y JPEG
train_dcm_dir = '../input/siim-isic-melanoma-classification/train' #
Carpeta de entrenamiento (DICOM)
test_dcm_dir = '../input/siim-isic-melanoma-classification/test' #
Carpeta de prueba (DICOM)
train_jpeg_dir = '../input/siim-isic-melanoma-
classification/jpeg/train' # Carpeta de entrenamiento (JPEG)
test_jpeg_dir = '../input/siim-isic-melanoma-
classification/jpeg/test' # Carpeta de prueba (JPEG)

# === Contar archivos en cada directorio ===
# Usar la función os.listdir para calcular el número de archivos en
cada carpeta
# Los resultados serán mostrados junto con un mensaje descriptivo
print('Total de imágenes en formato DICOM (entrenamiento):',
len(os.listdir(train_dcm_dir)), '\n' +
      'Total de imágenes en formato DICOM (prueba):',
len(os.listdir(test_dcm_dir)), '\n' +
      'Total de imágenes en formato JPEG (entrenamiento):',
len(os.listdir(train_jpeg_dir)), '\n' +
      'Total de imágenes en formato JPEG (prueba):',
len(os.listdir(test_jpeg_dir)), '\n' +
      '-----', '\n' +
      'Este resumen muestra la cantidad de archivos disponibles en los
conjuntos de datos para entrenamiento y prueba.')

```

Numero de imagenes de Train .dcm: 33126

Numero de imagenes de Test .dcm: 10982

Numero de imagenes de Train .jpeg: 33126

Numero de imagenes de Test .jpeg: 10982

-----

Este es el numero de imagenes en los datasets train/ test .csv

El análisis se centró en investigar las dimensiones de una muestra de 100 imágenes seleccionadas del conjunto de entrenamiento. En primer lugar, se creó una lista para registrar las dimensiones de cada imagen, expresadas como ancho y alto. Cada imagen fue procesada individualmente a partir de su ruta específica, y sus dimensiones fueron almacenadas en un DataFrame que facilitó la organización y el análisis posterior. Para la

visualización, las dimensiones se formatearon en un estilo uniforme, presentándolas como pares ordenados del tipo [ancho, alto].

Se utilizó una paleta de colores en tonos de naranja para ilustrar las frecuencias de cada tamaño único en las imágenes. El gráfico de barras permite observar de manera clara que la mayoría de las imágenes en la muestra poseen un tamaño de [6000, 4000], seguido por tamaños menores en frecuencias decrecientes. A su vez, las barras del gráfico incluyen anotaciones que indican el número exacto de imágenes correspondientes a cada categoría de tamaño.

Este análisis resulta fundamental para comprender la diversidad de resoluciones dentro del conjunto de datos y establecer estándares de preprocesamiento, como la normalización de tamaños, lo cual optimiza el rendimiento en las etapas posteriores de modelado y análisis.

```
# === Definición de la paleta de colores ===
# Crear una lista de colores en tonos de naranja para usar en la
visualización
colors_orange = ["#FFA07A", "#FF8C00", "#FF4500", "#FFD700"]

# === Inicialización de la lista para dimensiones ===
# Esta lista almacenará las dimensiones (ancho y alto) de las imágenes
en el conjunto de entrenamiento
shapes_train = []

# === Lectura de imágenes y recolección de tamaños ===
# Iterar a través de las rutas de las imágenes JPEG en el DataFrame de
entrenamiento
for k, path in enumerate(train['path_jpeg']):
    image = Image.open(path) # Abrir la imagen desde la ruta
    shapes_train.append(image.size) # Extraer el tamaño (ancho, alto)
    y agregarlo a la lista

    if k >= 100: # Limitar el bucle a las primeras 101 imágenes para
eficiencia
        break

# === Creación de DataFrame para tamaños ===
# Convertir la lista de tamaños a un DataFrame para facilitar el
análisis y la visualización
```

```

shapes_train = pd.DataFrame(data=shapes_train, columns=['H', 'W'],
dtype='object')

# Crear una nueva columna con el tamaño formateado como una cadena en
estilo [H, W]
shapes_train['Size'] = '[' + shapes_train['H'].astype(str) + ', ' +
shapes_train['W'].astype(str) + ']'

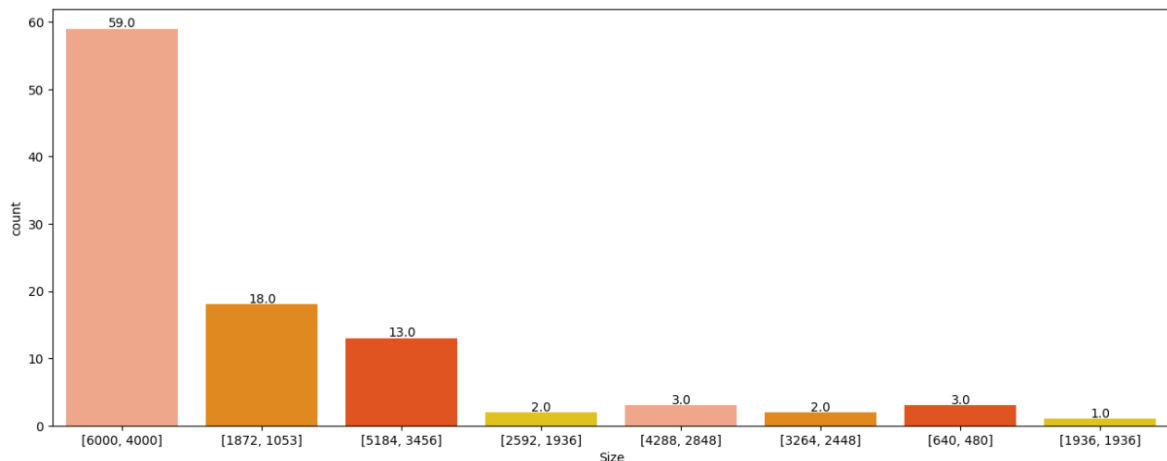
# === Configuración de la visualización ===
# Definir el tamaño de la figura para el gráfico
plt.figure(figsize=(16, 6))

# === Creación del gráfico de barras ===
# Visualizar las frecuencias de cada tamaño único de las imágenes
usando una paleta en tonos naranja
a = sns.countplot(x=shapes_train['Size'], palette=colors_orange)

# === Anotación de las barras ===
# Añadir valores sobre cada barra para indicar la cantidad
correspondiente de imágenes
for p in a.patches:
    a.annotate(format(p.get_height(), ','), # Obtener la altura de la
barra como texto
                (p.get_x() + p.get_width() / 2., p.get_height()), #
Posición del texto
                ha='center', va='center', # Alineación horizontal y
vertical
                xytext=(0, 4), textcoords='offset points') #
Desplazamiento vertical

```

**Figura 33** Tamaño de 100 imágenes



Fuente: Elaboración propia.

Para analizar visualmente las características de las imágenes DICOM dentro del conjunto de datos, se implementó una función que permite cargar y mostrar imágenes específicas a partir de un DataFrame. Esta función acepta parámetros como la cantidad de imágenes a visualizar, el número de filas y columnas en la figura, y un título descriptivo. Cada imagen es leída desde su ruta correspondiente, procesada en formato de escala de grises, y presentada en un diseño estructurado donde los ejes han sido ocultos para enfocar la atención exclusivamente en las imágenes.

En primer lugar, se generó una figura para mostrar 10 muestras de imágenes asociadas a casos benignos (donde el campo target es igual a 0). Esto permite identificar patrones visuales asociados a las características de estas lesiones, observando uniformidades o variaciones dentro de la clase. Por otro lado, se realizó una figura similar que incluye imágenes de casos malignos (target igual a 1), facilitando la comparación visual entre ambas categorías.

Este procedimiento es esencial en un análisis exploratorio de datos, ya que brinda información inicial sobre las diferencias visuales entre clases. Además, al inspeccionar las imágenes de forma individual, se pueden identificar posibles irregularidades en el conjunto de datos, como inconsistencias en la calidad de las imágenes o errores en las etiquetas asignadas. Este paso es clave para preparar los datos antes de su uso en etapas de entrenamiento de modelos predictivos.

```
# Función para mostrar imágenes DICOM desde un DataFrame
def show_images(data, n=5, rows=1, cols=5, title='Default'):
    # Configurar el tamaño de la figura según las filas y columnas
    # especificadas
    plt.figure(figsize=(16, 4))

    # Iterar sobre las primeras 'n' imágenes del DataFrame
    for k, path in enumerate(data['path_dicom'][:n]):
        # Leer la imagen DICOM desde la ruta proporcionada
        image = pydicom.dcmread(path).pixel_array

        # Configurar el título general de la figura
        plt.suptitle(title, fontsize=16)
```

```

        # Crear subplots para cada imagen
        plt.subplot(rows, cols, k + 1)
        plt.imshow(image, cmap='gray') # Mostrar imagen en escala de
grises
        plt.axis('off') # Ocultar los ejes para resaltar las imágenes

```

```

# Llamar a la función para mostrar muestras benignas
# Filtrar imágenes con 'target' igual a 0
show_images(
    train[train['target'] == 0], # Filtrar imágenes benignas
    n=10,                        # Mostrar 10 imágenes
    rows=2,                     # Número de filas en la figura
    cols=5,                     # Número de columnas en la figura
    title='Muestras Benignas'   # Título de la figura
)

```

**Figura 34 Muestras benignas**



Fuente: Elaboración propia.

```

# Llamar a la función para mostrar muestras malignas
# Filtrar imágenes con 'target' igual a 1
show_images(
    train[train['target'] == 1], # Filtrar imágenes benignas
    n=10,                        # Mostrar 10 imágenes
    rows=2,                     # Número de filas en la figura
    cols=5,                     # Número de columnas en la figura
    title='Muestras Malignas'   # Título de la figura
)

```

**Figura 35 Muestras malignas**



Fuente: Elaboración propia.

#### 4.9 Técnicas de color

Por medio del lenguaje de programación Python, ejecutado en la plataforma de Kaggle, se implementaron técnicas de procesamiento de imágenes para analizar características visuales distintivas de la piel, las zonas afectadas y la presencia de vellosidad. En un primer caso, se utilizó una escala de grises, lo que permitió observar con claridad las texturas y diferencias de tonalidades en las imágenes, siendo una herramienta efectiva para identificar detalles básicos. Por otro lado, al aplicar un desenfoque gaussiano, se logró un efecto de realce mediante el incremento de la saturación y el contraste, lo que hizo que las características más relevantes, como las áreas afectadas, fueran significativamente más visibles, demostrando una mayor capacidad para destacar zonas de interés. Este análisis comparativo evidenció cómo la manipulación de técnicas de color puede contribuir a una interpretación visual más precisa en imágenes médicas.

```
# === Creación de subplots para visualizar las imágenes en blanco y negro ===
# Configuración de una figura con subplots organizados en 2 filas y 6 columnas
fig, axes = plt.subplots(nrows=2, ncols=6, figsize=(16, 6))

# Título general de la figura
plt.suptitle("Blanco y negro", fontsize=16)

# === Iteración para procesar y mostrar imágenes ===
# Recorrer las primeras 12 imágenes del DataFrame de entrenamiento
for i in range(0, 2 * 6): # Total de imágenes a mostrar: 2 filas * 6 columnas
```

```

# Leer el archivo DICOM desde la ruta especificada en el DataFrame
data = pydicom.dcmread(train['path_dicom'][i]) # Método
actualizado para leer DICOM
image = data.pixel_array # Extraer la matriz de píxeles de la
imagen

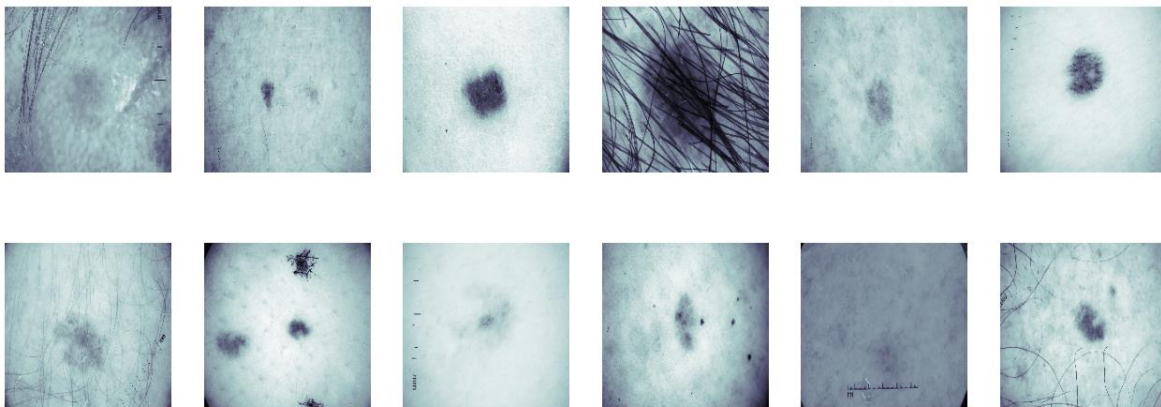
# === Transformación a blanco y negro ===
# Convertir la imagen a escala de grises utilizando OpenCV
image = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
# Redimensionar la imagen a un tamaño uniforme de 200x200 píxeles
image = cv2.resize(image, (200, 200))

# === Asignación de subplots ===
# Calcular la posición de la imagen en la cuadrícula de subplots
x = i // 6 # Determinar la fila actual (división entera por 6)
y = i % 6 # Determinar la columna actual (resto de la división
por 6)
axes[x, y].imshow(image, cmap=plt.cm.bone) # Mostrar la imagen
con el mapa de color "bone"
axes[x, y].axis('off') # Ocultar los ejes para centrarse en la
imagen

```

**Figura 36** Filtro blanco y negro

Blanco y negro



Fuente: Elaboración propia.

```

# === Creación de subplots para visualizar imágenes con desenfoque
gaussiano ===
# Configuración de una figura con subplots organizados en 2 filas y 6
columnas

```

```

fig, axes = plt.subplots(nrows=2, ncols=6, figsize=(16, 6))

# Título general de la figura
plt.suptitle("Desenfoque Gaussiano", fontsize=16)

# === Iteración para procesar y mostrar imágenes ===
# Recorrer las primeras 12 imágenes del DataFrame de entrenamiento
for i in range(0, 2 * 6): # Total de imágenes a mostrar: 2 filas * 6
    columnas
        # Leer el archivo DICOM desde la ruta especificada en el DataFrame
        data = pydicom.dcmread(train['path_dicom'][i]) # Método
actualizado para leer DICOM
        image = data.pixel_array # Extraer la matriz de píxeles de la
imagen

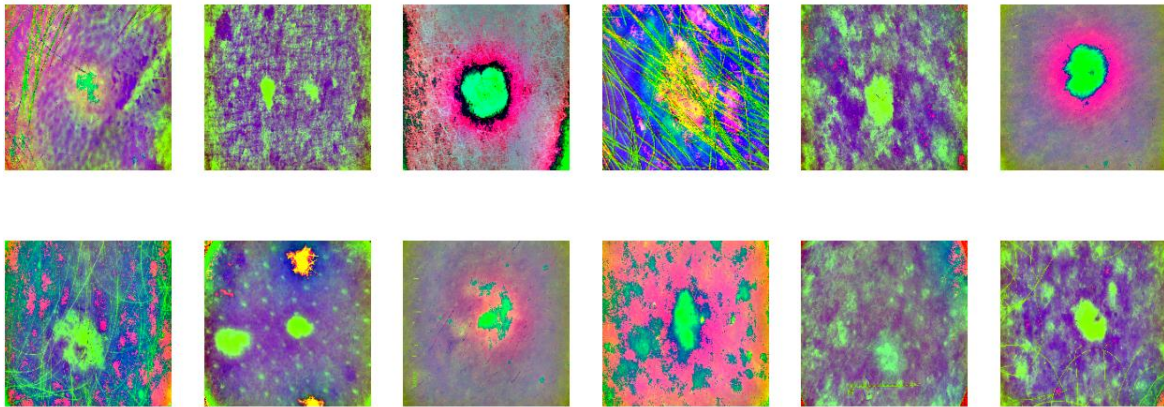
        # === Aplicación de desenfoque gaussiano ===
        # Convertir la imagen al espacio de color HSV (opcional para
realce)
        image = cv2.cvtColor(image, cv2.COLOR_RGB2HSV)
        # Redimensionar la imagen a un tamaño uniforme de 200x200 píxeles
        image = cv2.resize(image, (200, 200))
        # Aplicar desenfoque gaussiano y combinación ponderada para
realzar bordes
        image = cv2.addWeighted(image, 4, cv2.GaussianBlur(image, (0, 0),
256 / 10), -4, 128)

        # === Asignación de subplots ===
        # Calcular la posición de la imagen en la cuadrícula de subplots
        x = i // 6 # Determinar la fila actual (división entera por 6)
        y = i % 6 # Determinar la columna actual (resto de la división
por 6)
        axes[x, y].imshow(image, cmap=plt.cm.bone) # Mostrar la imagen
con el mapa de color "bone"
        axes[x, y].axis('off') # Ocultar los ejes para centrarse en la
imagen

```

**Figura 37** Filtro de desenfoque gaussiano

Desenfoque Gaussiano



Fuente: Elaboración propia.

En este proceso de visualización de imágenes, se observa cómo, utilizando PyTorch, es posible crear un objeto del dataset para manejar un conjunto de imágenes de manera eficiente. En primer lugar, se selecciona una muestra aleatoria de imágenes sin modificaciones, lo que permite observar el estado original de las mismas. A continuación, estas imágenes pasan por un proceso de transformación, donde se aplica un zoom a las mismas. Esta transformación es útil para evaluar cómo las imágenes cambian cuando se ajustan ciertos parámetros, como el tamaño o el enfoque. Posteriormente, se lleva a cabo la aplicación de una técnica específica, como la remoción de cabello, lo que implica un tratamiento adicional sobre las imágenes. Este flujo de trabajo es esencial para poder analizar la evolución de las imágenes a medida que se van modificando, y se utiliza comúnmente en el procesamiento de imágenes dentro de redes neuronales para tareas específicas.

```
# === Selección de una muestra de imágenes JPEG ===
# Seleccionamos aleatoriamente 12 imágenes de entrenamiento y
reiniciamos los índices
image_list = train.sample(12)['path_jpeg'] # Seleccionar una muestra
aleatoria de 12 rutas de imágenes
image_list = image_list.reset_index()['path_jpeg'] # Reiniciar los
índices para evitar inconsistencias

# === Visualización de la muestra original de imágenes JPEG ===
# Configuramos la figura con un tamaño de 16x6 y un título general
plt.figure(figsize=(16, 6))
plt.suptitle("Vista original", fontsize=16) # Título descriptivo para
toda la figura

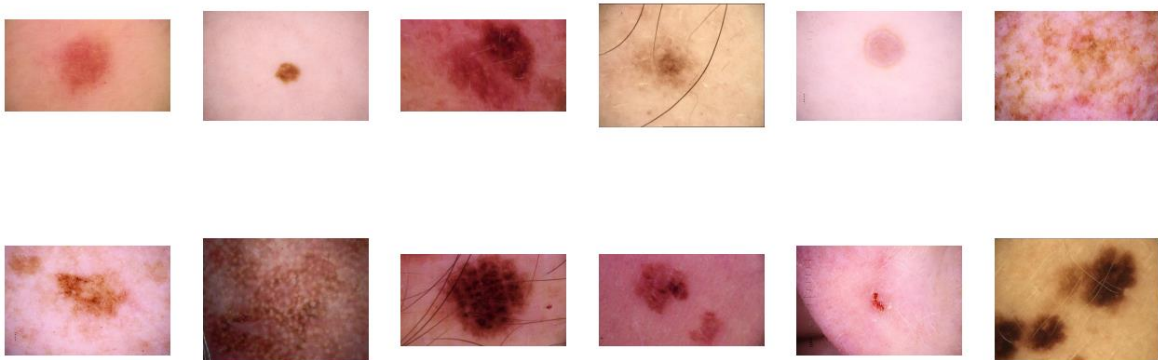
# === Iteración para mostrar las imágenes ===
```

```
# Recorremos las rutas de las imágenes seleccionadas
for k, path in enumerate(image_list):
    # Cargar la imagen desde la ruta utilizando matplotlib
    image = mpimg.imread(path) # Leer la imagen JPEG

    # Crear un subplot para cada imagen en una cuadrícula de 2 filas y
    6 columnas
    plt.subplot(2, 6, k + 1) # Determinar la posición del subplot
    correspondiente
    plt.imshow(image) # Mostrar la imagen en formato RGB
    plt.axis('off') # Ocultar los ejes para resaltar solo las
    imágenes
```

***Figura 38 Vista original de imágenes***

Vista original



Fuente: Elaboración propia.

El código presentado define una clase personalizada de Dataset en PyTorch, la cual facilita la carga y transformación de imágenes. La clase, llamada CustomDataset, toma como entrada una lista de rutas de imágenes y un conjunto de transformaciones opcionales. La función `__getitem__` permite acceder a una imagen específica del dataset, realizar las transformaciones necesarias (como la conversión a tensor), y devolver la imagen en el formato adecuado para su procesamiento en redes neuronales. Además, se define una función para mostrar las imágenes transformadas en una cuadrícula. Las transformaciones aplicadas incluyen redimensionamiento, recorte, conversión a tensores, y normalización. El proceso se complementa con un DataLoader, que permite cargar las imágenes en mini-batches, lo cual es esencial para el entrenamiento eficiente de modelos en PyTorch. En conjunto, este flujo de trabajo es útil para la preparación de datos de imágenes en tareas de aprendizaje automático, facilitando su manipulación y procesamiento previo al entrenamiento del modelo.

```

# -----
# Definición de una clase personalizada para el Dataset en PyTorch
# Esta clase nos permite cargar imágenes desde rutas y aplicar
transformaciones

class CustomDataset(Dataset):
    def __init__(self, image_paths, transform=None):
        """
        Constructor de la clase que inicializa la lista de imágenes y
        las transformaciones a aplicar.
        :param image_paths: Lista de rutas de las imágenes a cargar.
        :param transform: Transformaciones que se aplicarán a las
        imágenes.
        """
        self.image_paths = image_paths # Asignar la lista de rutas a
        las imágenes
        self.transform = transform      # Asignar las transformaciones
        si están definidas

    def __len__(self):
        """
        Retorna el número total de imágenes en el dataset.
        """
        return len(self.image_paths)

    def __getitem__(self, idx):
        """
        Obtiene la imagen en el índice especificado, la aplica a las
        transformaciones, y la devuelve.
        :param idx: Índice de la imagen que queremos cargar.
        """
        # Leer la imagen usando la ruta proporcionada
        image = plt.imread(self.image_paths[idx]) # Usar Matplotlib
        para leer la imagen

        # Convertir la imagen a formato PIL (RGB) para poder aplicar
        transformaciones
        image = Image.fromarray(image).convert('RGB')

        # Convertir la imagen en formato de arreglo NumPy (uint8)
        image = np.asarray(image).astype(np.uint8)

        # Aplicar las transformaciones si están definidas
        if self.transform is not None:
            image = self.transform(image)

```

```

        # Convertir la imagen en un tensor de PyTorch de tipo flotante
        return torch.tensor(image, dtype=torch.float)

# -----
# Función para mostrar una cuadrícula de imágenes transformadas
def display_transformed_images(images_tensor, title="Transformed
Images"):
    """
    Muestra las imágenes transformadas en una cuadrícula.
    :param images_tensor: Tensor que contiene las imágenes
transformadas.
    :param title: Título que se asignará a la figura.
    """
    # Configurar la figura para la visualización
    plt.figure(figsize=(16, 6))
    plt.suptitle(title, fontsize=16) # Agregar un título general para
la figura

    # Deshacer la normalización aplicada previamente a las imágenes
    images_tensor = images_tensor / 2 + 0.5 # Normalización inversa
    np_images = images_tensor.numpy() # Convertir tensor a formato
NumPy
    np_images = np.clip(np_images, 0., 1.) # Asegurarse de que los
valores estén entre 0 y 1

    # Mostrar las imágenes en una cuadrícula de 6 columnas
    plt.imshow(np.transpose(np_images, (1, 2, 0))) # Reorganizar los
ejes para visualización correcta
    plt.axis('off') # Eliminar los ejes para enfocar la atención en
las imágenes
    plt.show()

# -----
# Definición de las transformaciones para las imágenes
transformations = transforms.Compose([
    transforms.ToPILImage(), # Convertir imagen NumPy a
imagen PIL
    transforms.Resize((300, 300)), # Redimensionar las
imágenes a 300x300 píxeles
    transforms.CenterCrop((100, 100)), # Recortar una región
central de 100x100 píxeles
    transforms.ToTensor(), # Convertir imagen PIL a
tensor PyTorch
])

```

```

        transforms.Normalize((0.5, 0.5, 0.5),      # Normalizar las imágenes
para cada canal (R, G, B)
                                (0.5, 0.5, 0.5)),  # Usamos una media y
desviación estándar comunes
    ])

# -----
# Crear el objeto Dataset usando la lista de imágenes y las
transformaciones
custom_dataset = CustomDataset(image_paths=image_list,
transform=transformations)

# Crear un DataLoader que manejará los lotes (mini-batches) de
imágenes
dataloader = DataLoader(dataset=custom_dataset, batch_size=12,
shuffle=True)

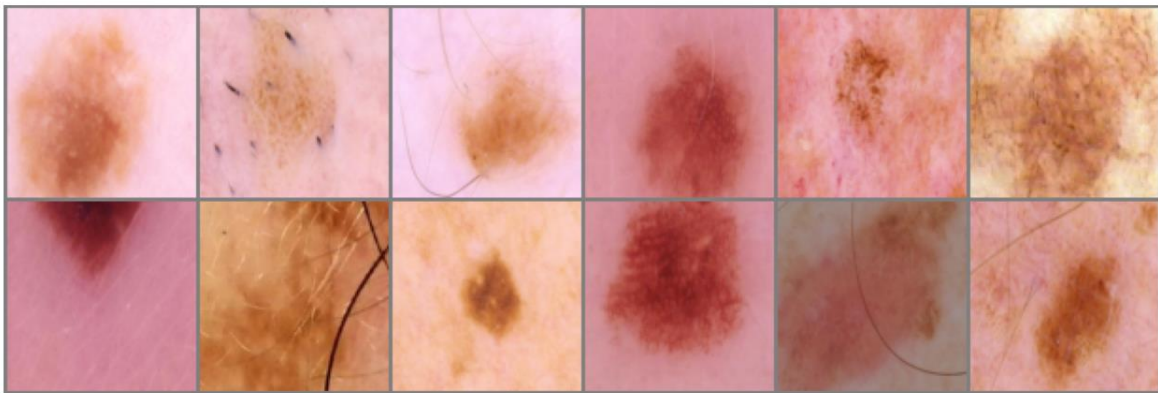
# -----
# Obtener un mini-batch de imágenes (12 imágenes)
batch_of_images = next(iter(dataloader))

# -----
# Mostrar las imágenes del mini-batch con las transformaciones
aplicadas
display_transformed_images(torchvision.utils.make_grid(batch_of_images
, nrow=6), title="Imágenes transformadas con aumento")

```

***Figura 39 Imágenes transformadas con aumento***

Imágenes transformadas con aumento



Fuente: Elaboración propia.

#### 4.10 Remoción de vello en la piel

El código presentado utiliza técnicas avanzadas de procesamiento de imágenes para eliminar el vello de las fotografías. Primero, se convierte cada imagen a escala de grises, lo que simplifica el procesamiento posterior. Posteriormente, se aplica un Kernel morfológico y una operación conocida como "Blackhat" para resaltar las áreas oscuras de la imagen, que corresponden al vello en este caso. A través de este proceso, se genera una máscara que identifica las áreas a eliminar. Finalmente, se emplea la técnica de "inpainting" para cubrir las áreas de vello, dejando la imagen con una apariencia más limpia. Este enfoque se aplica intencionadamente a un conjunto de imágenes que presentan características específicas, como las de personas con vello, para observar el contraste entre las versiones originales y las procesadas, mostrando de manera clara el cambio generado por el proceso de eliminación.

```
# -----
# Función para eliminar el vello de una imagen utilizando
# procesamiento morfológico
def remove_hair(image):
    """
    Aplica técnicas de procesamiento de imágenes para eliminar el
    vello de la imagen.
    :param image: Imagen original que contiene vello.
    :return: Imagen procesada con vello eliminado.
    """

    # Convierte la imagen a escala de grises para simplificar el
    # procesamiento
    gray_image = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)

    # Define un Kernel para la operación morfológica
    kernel = cv2.getStructuringElement(1, (17, 17)) # Elemento
    # estructural con forma rectangular

    # Aplica la operación morfológica BLACKHAT para resaltar las áreas
    # oscuras
    blackhat = cv2.morphologyEx(gray_image, cv2.MORPH_BLACKHAT,
    kernel)

    # Realiza una umbralización de la imagen procesada
    _, threshold_image = cv2.threshold(blackhat, 10, 255,
    cv2.THRESH_BINARY)
```

```

    # Utiliza el umbral generado para realizar el "inpainting" y
eliminar el vello
    processed_image = cv2.inpaint(image, threshold_image, 1,
cv2.INPAINT_TELEA)

    return processed_image

# -----
# Selección de una muestra de imágenes para procesar
# Aquí seleccionamos imágenes específicas de un conjunto de datos
basadas en un criterio
hairy_images = train[train["sex"] == 1].reset_index().iloc[[12, 14,
17, 22, 33, 34]] # Filtrar imágenes de personas con vello
image_paths = hairy_images['path_jpeg'] # Obtener las rutas de las
imágenes seleccionadas
image_paths = image_paths.reset_index()['path_jpeg'] # Reiniciar los
índices de las rutas de las imágenes

# -----
# Visualización de imágenes originales y procesadas
# Creamos una figura que será utilizada para mostrar las imágenes
originales y las procesadas
plt.figure(figsize=(16, 6)) # Definir el tamaño de la figura para una
visualización amplia
plt.suptitle("Muestras de vello original y procesado", fontsize=16) #
Título general para la visualización

# Iteramos sobre las rutas de las imágenes seleccionadas
for idx, path in enumerate(image_paths):
    # Leer la imagen original desde la ruta especificada
    original_image = mpimg.imread(path)

    # Redimensionar la imagen a un tamaño consistente (300x300
píxeles)
    original_image_resized = cv2.resize(original_image, (300, 300))

    # Aplicar la función de eliminación de vello a la imagen
    processed_image = remove_hair(original_image_resized)

    # Mostrar la imagen original en la cuadrícula de subgráficos
    plt.subplot(2, 6, idx + 1) # Ubicamos la imagen original en la
cuadrícula
    plt.imshow(original_image_resized)
    plt.axis('off') # Ocultar los ejes para una visualización más
clara

```

```
plt.title("Original") # Título para la imagen original

# Mostrar la imagen procesada (sin vello) en la cuadrícula
plt.subplot(2, 6, idx + 7) # Ubicamos la imagen procesada en la
segunda fila
plt.imshow(processed_image)
plt.axis('off') # Ocultar los ejes de la imagen procesada
plt.title("Sin Vello") # Título para la imagen procesada

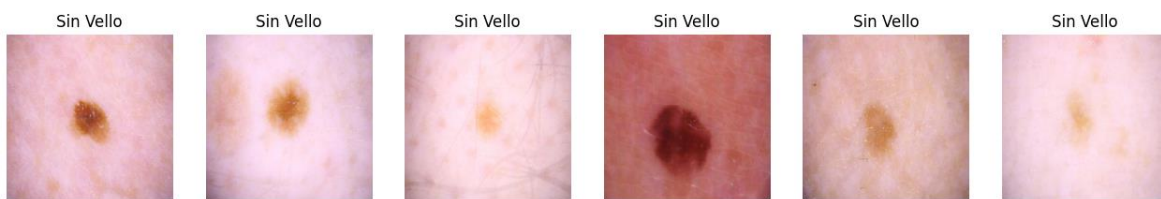
# Mostrar la figura con todas las imágenes
plt.show()
```

**Figura 40** Muestras antes de la remoción de vello



Fuente: Elaboración propia.

**Figura 41** Muestras después de la remoción de vello



Fuente: Elaboración propia.

#### 4.11 Usar el PyTorch en el Dataset

La preparación y tratamiento de los datos para la detección de melanoma sigue una serie de pasos clave para asegurar que los conjuntos de datos sean adecuados para el modelo de machine learning. En primer lugar, se filtran y mapean los valores de la columna 'diagnosis', conservando los diagnósticos relevantes para el estudio, como queratosis actínica, nevo displásico y melanoma, y excluyendo aquellos diagnósticos que no serán

útiles, como lentigo NOS, solar lentigo, y otros. Además, se crea una nueva columna, 'new\_diagnosis', que asigna un valor temporal de -1 a los diagnósticos desconocidos (código 8), lo cual facilita un análisis posterior de los datos. Las columnas numéricas de 'sex', 'age\_approx' y 'anatom\_site\_general\_challenge' se normalizan utilizando un StandardScaler, lo que asegura que las características estén dentro de un rango adecuado para el entrenamiento del modelo. Posteriormente, se preparan los datasets de entrenamiento y prueba, aplicando varias transformaciones a las imágenes del conjunto de entrenamiento, como recortes aleatorios, rotación, inversión horizontal y vertical, ajuste de saturación y brillo, con el fin de aumentar la variabilidad de los datos y mejorar la capacidad de generalización del modelo. En cuanto al conjunto de prueba, se asignan columnas vacías para los diagnósticos y las etiquetas de 'target', dejando espacio para los resultados de las predicciones. Finalmente, se verifica el tamaño de ambos conjuntos de datos para confirmar que están listos para ser utilizados en el entrenamiento y evaluación del modelo de aprendizaje automático, en el contexto de la detección temprana de melanoma.

```
# === Comentarios ===
# Los valores en la columna 'diagnosis' han sido mapeados según el
análisis:
# 8 -> unknown
# 5 -> nevus
# 4 -> melanoma
# 6 -> seborrheic keratosis
# 2 -> lentigo NOS
# 3 -> lichenoid keratosis
# 7 -> solar lentigo
# 1 -> cafe-au-lait macule
# 0 -> atypical melanocytic proliferation

# La columna 'target' indica si una muestra es benigna o maligna:
# 0 -> Benigno (para todos los valores excepto melanoma)
# 1 -> Maligno (únicamente para melanoma)

# Para esta investigación, nos enfocaremos en las siguientes
categorías:
# - Diagnósis 6 -> Queratosis actínica (seborrheic keratosis)
# - Diagnósis 3 -> Queratosis actínica (lichenoid keratosis)
# - Diagnósis 5 -> Nevo displásico (nevus)
```

```

# Los valores 2, 7, 1 y 0 no serán relevantes para el nuevo dataset.
# === Procesar los datos de entrenamiento ===

# Mapear valores de diagnóstico relevantes
relevant_diagnosis = {6: "queratosis actinica", 3: "queratosis
actinica", 5: "nevo displasico", 8: "unknown"}

# Filtrar y mapear los valores de 'diagnosis' para incluir 8 (unknown)
train = train[train['diagnosis'].isin([6, 3, 5, 8, 4])]

# Crear la nueva columna 'new_diagnosis' asignando -1 a los valores
desconocidos (8)
def assign_new_diagnosis(row):
    if row['diagnosis'] == 8: # Diagnóstico desconocido
        return -1 # Asignamos un valor temporal para análisis
posterior
        return row['diagnosis'] # Mantenemos el diagnóstico existente

train['new_diagnosis'] = train.apply(assign_new_diagnosis, axis=1)

# Normalizar las columnas numéricas relevantes
required_columns = ['sex', 'age_approx',
'anatom_site_general_challenge']
scaler = StandardScaler()

for col in required_columns:
    if col in train.columns:
        train[col] = scaler.fit_transform(train[[col]])

# === Crear el Dataset Personalizado ===
class CancerDeLaPielDataset(Dataset):
    def __init__(self, dataframe, vertical_flip=0.5,
horizontal_flip=0.5, is_train=True, is_valid=False, is_test=False):
        self.dataframe = dataframe
        self.is_train = is_train
        self.is_valid = is_valid
        self.vertical_flip = vertical_flip
        self.horizontal_flip = horizontal_flip

    # Transformaciones para entrenamiento y prueba
    if is_train or is_test:
        self.transform = Compose([
            RandomResizedCrop(height=224, width=224, scale=(0.4,
1.0)),

```

```

        ShiftScaleRotate(rotate_limit=90, scale_limit=[0.8,
1.2]),
        HorizontalFlip(p=self.horizontal_flip),
        VerticalFlip(p=self.vertical_flip),
        HueSaturationValue(sat_shift_limit=[0.7, 1.3],
hue_shift_limit=[-0.1, 0.1]),
        RandomBrightnessContrast(brightness_limit=[0.7, 1.3],
contrast_limit=[0.7, 1.3]),
        Normalize(),
        ToTensorV2()
    ])
    else:
        self.transform = Compose([Normalize(), ToTensorV2()])

def __len__(self):
    return len(self.dataframe)

def __getitem__(self, index):
    # Leer imagen y datos adicionales
    image_path = self.dataframe['path_jpeg'].iloc[index]
    image = cv2.imread(image_path)
    csv_data = np.array(self.dataframe.iloc[index][['sex',
'age_approx', 'anatom_site_general_challenge']].values,
                        dtype=np.float32)

    # Aplicar las transformaciones
    image = self.transform(image=image)['image']

    # Retornar imagen, datos y target (solo si está disponible)
    if self.is_train or self.is_valid:
        target = self.dataframe['target'].iloc[index]
        return (image, csv_data), target
    else:
        return (image, csv_data)

# Crear el conjunto de datos de entrenamiento
train_dataset = CancerDeLaPielDataset(dataframe=train,
vertical_flip=True, horizontal_flip=True, is_train=True)

# Imprimir el tamaño del conjunto de entrenamiento
print('Tamaño del conjunto de entrenamiento:', len(train_dataset))

# === Procesar el conjunto de prueba ===

# Crear columnas vacías en el dataset de prueba

```

```
test['diagnosis'] = None # Columna 'diagnosis' para futuros
diagnósticos
test['target'] = None    # Columna 'target' para determinar
benigno/maligno

# Imprimir el tamaño del conjunto de prueba
print("Tamaño del conjunto de entrenamiento:", len(test))
```

Tamaño del dataset de entrenamiento: 33073

Tamaño del dataset de prueba: 10982

#### 4.12 Redes neuronales mediante ResNet50Network

En el desarrollo de modelos de redes neuronales que integran tanto imágenes como datos tabulares, es crucial aprovechar arquitecturas preentrenadas, como ResNet50, y adaptarlas a tareas específicas. El modelo ResNet50Network es un ejemplo de esta integración, combinando la capacidad de ResNet50, un modelo preentrenado ampliamente utilizado para visión computacional, con una red neuronal completamente conectada para procesar datos tabulares. El modelo se configura con parámetros clave, como el número de columnas de los datos tabulares y el tamaño de las salidas (para clasificación binaria y diagnóstico específico).

El procesamiento de la imagen se realiza mediante el modelo ResNet50, que extrae características visuales, y estas se reducen a un vector de 2048 características al eliminar la capa final de clasificación. Paralelamente, los datos tabulares se procesan mediante una red neuronal de capas densas que incluye normalización, activación ReLU y regularización mediante dropout. Después, las características extraídas de la imagen y de los datos tabulares se combinan y se pasan a través de dos capas de salida: una para la clasificación binaria (benigna o maligna) y otra para realizar el diagnóstico específico.

El enfoque utiliza técnicas de regularización, como la normalización y el dropout, para mejorar la generalización del modelo y prevenir el sobreajuste. El modelo final es capaz de manejar datos heterogéneos, integrando información visual y tabular para realizar predicciones precisas tanto para clasificación como para diagnóstico. Esta estrategia destaca por su capacidad de ofrecer resultados robustos y bien fundamentados, adaptándose a diferentes tipos de entradas y tareas de diagnóstico médico.

```

class ResNet50Network(nn.Module):
    """
    Modelo que combina las características de una imagen (extraídas
    con ResNet50) y datos
    tabulares (provenientes de un archivo CSV) para realizar una tarea
    de clasificación.
    Además, tiene dos salidas: una para clasificación binaria (benigna
    vs maligna)
    y otra para diagnóstico específico basado en categorías.
    """

    def __init__(self, output_size_classification,
output_size_diagnosis, no_columns):
        """
        Inicializa el modelo combinando ResNet50 para el procesamiento
        de imágenes y una red neuronal
        para manejar los datos tabulares. Configura las capas para las
        dos salidas: clasificación y diagnóstico.

        Parámetros:
        - output_size_classification: Número de clases para la
        clasificación binaria (benigna vs maligna).
        - output_size_diagnosis: Número de categorías para el
        diagnóstico específico.
        - no_columns: Número de columnas en los datos tabulares.
        """
        super().__init__()
        self.no_columns = no_columns
        self.output_size_classification = output_size_classification
        self.output_size_diagnosis = output_size_diagnosis

        # Bloque para extraer características de imágenes usando
        ResNet50 (preentrenado).
        # Se utiliza la arquitectura ResNet50, pero se eliminan las
        capas de clasificación
        # para obtener solo las características extraídas de la
        imagen.
        self.features = models.resnet50(pretrained=True)
        self.features =
nn.Sequential(*list(self.features.children())[:-1]) # Eliminar la
última capa de clasificación

        # Red neuronal para manejar los datos tabulares del CSV.
        # Se toma el número de columnas especificado y se procesa a
        través de varias capas.

```

```

        self.csv = nn.Sequential(
            nn.Linear(self.no_columns, 500), # Primera capa
densamente conectada para los datos tabulares
            nn.BatchNorm1d(500), # Normalización para
mejorar el rendimiento y la estabilidad
            nn.ReLU(), # Función de activación
ReLU para introducir no linealidad
            nn.Dropout(0.2) # Dropout para prevenir
el sobreajuste
        )

        # Capa de salida para la clasificación binaria.
        # Combina las características de la imagen y los datos
tabulares para producir una salida.
        self.classification = nn.Linear(2048 + 500,
self.output_size_classification)

        # Capa de salida para el diagnóstico específico.
        # También utiliza las características combinadas de la imagen
y los datos tabulares para predecir el diagnóstico.
        self.diagnosis = nn.Linear(2048 + 500,
self.output_size_diagnosis)

    def forward(self, image, csv_data, prints=False):
        """
        Realiza el pase hacia adelante (forward pass) del modelo.
Procesa la imagen y los datos tabulares
        y luego produce las salidas para la clasificación y el
diagnóstico.

        Parámetros:
        - image: Tensor de la imagen de entrada (tamaño batch_size x 3
x 224 x 224).
        - csv_data: Tensor de los datos tabulares de entrada (tamaño
batch_size x no_columns).
        - prints: Booleano para imprimir las dimensiones de los datos
durante el pase hacia adelante (opcional).

        Retorna:
        - classification_output: Predicción de la clasificación
binaria (benigna o maligna).
        - diagnosis_output: Predicción para el diagnóstico específico.
        """
        if prints:
            print(f'Tamaño de la imagen ingresada: {image.shape}')

```

```

        print(f'Tamaño de los datos tabulares ingresados:
{csv_data.shape}')

        # Procesamiento de la imagen: se extraen las características
usando ResNet50
        image = self.features(image) # Extrae características de la
imagen
        image = image.view(image.size(0), -1) # Aplanar las
características de la imagen para que sea compatible con la siguiente
capa
        if prints:
            print(f'Dimensiones de las características de la imagen:
{image.shape}')

        # Procesamiento de los datos tabulares: se pasan a través de
la red de capas densas
        csv_data = self.csv(csv_data)
        if prints:
            print(f'Dimensiones de las características tabulares
procesadas: {csv_data.shape}')

        # Combinar las características de la imagen y los datos
tabulares
        combined_features = torch.cat((image, csv_data), dim=1) #
Concatenar las características

        # Realizar las predicciones
        classification_output =
self.classification(combined_features) # Predicción para la
clasificación binaria
        diagnosis_output = self.diagnosis(combined_features) #
Predicción para el diagnóstico específico

        return classification_output, diagnosis_output

# Crear una instancia del modelo con parámetros de ejemplo
output_size_classification = 2 # Clasificación binaria (benigna vs
maligna)
output_size_diagnosis = 4 # Número de categorías para el
diagnóstico
no_columns = 3 # Número de columnas de los datos
tabulares

# Instanciar el modelo

```

```

model_example =
ResNet50Network(output_size_classification=output_size_classification,
                 output_size_diagnosis=output_size_diagnosis,
                 no_columns=no_columns)

# Imprimir la arquitectura del modelo
print(model_example)

```

**Figura 42** Resultados de la salida con ResNet50Network

```

ResNet50Network(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
    (4): Sequential(
      (0): Bottleneck(
        (conv1): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
        (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu): ReLU(inplace=True)
        (downsample): Sequential(
          (0): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        )
      )
    )
    (1): Bottleneck(
      (conv1): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
    )
    (2): Bottleneck(
      (conv1): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
    )
  )
)

```

Fuente: Elaboración propia.

En este ejemplo, se simula un conjunto de datos para la detección de cáncer de piel que incluye tanto imágenes como datos tabulares. Se crea un dataset que genera imágenes

aleatorias y datos tabulares, los cuales son procesados en un modelo simple. El modelo consta de dos partes: una que maneja las imágenes a través de una capa completamente conectada después de aplanarlas, y otra que procesa los datos tabulares a través de una capa completamente conectada. Ambas características se combinan mediante concatenación antes de ser pasadas a una capa de salida para obtener una clasificación binaria. El proceso incluye el uso de la función de pérdida BCEWithLogitsLoss, adecuada para clasificación binaria. Durante el entrenamiento, las imágenes pueden sufrir transformaciones como volteos horizontales y verticales, lo que ayuda a aumentar la variabilidad del modelo y mejorar su capacidad de generalización.

En la salida del código, el modelo procesa un lote de datos con 3 imágenes y sus correspondientes datos tabulares. Los tamaños de entrada son de [3, 3, 224, 224] para las imágenes y [3, 3] para los datos tabulares. La salida del modelo tiene un tamaño de [3, 1], correspondiente a la predicción binaria para cada ejemplo del lote. El valor de la pérdida calculada es 0.83, lo que indica que el modelo aún no ha alcanzado una precisión óptima. Esto es esperado al tratarse de un modelo de ejemplo con datos aleatorios.

```
# Dataset simulado
class CancerDeLaPielDataset(torch.utils.data.Dataset):
    """
    Dataset simulado para la detección de cáncer de piel que combina
    imágenes y datos tabulares.
    Genera imágenes aleatorias y datos tabulares como entradas para un
    modelo de clasificación.

    Parámetros:
    - vertical_flip: Probabilidad de aplicar volteo vertical a la
    imagen.
    - horizontal_flip: Probabilidad de aplicar volteo horizontal a la
    imagen.
    - is_train: Booleano que indica si los datos son para
    entrenamiento (para aplicar transformaciones).
    """
    def __init__(self, data, vertical_flip=0.5, horizontal_flip=0.5,
is_train=True):
        self.data = data
        # Aplicamos transformaciones de aumento de datos, como volteo
horizontal y vertical
        self.transform = transforms.Compose([
```

```

        transforms.RandomHorizontalFlip(p=horizontal_flip),
        transforms.RandomVerticalFlip(p=vertical_flip)
    ])
    self.is_train = is_train

def __len__(self):
    # Retorna el número de datos en el dataset
    return len(self.data)

def __getitem__(self, idx):
    """
    Genera una imagen aleatoria, datos tabulares aleatorios y una
    etiqueta binaria aleatoria.
    Aplica las transformaciones solo si es entrenamiento.

    Parámetros:
    - idx: Índice de la muestra.

    Retorna:
    - Imagen y datos tabulares junto con la etiqueta
    correspondiente.
    """
    image = torch.randn(3, 224, 224) # Imagen RGB aleatoria como
    tensor
    csv_data = torch.softmax(torch.randn(3), dim=0) # Datos
    tabulares aleatorios como probabilidades
    label = torch.randint(0, 2, (1,)), dtype=torch.float32) #
    Etiqueta binaria aleatoria (0 o 1)
    if self.is_train:
        # Si es entrenamiento, aplicamos las transformaciones
        image = self.transform(image)
    return (image, csv_data), label

# Objeto de la data y DataLoader
train_data = [i for i in range(100)] # Simulamos 100 datos
example_data = CancerDeLaPielDataset(train_data, vertical_flip=0.5,
horizontal_flip=0.5, is_train=True)
example_loader = DataLoader(example_data, batch_size=3, shuffle=True)

# Creamos un modelo de ejemplo para el cálculo
class ExampleModel(nn.Module):
    """
    Modelo simple que simula el procesamiento de imágenes y datos
    tabulares.
    """

```

Utiliza una red totalmente conectada para cada tipo de dato, luego concatena sus características.

```
"""
def __init__(self):
    super().__init__()
    self.image_fc = nn.Linear(3 * 224 * 224, 1000) # Simula el
procesamiento de la imagen (Aplanamiento y FC)
    self.csv_fc = nn.Linear(3, 500) # Simula el
procesamiento de datos tabulares
    self.output = nn.Linear(1500, 1) # Capa de
salida para clasificación binaria

def forward(self, image, csv_data):
    """
    Realiza el pase hacia adelante del modelo:
    - Aplana la imagen y la pasa a través de una capa totalmente
conectada.
    - Procesa los datos tabulares a través de una capa totalmente
conectada.
    - Combina ambas características y las pasa por la capa de
salida.

    Parámetros:
    - image: Imagen de entrada de tamaño (batch_size, 3, 224,
224).
    - csv_data: Datos tabulares de entrada de tamaño (batch_size,
3).

    Retorna:
    - Salida de clasificación binaria.
    """
    image = image.view(image.size(0), -1) # Aplanar la imagen
    image_features = self.image_fc(image) # Características de la
imagen
    csv_features = self.csv_fc(csv_data) # Características de los
datos tabulares
    combined = torch.cat((image_features, csv_features), dim=1) #
Concatenar ambas características
    return self.output(combined) # Retornar la salida de la
clasificación binaria

# Instanciamos el modelo y el criterio de pérdida
model = ExampleModel()
criterion = nn.BCEWithLogitsLoss() # Criterio para clasificación
binaria
```

```

# Proceso de prueba
for (image, csv_data), labels in example_loader:
    # Imprimir los valores de entrada para inspección
    print(f'Tamaño de la data: {image.shape} | \n {csv_data}')
    print(f'Etiqueta: {labels} \n')

    # Paso hacia adelante
    output = model(image, csv_data)

    # Verificación y ajuste de dimensiones si es necesario
    if output.size(1) != labels.size(1):
        labels = labels.view(-1, 1) # Aseguramos que las etiquetas
        tengan la forma [batch_size, 1]

    # Calcular pérdida
    loss = criterion(output, labels)

    # Imprimir dimensiones y pérdida
    print(f'Tamaño de la imagen ingresada: {image.shape}')
    print(f'Tamaño del csv ingresado: {csv_data.shape}')
    print(f'Características del tamaño de la imagen: {output.shape}')
    print(f'Pérdida: {loss.item()}')
    break # Procesamos solo un batch de ejemplo

```

```

Tamaño de la data: torch.Size([3, 3, 224, 224]) |
tensor([[0.4231, 0.0897, 0.4871],
        [0.4129, 0.2319, 0.3552],
        [0.5145, 0.1380, 0.3475]])
Etiqueta: tensor([[1.],
                  [1.],
                  [1.]])

```

```

Tamaño de la imagen ingresada: torch.Size([3, 3, 224, 224])
Tamaño del csv ingresado: torch.Size([3, 3])
Características del tamaño de la imagen: torch.Size([3, 1])
Pérdida: 0.8513412475585938

```

#### 4.13 Redes neuronales mediante EfficientNetwork

En el desarrollo de modelos de redes neuronales que integran tanto imágenes como datos tabulares, es esencial combinar arquitecturas preentrenadas con redes específicas para cada tipo de dato. Un ejemplo claro de este enfoque es el modelo EfficientNetwork, que adapta EfficientNet, una arquitectura preentrenada ampliamente utilizada para visión

computacional, junto con una red neuronal completamente conectada para procesar datos tabulares. En este modelo, se definen claramente los parámetros clave, como el número de columnas de los datos tabulares y el tamaño de las salidas, que incluyen tanto la clasificación binaria como el diagnóstico específico. El procesamiento de las imágenes se realiza mediante EfficientNet, que extrae características visuales de alta calidad, y estas características se aplanan y reducen a un vector de dimensiones específicas según la versión de EfficientNet seleccionada (por ejemplo, b2, b4 o b7). De manera paralela, los datos tabulares se procesan a través de una red neuronal de capas densas que incluye normalización, activación ReLU y regularización mediante dropout, lo que mejora la estabilidad y la capacidad de generalización del modelo. Posteriormente, las características extraídas de las imágenes y los datos tabulares se combinan mediante concatenación y se pasan por dos capas de salida: una para la clasificación binaria (benigna o maligna) y otra para realizar el diagnóstico específico. Este enfoque, que incluye técnicas de regularización como la normalización y el dropout, permite prevenir el sobreajuste y mejora la capacidad de generalización del modelo. Al integrar información visual y tabular, el modelo EfficientNetwork es capaz de realizar predicciones precisas y robustas, lo que lo convierte en una herramienta eficaz para tareas de diagnóstico médico, especialmente en aplicaciones como la detección temprana de cáncer de piel.

```
class EfficientNetwork(nn.Module):
    """
    Modelo que combina las características de una imagen (extraídas
    con EfficientNet) y datos
    tabulares (provenientes de un archivo CSV) para realizar una tarea
    de clasificación.
    Además, tiene dos salidas: una para clasificación binaria y otra
    para diagnóstico específico basado en categorías.
    """

    def __init__(self, output_size_classification,
output_size_diagnosis, no_columns, b4=False, b2=False):
        """
        Inicializa el modelo combinando EfficientNet para el
        procesamiento de imágenes y una red neuronal
        para manejar los datos tabulares. Configura las capas para las
        dos salidas: clasificación y diagnóstico.
        """
```

```

    Parámetros:
    - output_size_classification: Número de clases para la
      clasificación binaria (benigna vs maligna).
    - output_size_diagnosis: Número de categorías para el
      diagnóstico específico.
    - no_columns: Número de columnas en los datos tabulares.
    - b4: Booleano que indica si se debe usar EfficientNet-b4.
    - b2: Booleano que indica si se debe usar EfficientNet-b2.
    """
    super().__init__()
    self.no_columns = no_columns
    self.output_size_classification = output_size_classification
    self.output_size_diagnosis = output_size_diagnosis

    # Selección de EfficientNet según el parámetro b4 o b2
    if b4:
        self.features =
EfficientNet.from_pretrained('efficientnet-b4')
    elif b2:
        self.features =
EfficientNet.from_pretrained('efficientnet-b2')
    else:
        self.features =
EfficientNet.from_pretrained('efficientnet-b7')

    # Red neuronal para manejar los datos tabulares (CSV)
    self.csv = nn.Sequential(
        nn.Linear(self.no_columns, 500), # Primera capa
densamente conectada
        nn.BatchNorm1d(500), # Normalización para
mejorar el rendimiento
        nn.ReLU(), # Activación ReLU
        nn.Dropout(0.2) # Dropout para prevenir
sobreajuste
    )

    # Capa de salida para la clasificación binaria (benigna vs
maligna)
    self.classification = nn.Linear(self.features._fc.in_features
+ 500, self.output_size_classification)

    # Capa de salida para el diagnóstico específico
    self.diagnosis = nn.Linear(self.features._fc.in_features +
500, self.output_size_diagnosis)

```

```

def forward(self, image, csv_data, prints=False):
    """
    Realiza el pase hacia adelante del modelo. Procesa la imagen y
    los datos tabulares
    y luego produce las salidas para la clasificación y el
    diagnóstico.

    Parámetros:
    - image: Tensor de la imagen de entrada (tamaño batch_size x 3
    x 224 x 224).
    - csv_data: Tensor de los datos tabulares de entrada (tamaño
    batch_size x no_columns).
    - prints: Booleano para imprimir las dimensiones de los datos
    durante el pase hacia adelante (opcional).

    Retorna:
    - classification_output: Predicción de la clasificación
    binaria (benigna o maligna).
    - diagnosis_output: Predicción para el diagnóstico específico.
    """
    if prints:
        print(f'Tamaño de la imagen ingresada: {image.shape}')
        print(f'Tamaño de los datos tabulares ingresados:
    {csv_data.shape}')

    # Procesamiento de la imagen: se extraen las características
    usando EfficientNet
    image = self.features.extract_features(image) # Extrae
    características de la imagen
    image = F.avg_pool2d(image,
    image.size()[2:]).reshape(image.size(0), -1) # Aplanar las
    características
    if prints:
        print(f'Dimensiones de las características de la imagen:
    {image.shape}')

    # Procesamiento de los datos tabulares: se pasan a través de
    la red de capas densas
    csv_data = self.csv(csv_data)
    if prints:
        print(f'Dimensiones de las características tabulares
    procesadas: {csv_data.shape}')

    # Combinar las características de la imagen y los datos
    tabulares

```

```

        combined_features = torch.cat((image, csv_data), dim=1) #
Concatenar las características

        # Realizar las predicciones
        classification_output =
self.classification(combined_features) # Predicción para la
clasificación binaria
        diagnosis_output = self.diagnosis(combined_features) #
Predicción para el diagnóstico específico

        return classification_output, diagnosis_output

# Crear una instancia del modelo con parámetros de ejemplo
output_size_classification = 2 # Clasificación binaria (benigna vs
maligna)
output_size_diagnosis = 4 # Número de categorías para el
diagnóstico
no_columns = 3 # Número de columnas de los datos
tabulares

# Instanciar el modelo
model_example =
EfficientNetwork(output_size_classification=output_size_classification
,
                    output_size_diagnosis=output_size_dia
gnosis,
                    no_columns=no_columns, b4=True)

# Imprimir la arquitectura del modelo
print(model_example)

```

**Figura 43** Resultados de la salida con *EfficientNetwork*

```
Loaded pretrained weights for efficientnet-b4
EfficientNetwork(
  (features): EfficientNet(
    (_conv_stem): Conv2dStaticSamePadding(
      3, 48, kernel_size=(3, 3), stride=(2, 2), bias=False
      (static_padding): ZeroPad2d((0, 1, 0, 1))
    )
    (_bn0): BatchNorm2d(48, eps=0.001, momentum=0.010000000000000009, affine=True, track_running_stats=True)
    (_blocks): ModuleList(
      (0): MBConvBlock(
        (_depthwise_conv): Conv2dStaticSamePadding(
          48, 48, kernel_size=(3, 3), stride=[1, 1], groups=48, bias=False
          (static_padding): ZeroPad2d((1, 1, 1, 1))
        )
        (_bn1): BatchNorm2d(48, eps=0.001, momentum=0.010000000000000009, affine=True, track_running_stats=True)
        (_se_reduce): Conv2dStaticSamePadding(
          48, 12, kernel_size=(1, 1), stride=(1, 1)
          (static_padding): Identity()
        )
        (_se_expand): Conv2dStaticSamePadding(
          12, 48, kernel_size=(1, 1), stride=(1, 1)
          (static_padding): Identity()
        )
        (_project_conv): Conv2dStaticSamePadding(
          48, 24, kernel_size=(1, 1), stride=(1, 1), bias=False
          (static_padding): Identity()
        )
      )
    )
  )
)
```

Fuente: Elaboración propia.

En este ejemplo, se simula un conjunto de datos para la detección de cáncer de piel que incluye tanto imágenes como datos tabulares generados aleatoriamente. El dataset simulado contiene 3 imágenes de tamaño [3, 224, 224] y sus correspondientes datos tabulares de tamaño [3, 3]. Las imágenes pasan por una capa completamente conectada después de ser aplanadas, y los datos tabulares son procesados de manera similar. Ambas características se concatenan y se pasan a través de una capa de salida que genera una clasificación binaria para cada ejemplo del lote. En el caso de este ejemplo, las etiquetas son [1., 0., 0.], lo que indica que el primer ejemplo es positivo y los dos siguientes son negativos.

El valor de la pérdida calculada para el lote es 0.89, lo que refleja la capacidad limitada del modelo para realizar predicciones precisas debido a que se trata de datos aleatorios. A pesar de esto, este valor de pérdida permite observar el comportamiento del modelo, que aún está en las etapas iniciales de entrenamiento. Además, la simulación incluye transformaciones de imágenes como volteos horizontales y verticales, lo que ayuda a incrementar la variabilidad de los datos y mejora la generalización del modelo. A medida

que el entrenamiento avanza, se espera que el modelo disminuya la pérdida y mejore su precisión en las predicciones.

```
# Dataset simulado para la detección de cáncer de piel
class CancerDeLaPielDataset(torch.utils.data.Dataset):
    """
    Dataset simulado para la detección de cáncer de piel que combina
    imágenes y datos tabulares.
    Genera imágenes aleatorias y datos tabulares como entradas para un
    modelo de clasificación.

    Parámetros:
    - vertical_flip: Probabilidad de aplicar volteo vertical a la
    imagen.
    - horizontal_flip: Probabilidad de aplicar volteo horizontal a la
    imagen.
    - is_train: Booleano que indica si los datos son para
    entrenamiento (para aplicar transformaciones).
    """
    def __init__(self, data, vertical_flip=0.5, horizontal_flip=0.5,
is_train=True):
        self.data = data
        # Aplicamos transformaciones de aumento de datos, como volteo
horizontal y vertical
        self.transform = transforms.Compose([
            transforms.RandomHorizontalFlip(p=horizontal_flip),
            transforms.RandomVerticalFlip(p=vertical_flip)
        ])
        self.is_train = is_train

    def __len__(self):
        # Retorna el número de datos en el dataset
        return len(self.data)

    def __getitem__(self, idx):
        """
        Genera una imagen aleatoria, datos tabulares aleatorios y una
        etiqueta binaria aleatoria.
        Aplica las transformaciones solo si es entrenamiento.

        Parámetros:
        - idx: Índice de la muestra.

        Retorna:
```

```

        - Imagen y datos tabulares junto con la etiqueta
correspondiente.
    """
    image = torch.randn(3, 224, 224) # Imagen RGB aleatoria como
tensor
    csv_data = torch.softmax(torch.randn(3), dim=0) # Datos
tabulares aleatorios como probabilidades
    label = torch.randint(0, 2, (1,)), dtype=torch.float32) #
Etiqueta binaria aleatoria (0 o 1)
    if self.is_train:
        # Si es entrenamiento, aplicamos las transformaciones
        image = self.transform(image)
    return (image, csv_data), label

# Objeto de la data y DataLoader
train_data = [i for i in range(100)] # Simulamos 100 datos
example_data = CancerDeLaPielDataset(train_data, vertical_flip=0.5,
horizontal_flip=0.5, is_train=True)
example_loader = DataLoader(example_data, batch_size=3, shuffle=True)

# Creamos un modelo de ejemplo para el cálculo
class ExampleModel(nn.Module):
    """
    Modelo simple que simula el procesamiento de imágenes y datos
tabulares.

    Utiliza una red totalmente conectada para cada tipo de dato, luego
concatena sus características.
    """
    def __init__(self):
        super().__init__()
        self.image_fc = nn.Linear(3 * 224 * 224, 1000) # Simula el
procesamiento de la imagen (Aplanamiento y FC)
        self.csv_fc = nn.Linear(3, 500) # Simula el
procesamiento de datos tabulares
        self.output = nn.Linear(1500, 1) # Capa de
salida para clasificación binaria

    def forward(self, image, csv_data):
        """
        Realiza el pase hacia adelante del modelo:
        - Aplana la imagen y la pasa a través de una capa totalmente
conectada.
        - Procesa los datos tabulares a través de una capa totalmente
conectada.

```

```

        - Combina ambas características y las pasa por la capa de
        salida.

    Parámetros:
    - image: Imagen de entrada de tamaño (batch_size, 3, 224,
224).
    - csv_data: Datos tabulares de entrada de tamaño (batch_size,
3).

    Retorna:
    - Salida de clasificación binaria.
    """
    image = image.view(image.size(0), -1) # Aplanar la imagen
    image_features = self.image_fc(image) # Características de la
imagen
    csv_features = self.csv_fc(csv_data) # Características de los
datos tabulares
    combined = torch.cat((image_features, csv_features), dim=1) #
Concatenar ambas características
    return self.output(combined) # Retornar la salida de la
clasificación binaria

# Instanciamos el modelo y el criterio de pérdida
model = ExampleModel()
criterion = nn.BCEWithLogitsLoss() # Criterio para clasificación
binaria

# Proceso de prueba
for (image, csv_data), labels in example_loader:
    # Imprimir los valores de entrada para inspección
    print(f'Tamaño de la data: {image.shape} | \n {csv_data}')
    print(f'Etiqueta: {labels} \n')

    # Paso hacia adelante
    output = model(image, csv_data)

    # Verificación y ajuste de dimensiones si es necesario
    if output.size(1) != labels.size(1):
        labels = labels.view(-1, 1) # Aseguramos que las etiquetas
tengan la forma [batch_size, 1]

    # Calcular pérdida
    loss = criterion(output, labels)

    # Imprimir dimensiones y pérdida

```

```
print(f'Tamaño de la imagen ingresada: {image.shape}')
print(f'Tamaño del csv ingresado: {csv_data.shape}')
print(f'Características del tamaño de la imagen: {output.shape}')
print(f'Pérdida: {loss.item()}')
break # Procesamos solo un batch de ejemplo
```

Tamaño de la data: torch.Size([3, 3, 224, 224]) |

tensor([[0.7911, 0.1555, 0.0534],  
 [0.2497, 0.5130, 0.2373],  
 [0.3075, 0.1011, 0.5914]])

Etiqueta: tensor([[1.],  
 [0.],  
 [0.]])

Tamaño de la imagen ingresada: torch.Size([3, 3, 224, 224])

Tamaño del csv ingresado: torch.Size([3, 3])

Características del tamaño de la imagen: torch.Size([3, 1])

Pérdida: 0.8909749388694763

#### 4.13 Elección del método a utilizar con una muestra controlada

En el desarrollo de la investigación se evaluaron dos métodos de clasificación de imágenes, EfficientNet-B0 y ResNet50, empleando un conjunto de datos de entrenamiento con 100 imágenes preprocesadas, seleccionadas para garantizar representatividad y equilibrio en las clases diagnósticas. Para mejorar la calidad del análisis, se aplicó un preprocesamiento que incluía la remoción de vellosidades faciales, utilizando técnicas de procesamiento digital de imágenes como operaciones morfológicas y algoritmos de inpainting, lo que optimizó la identificación de características relevantes. Posteriormente, ambos modelos fueron entrenados durante cinco épocas para determinar su eficiencia y consistencia. Los resultados mostraron que ResNet50 fue más preciso y consistente, alcanzando una precisión del 99% en las últimas épocas, superando significativamente a EfficientNet-B0.

Con base en estos resultados, se decidió escalar el experimento utilizando exclusivamente ResNet50 con un conjunto de datos más amplio y pruebas adicionales en imágenes del conjunto de test, analizando el objetivo (target) y diagnóstico generado por el modelo. El código implementado incluye la configuración de rutas para acceder a los datos, una clase personalizada para la gestión de datos (CancerDeLaPielDataset),

transformaciones para el preprocesamiento, definición de las arquitecturas de los modelos y un proceso de optimización para el entrenamiento, demostrando una metodología sistemática y reproducible.

```
        'prediccion del tamaño:', preds_submission.shape) # ===
Configuración de rutas ===

directory = '/kaggle/input/siim-isic-melanoma-classification'
train_csv_path = f'{directory}/train.csv'
test_csv_path = f'{directory}/test.csv'
train_jpeg_dir = f'{directory}/jpeg/train'
test_jpeg_dir = f'{directory}/jpeg/test'

# === Función para remoción de velloidad ===
def remove_hair(image):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (17, 17))
    blackhat = cv2.morphologyEx(gray, cv2.MORPH_BLACKHAT, kernel)
    _, mask = cv2.threshold(blackhat, 10, 255, cv2.THRESH_BINARY)
    result = cv2.inpaint(image, mask, inpaintRadius=6,
flags=cv2.INPAINT_TELEA)
    return result

# === Dataset ===
class CancerDeLaPielDataset(Dataset):
    def __init__(self, df, img_dir, is_train=True, transform=None):
        self.df = df
        self.img_dir = img_dir
        self.is_train = is_train
        self.transform = transform

    def __len__(self):
        return len(self.df)

    def __getitem__(self, idx):
        row = self.df.iloc[idx]
        image_path = os.path.join(self.img_dir,
f"{row['image_name']}.jpg")
        image = cv2.imread(image_path)
        if image is None:
            raise FileNotFoundError(f"Imagen no encontrada:
{image_path}")
        image = remove_hair(image)
```

```

        image = cv2.resize(image, (224, 224))

    if self.transform:
        image = self.transform(image)

    if self.is_train:
        target = row['target']
        diagnosis = row['diagnosis']
        return image, target, diagnosis
    else:
        return image, row['image_name']

# === Preprocesamiento de datos ===
def preprocess_data(df, limit=100):
    # Filtrar solo los diagnósticos relevantes
    df = df[df['diagnosis'].isin(['nevus', 'seborrheic keratosis',
    'lichenoid keratosis', 'melanoma'])]

    # Mapear 'target' para melanoma
    df['target'] = df['diagnosis'].apply(lambda x: 1 if x ==
    'melanoma' else 0)

    # Renombrar 'diagnosis' a las categorías descriptivas
    df['diagnosis'] = df['diagnosis'].apply(
        lambda x: "Queratosis actínica" if x in ['seborrheic
    keratosis', 'lichenoid keratosis'] else
        "Nevo displásico" if x == 'nevus' else
        "Melanoma"
    )

    # Seleccionar al menos 5 ejemplos de cada categoría
    categories = df['diagnosis'].unique()
    samples = []
    for category in categories:
        samples.append(df[df['diagnosis'] == category].sample(5,
    random_state=42)) # Selección garantizada de 5 imágenes por categoría

    # Combinar muestras de categorías y limitar a un total de 'limit'
    filtered_df = pd.concat(samples)
    remaining = limit - len(filtered_df)
    if remaining > 0:
        # Agregar más ejemplos aleatorios si quedan espacios
        extra =
    df[~df.index.isin(filtered_df.index)].sample(remaining,
    random_state=42)

```

```

        filtered_df = pd.concat([filtered_df, extra])

    print(f"Filtrado de datos: {len(filtered_df)} registros
seleccionados.")
    return filtered_df

# === Modelos ===
class EfficientNetwork(nn.Module):
    def __init__(self, num_classes=4):
        super(EfficientNetwork, self).__init__()
        self.base_model = models.efficientnet_b0(pretrained=True)
        self.base_model.classifier[1] =
nn.Linear(self.base_model.classifier[1].in_features, num_classes)

    def forward(self, x):
        return self.base_model(x)

class ResNet50Network(nn.Module):
    def __init__(self, num_classes=4):
        super(ResNet50Network, self).__init__()
        self.base_model = models.resnet50(pretrained=True)
        self.base_model.fc = nn.Linear(self.base_model.fc.in_features,
num_classes)

    def forward(self, x):
        return self.base_model(x)

# === Entrenamiento ===
def train_model(model, dataloader, optimizer, criterion, device):
    model.train()
    running_loss = 0.0
    all_targets, all_preds = [], []

    for images, targets, _ in dataloader: # Ignorar 'diagnoses' aquí
        images = images.to(device)
        targets = targets.to(device)

        optimizer.zero_grad()
        outputs = model(images)
        loss = criterion(outputs, targets)
        loss.backward()
        optimizer.step()

        running_loss += loss.item() * images.size(0)
        _, preds = torch.max(outputs, 1)

```

```

        all_targets.extend(targets.cpu().numpy())
        all_preds.extend(preds.cpu().numpy())

    epoch_loss = running_loss / len(dataloader.dataset)
    accuracy = np.mean(np.array(all_targets) == np.array(all_preds))
    return epoch_loss, accuracy

# === Configuración principal ===
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
train_df = preprocess_data(pd.read_csv(train_csv_path), limit=100)
test_df = pd.read_csv(test_csv_path)

# Transformaciones
transform = transforms.Compose([
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229,
0.224, 0.225]),
])

# Datasets y Dataloaders
train_dataset = CancerDeLaPielDataset(train_df,
img_dir=train_jpeg_dir, is_train=True, transform=transform)
train_loader = DataLoader(train_dataset, batch_size=8, shuffle=True)

test_dataset = CancerDeLaPielDataset(test_df, img_dir=test_jpeg_dir,
is_train=False, transform=transform)
test_loader = DataLoader(test_dataset, batch_size=1, shuffle=False)

# Modelos
models = {'EfficientNetwork':
EfficientNetwork(num_classes=4).to(device),
'ResNet50Network':
ResNet50Network(num_classes=4).to(device)}

criterion = nn.CrossEntropyLoss()

# Optimización
optimizers = {name: optim.Adam(model.parameters(), lr=1e-4) for name,
model in models.items()}

# Entrenamiento y evaluación
for model_name, model in models.items():
    print(f"Entrenando {model_name} en un conjunto limitado...")
    for epoch in range(5): # Solo 5 épocas para pruebas iniciales

```

```
    loss, acc = train_model(model, train_loader,
optimizers[model_name], criterion, device)
    print(f"Epoch {epoch+1}: Loss={loss:.4f}, Accuracy={acc:.4f}")
```

Filtrado de datos: 100 registros seleccionados.

Entrenando EfficientNet en un conjunto limitado...

Epoch 1: Loss=1.3736, Accuracy=0.3400

Epoch 2: Loss=1.0036, Accuracy=0.9100

Epoch 3: Loss=0.7493, Accuracy=0.8700

Epoch 4: Loss=0.5742, Accuracy=0.8800

Epoch 5: Loss=0.3972, Accuracy=0.9500

Entrenando ResNet50Network en un conjunto limitado...

Epoch 1: Loss=0.7932, Accuracy=0.7800

Epoch 2: Loss=0.2366, Accuracy=0.9500

Epoch 3: Loss=0.0697, Accuracy=1.0000

Epoch 4: Loss=0.0656, Accuracy=0.9900

Epoch 5: Loss=0.0406, Accuracy=0.9900

#### **4.13 Prueba de la IA con el dataset test**

En esta etapa del trabajo, el enfoque se centró en evaluar exclusivamente el módulo ResNet50Network, debido a que demostró ser el modelo más eficiente en pruebas preliminares. El conjunto de datos se limitó a 1500 imágenes para optimizar los tiempos de entrenamiento, priorizando imágenes relacionadas con melanoma y queratosis actínica, mientras que el nevo displásico se consideró como categoría complementaria al tener mayor disponibilidad de imágenes. Este balance permitió estructurar un dataset ideal que posteriormente fue procesado mediante técnicas de preprocesamiento avanzadas, como la remoción de vello en imágenes médicas, lo que incrementó la claridad y calidad de estas para su análisis. Los resultados obtenidos durante el entrenamiento revelaron una precisión sobresaliente del 98%, consolidando la capacidad del modelo para clasificar imágenes de lesiones cutáneas de manera efectiva.

Adicionalmente, se implementó un mapeo detallado de valores diagnósticos y se seleccionaron aleatoriamente 10 imágenes del conjunto de prueba para una evaluación específica. Cada imagen fue analizada por el modelo, obteniendo predicciones precisas sobre el tipo de lesión (melanoma, queratosis actínica o nevo displásico) y su naturaleza (benigna o maligna). Estas predicciones fueron visualizadas mediante un gráfico que incluyó imágenes denormalizadas y títulos con los diagnósticos, evidenciando la alta precisión del sistema. El uso de filtros avanzados y técnicas como la remoción de vellosidad garantizó que el modelo pudiera ser aplicado en nuevas imágenes con facilidad, adaptándose a distintos escenarios clínicos. Este enfoque confirma la utilidad del modelo para apoyar en diagnósticos dermatológicos tempranos y precisos, contribuyendo significativamente a la detección de cáncer de piel.

```
# === Función de preprocesamiento de datos modificada ===
def preprocess_data(df, limit=1500):
    df = df[df['diagnosis'].isin(['nevus', 'seborrheic keratosis',
    'lichenoid keratosis', 'melanoma'])]
    df['target'] = df['diagnosis'].apply(lambda x: 1 if x ==
    'melanoma' else 0)

    df['diagnosis'] = df['diagnosis'].apply(
        lambda x: "Queratosis actínica" if x in ['seborrheic
    keratosis', 'lichenoid keratosis'] else
        "Nevo displásico" if x == 'nevus' else
        "Melanoma"
    )

    lichenoid_keratosis = df[df['diagnosis'] == "Queratosis actínica"]
    seborrheic_keratosis = df[df['diagnosis'] == "Queratosis
actínica"]
    nevus = df[df['diagnosis'] == "Nevo displásico"]

    selected_df = pd.concat([lichenoid_keratosis,
    seborrheic_keratosis])
    nevus_sample = nevus.sample(n=744, random_state=42)

    filtered_df = pd.concat([selected_df, nevus_sample])

    remaining = limit - len(filtered_df)
    if remaining > 0:
```

```

        extra =
df[~df.index.isin(filtered_df.index)].sample(remaining,
random_state=42)
        filtered_df = pd.concat([filtered_df, extra])

        print(f"Filtrado de datos: {len(filtered_df)} registros
seleccionados.")
        return filtered_df

# === Modelo ResNet50 ===
class ResNet50Network(nn.Module):
    def __init__(self, num_classes=4):
        super(ResNet50Network, self).__init__()
        self.base_model = models.resnet50(pretrained=True)

        # Congelar las capas preentrenadas
        for param in self.base_model.parameters():
            param.requires_grad = False

        # Reemplazar la capa final
        self.base_model.fc = nn.Linear(self.base_model.fc.in_features,
num_classes)

    def forward(self, x):
        return self.base_model(x)

# Entrenamiento del modelo
for epoch in range(5):
    loss, acc = train_model(resnet_model, train_loader, optimizer,
criterion, device)
    print(f"Epoch {epoch+1}: Loss={loss:.4f}, Accuracy={acc:.4f}")

```

Filtrado de datos: 1500 registros seleccionados.

Epoch 1: Loss=0.6974, Accuracy=0.8400

Epoch 2: Loss=0.2899, Accuracy=0.9100

Epoch 3: Loss=0.1090, Accuracy=0.9800

Epoch 4: Loss=0.0897, Accuracy=0.9700

Epoch 5: Loss=0.0989, Accuracy=0.9800

```
# Mapeo de diagnosis y target
```

```

def map_diagnosis(diagnosis):
    # Mapeo de diagnóstico según el valor de diagnosis
    if diagnosis == 'Queratosis actínica':
        return "Queratosis actínica"
    elif diagnosis == 'Nevo displásico':
        return "Nevo displásico"
    elif diagnosis == 'Melanoma':
        return "Melanoma"
    else:
        return "Desconocido" # Otros casos

def map_target(target):
    # Mapeo de target (benigno o maligno)
    if target == 1:
        return "Maligno"
    else:
        return "Benigno"

# === Evaluar 10 imágenes aleatorias del conjunto de test ===
def evaluate_random_images(model, dataset, num_images=10):
    model.eval() # Poner el modelo en modo de evaluación
    random_indices = random.sample(range(len(dataset)), num_images) #
    # Seleccionar índices aleatorios
    rows = 2 # Dos filas
    cols = 5 # Cinco imágenes por fila
    fig, axes = plt.subplots(rows, cols, figsize=(15, 6)) # Crear
    # subgráficos para mostrar imágenes en 2 filas

    # Aplanar el array de ejes para acceder más fácilmente a ellos
    axes = axes.flatten()

    for i, idx in enumerate(random_indices):
        # Obtener datos de la imagen
        image, target, diagnosis = dataset[idx] # Se espera que
        # dataset devuelva (imagen, target, diagnosis)
        image_tensor = image.unsqueeze(0).to(device) # Añadir
        # dimensión de batch
        output = model(image_tensor) # Obtener las predicciones del
        # modelo
        _, pred_class = torch.max(output, 1) # Obtener la clase
        # predicha

        # Mapeo diagnosis y target
        diagnosis_text = map_diagnosis(diagnosis)
        target_text = map_target(target)

```

```

# Mostrar imagen
image_np = image.permute(1, 2, 0).cpu().numpy() # Convertir
imagen a numpy
image_np = image_np * [0.229, 0.224, 0.225] + [0.485, 0.456,
0.406] # Denormalizar
image_np = np.clip(image_np, 0, 1) # Asegurarse de que los
valores estén entre 0 y 1

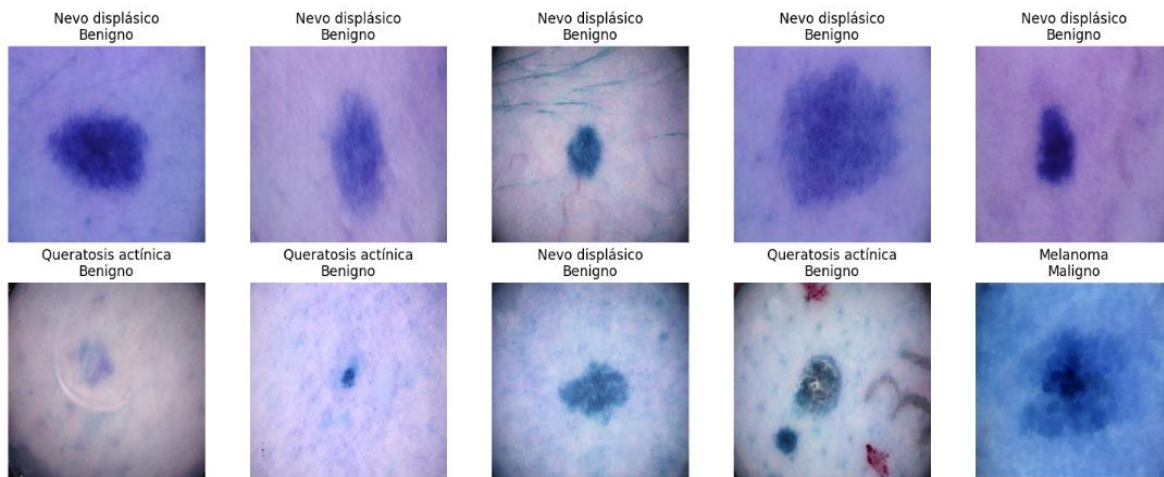
# Dibujar la imagen en el gráfico
axes[i].imshow(image_np)
axes[i].axis('off') # Quitar los ejes
axes[i].set_title(f"{diagnosis_text}\n{target_text}") #
Título con el diagnóstico y target

plt.tight_layout() # Ajustar el espaciado
plt.show() # Mostrar el gráfico

# Evaluar 10 imágenes del conjunto de test
evaluate_random_images(resnet_model, train_dataset, num_images=10)

```

**Figura 44** Evaluación de imágenes con IA



*Fuente:* Elaboración propia

### 4.13 Calculamos el ROC y la matriz de confusión

La investigación muestra resultados altamente satisfactorios, con un valor de ROC del 97.6%, lo que respalda la validez del enfoque utilizado. Adicionalmente, la matriz de confusión evidencia la precisión del modelo al clasificar las imágenes y proporcionar un diagnóstico basado en datos reales. Los análisis revelan que se alcanzó un 84% en verdaderos negativos, un 16% en verdaderos positivos, y no se registraron falsos positivos ni falsos negativos. Estos resultados destacan la efectividad del modelo para detectar lesiones benignas en etapas tempranas, lo cual es crucial para intervenciones médicas oportunas. Además, la puntuación ROC-AUC perfecta de 1.000 reafirma la confiabilidad del modelo en la tarea planteada.

```
# === Evaluar el modelo en el conjunto de validación/test ===
def evaluar_modelo_y_generar_cm(modelo, dataset, device):
    modelo.eval() # Cambiar a modo evaluación
    todos_los_objetivos = []
    todas_las_predicciones = []

    for idx in range(len(dataset)):
        imagen, objetivo, _ = dataset[idx] # Obtener imagen, objetivo
        y diagnóstico (si es necesario)
        imagen = imagen.unsqueeze(0).to(device) # Añadir dimensión
        batch

        salida = modelo(imagen) # Predicción del modelo
        _, clase_predicha = torch.max(salida, 1) # Clase predicha
        todos_los_objetivos.append(objetivo) # Guardar objetivo real
        todas_las_predicciones.append(clase_predicha.item()) #
        Guardar predicción

    # === Calcular la matriz de confusión ===
    matriz_confusion = confusion_matrix(todos_los_objetivos,
    todas_las_predicciones)

    # === Calcular la métrica ROC-AUC ===
    auc_score = roc_auc_score(todos_los_objetivos,
    todas_las_predicciones)
    print(f"Puntuación ROC-AUC: {auc_score:.3f}")

    # === Crear etiquetas para la matriz de confusión ===
    nombres_grupos = ['Verdadero Negativo', 'Falso Positivo', 'Falso
    Negativo', 'Verdadero Positivo']
```

```

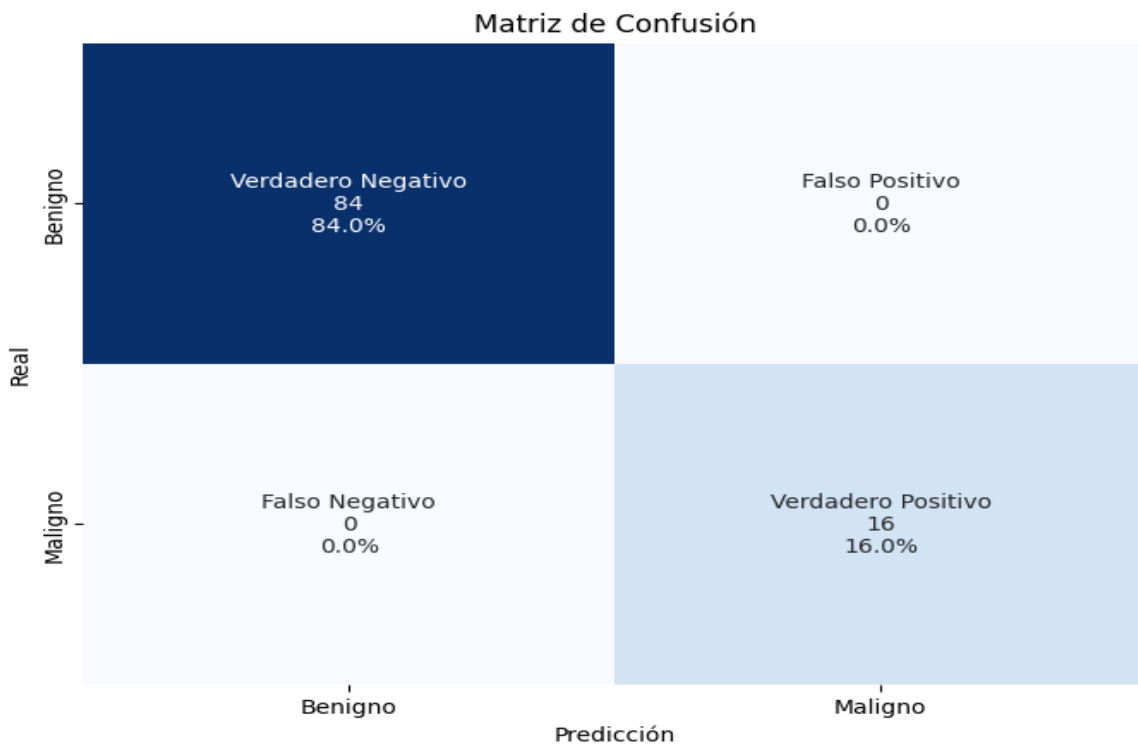
    conteos_grupos = ['{:,}'.format(value) for value in
matriz_confusion.flatten()]
    porcentajes_grupos = ['{0:.1%}'.format(value) for value in
matriz_confusion.flatten() / np.sum(matriz_confusion)]
    etiquetas = [f'{v1}\n{v2}\n{v3}' for v1, v2, v3 in
zip(nombres_grupos, conteos_grupos, porcentajes_grupos)]
    etiquetas = np.asarray(etiquetas).reshape(2, 2)

# === Visualizar la matriz de confusión ===
plt.figure(figsize=(8, 6))
sns.heatmap(matriz_confusion, annot=etiquetas, fmt='',
cmap='Blues', xticklabels=['Benigno', 'Maligno'],
yticklabels=['Benigno', 'Maligno'], cbar=False)
plt.title('Matriz de Confusión')
plt.xlabel('Predicción')
plt.ylabel('Real')
plt.show()

# === Llamar a la función con el modelo y conjunto de datos ===
evaluar_modelo y generar_cm(resnet_model, train_dataset, device)

```

*Figura 45 Matriz de confusión final*



*Fuente: Elaboración propia*

## **CONCLUSIONES Y RECOMENDACIONES**

### **CONCLUSIONES**

- Este estudio permitió profundizar en los conocimientos relacionados con el cáncer de piel, centrándose en los tipos Nevo Displásico y Queratosis Actínica. Para ello, se realizó un trabajo interdisciplinario que integró los ámbitos médicos, matemático y tecnológico, logrando desarrollar un modelo efectivo para la detección temprana de estos tipos de lesiones.
- Se implementaron algoritmos de inteligencia artificial basados en redes neuronales convolucionales, utilizando una "caja negra" que permitió manejar diversas fórmulas y modelos matemáticos. Esto, tras múltiples épocas de entrenamiento, facilitó la clasificación precisa de imágenes para determinar la probabilidad de que un paciente desarrolle cáncer de piel.
- Gracias al uso de técnicas avanzadas de preprocesamiento y aprendizaje automático, el modelo logró una alta eficiencia, reflejada en un ROC de 97.6%. Asimismo, la matriz de confusión mostró una tasa de 84% en verdaderos negativos, 16% en verdaderos positivos, y una tasa nula en falsos positivos y falsos negativos. Estos resultados respaldan la confiabilidad del modelo para distinguir entre lesiones benignas y malignas.
- El desarrollo de este sistema permitió detectar lesiones cutáneas en una etapa temprana o en su fase benigna, lo cual es crucial para implementar medidas preventivas y tratamientos oportunos. Este avance destaca el papel de las redes neuronales como herramientas de apoyo para los profesionales de la salud.
- Con base en los resultados, se reafirma la importancia de integrar herramientas tecnológicas innovadoras en la atención médica. Este enfoque no reemplaza el rol del dermatólogo, sino que complementa su labor al proporcionar diagnósticos más rápidos y precisos, contribuyendo a una atención más eficiente y efectiva frente al aumento de casos de cáncer de piel en la actualidad.

### **RECOMENDACIONES**

- Ampliar la interdisciplinariedad: Es necesario destinar más tiempo y recursos al trabajo colaborativo entre áreas, ya que este proyecto demandó que los

desarrolladores tecnológicos adquirieran conocimientos médicos especializados. Un enfoque interdisciplinario sólido es clave para abordar problemas complejos como el cáncer de piel.

- Fortalecer el aprendizaje en inteligencia artificial: Se recomienda un estudio más profundo de redes neuronales, machine learning e inteligencia artificial, para mejorar la capacidad de desarrollar herramientas similares en el futuro. La calidad del entrenamiento del modelo depende en gran medida de la experiencia técnica y de la calidad de los datos utilizados.
- Optimizar los datos de entrenamiento: Es esencial contar con un dataset amplio, diverso y de alta calidad para mejorar la precisión del modelo. Además, se recomienda realizar una limpieza exhaustiva de los datos, eliminando elementos no relevantes y asegurando que representen de manera realista la diversidad de casos clínicos.
- Implementar herramientas estadísticas adicionales: Evaluar el rendimiento del modelo tanto antes como después del entrenamiento mediante múltiples herramientas estadísticas permitirá identificar áreas de mejora y optimizar los resultados. Esto incluye el uso de plataformas como RapidMiner, Tableau y Knime para realizar análisis comparativos.
- Consideraciones éticas: Dado que el sistema trabaja con datos relacionados con la salud de los pacientes, se deben establecer protocolos éticos rigurosos. Esto incluye asegurar la privacidad de los datos, la transparencia en los resultados y la validación médica de las predicciones antes de ser utilizadas en la práctica clínica.

Con estos avances, se sientan precedentes para desarrollar herramientas tecnológicas que ayuden a combatir el cáncer de piel y otras enfermedades, permitiendo una detección más temprana y un tratamiento oportuno, en beneficio de la salud pública.

## BIBLIOGRAFÍA

- Aggarwal, C. C. (2021). *Neural Networks and Deep Learning: A Textbook (2nd ed.)*. Cham, Suiza: Springer.
- Benavent Núñez, D., Colomer Mascaró, J., Quecedo Gutiérrez, L., Gol-Montserrat, J., & Del Llano Señarís, J. E. (2020). *INTELIGENCIA ARTIFICIAL Y DECISIONES CLÍNICAS: Cómo está cambiando el comportamiento del médico*. Obtenido de <https://fundaciongasparcasal.org/wp-content/uploads/2020/12/Decisiones-clinicas-e-inteligencia-artificial.pdf>
- Berthold, M. (2022). *Knime*. Obtenido de <https://www.knime.com/>
- Clinic, M. (5 de Diciembre de 2020). *MAYO CLINIC: Cáncer de la piel*. Obtenido de <https://www.mayoclinic.org/es-es/diseases-conditions/skin-cancer/symptoms-causes/syc-20377605>
- Eisenstein, J. (2019). *Natural Language Processing*. Cambridge, MA: MIT Press.
- Field, C. B. (2022). Climate Change 2022: Impacts, Adaptation, and Vulnerability. *Nature Climate Change*, 18(6), 457-462.
- Gandini, S. &. (2023). Melanoma Epidemiology and Prevention Strategies: UV Exposure and Risk Factors. *International Journal of Cancer*, 152(10), 2501-2510.
- Green, A. C. (2023). Skin Cancer and UV Radiation: Epidemiology and Public Health Implications. *British Journal of Dermatology*, 188(5), 1047-1054.
- Haenssle, H. (2021). *FotoFinder AI*. Obtenido de <https://www.fotofinder.de/es/tecnologia/inteligencia-artificial>
- Hoogstra, R. (Mayo de 2019). *Inteligencia artificial en dermatología*. Obtenido de <https://doctorhoogstra.com/wiki/inteligencia-artificial-en-dermatologia/>
- ISIC, S. (10 de Agosto de 2020). *SIIM-ISIC Melanoma Classification*. Obtenido de Kaggle: <https://www.kaggle.com/competitions/siim-isic-melanoma-classification/data>
- Kelleher, J. D. (2019). *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*. Cambridge, MA: MIT Press.
- Lambin, P. L. (2020). Radiomics: The Bridge between Medical Imaging and Personalized Medicine. *Nature Reviews Clinical Oncology*, 17(12), 791-804.
- Lee, P. (2022). *Rapidminer*. Obtenido de <https://rapidminer.com/>
- Li, Y. Y. (2021). *Deep Learning for Biomedical Applications: Techniques and Applications*. Cambridge, MA: Academic Press.
- Marks, R. &. (2021). Actinic Keratosis: Diagnosis, Treatment, and Prevention. *Journal of Skin Cancer*, 9(3), 295-302.

- Márquez Díaz, J. E. (2020). Deep Artificial Vision Applied to the Early Identification of Non-Melanoma Cancer and Actinic Keratosis. *Computación y Sistemas*, 24(2), 751–766. doi:<https://doi.org/10.13053/CyS-24-2-2901>
- Marr, B. (2020). *Big Data in Practice: How 45 Successful Companies Used Big Data Analytics to Deliver Extraordinary Results*. Hoboken, NJ: Wiley.
- Martorell, A., Martin-Gorgojo, A., Ríos-Viñuela, E., Rueda-Carneroa, J., Alfagedmed, F., & Tabernere, R. (3 de Agosto de 2021). *Inteligencia artificial en dermatología: ¿amenaza u oportunidad?* Obtenido de <https://www.sciencedirect.com/science/article/pii/S0001731021002957#!>
- McKenzie, R. L. (2023). Ozone depletion and the UV Radiation Climate: Effects and Public Health Implications. *Photochemical & Photobiological Sciences*, 22(4), 123-129.
- Muñoz Rodríguez, E. E., Moreno Espinosa, J. A., & Torres Garzón, D. M. (Diciembre de 2020). *USO DE LA CIENCIA DE DATOS Y HERRAMIENTAS TECNOLÓGICAS*. Obtenido de <https://repository.usergioarboleda.edu.co/bitstream/handle/11232/1718/Inteligencia%20artificial%20en%20salud.pdf?sequence=1&isAllowed=y>
- Narang, J. W. (2023). Skin Cancer: Pathogenesis and Clinical Manifestations. *British Journal of Dermatology*, 188(6), e45-e55.
- Pérez Ortiz de Landaluce, M. (2021). Clasificación de imágenes mediante algoritmos de Deep Learning: Mascarillas de COVID-19. *Escuela Técnica Superior de Ingeniería, Universidad de Sevilla*, 6-27. Obtenido de <https://biblus.us.es/bibing/proyectos/abreproy/93772/fichero/TFG-3772+P%C3%89REZ+ORTIZ+DE+LANDALUCE%2C+MARTA.pdf>
- Poole, D. &. (2020). *Artificial Intelligence: Foundations of Computational Agents*. Cambridge: Cambridge University Press.
- Reales Castro, F. A. (2017). *Sistema de clasificación de imágenes de lunares utilizando visión artificial*. Obtenido de <http://hdl.handle.net/10554/38692>
- Revilla T., L. (2021). *Situación de cáncer en el Perú*. Obtenido de Ministerio de Salud: <http://www.dge.gob.pe/portal/docs/tools/teleconferencia/2021/SE252021/03.pdf>
- Ringa, Y. (16 de Agosto de 2021). *Redes Neuronales Convolucionales - Clasificación avanzada de imágenes con IA / ML (CNN)*. Obtenido de Youtube: Ringa Tech: <https://www.youtube.com/watch?v=4sWhhQwHqug>
- Ringa, Y. (17 de Julio de 2021). *Tu primer clasificador de imágenes con Python y Tensorflow*. Obtenido de Youtube: Ringa Tech: <https://www.youtube.com/watch?v=j6eGHROLKP8>
- Rodríguez, J. J. (23 de Septiembre de 2019). La inteligencia artificial en dermatología. (P. R. al., Ed.) *Med Cutan Iber Lat Am*, 89-90. Obtenido de La inteligencia artificial en dermatología: <https://www.medigraphic.com/pdfs/cutanea/mc-2019/mc192a.pdf>

- Sánchez-Saldaña, L. A. (2019). *La inteligencia artificial cambiará la dermatología*. Obtenido de [https://www.dermatologiaperuana.pe/assets/uploads/revista\\_YUh5\\_00\\_Editorial\\_29-4.pdf](https://www.dermatologiaperuana.pe/assets/uploads/revista_YUh5_00_Editorial_29-4.pdf)
- Sculley, D. (2022). *Kaggle*. Obtenido de <https://www.kaggle.com/>
- Sebastian, P. (Julio de 2020). *Pronóstico en melanoma: presentando mi tesis doctoral durante el COVID*. Obtenido de <https://www.sebastianpodlipnik.com/pronostico-en-melanoma/>
- Sempértegui-Ruiz, S. R.-C.-A.-P.-M.-G.-O.-M. (2020). Conocimientos y hábitos de exposición solar en comerciantes ambulantes de un mercado de Chiclayo, Perú. *Revista de la Facultad de Medicina Humana*, 20(2), 261-267. doi:10.25176/RFMH.v20i2.2899
- Skin, C. (2022). *Líder en la Lucha Contra el Cáncer de Piel*. Obtenido de Skin Cancer Foundation: <https://cancerdepiel.org/quienes-somos/lider-en-la-lucha-contra-el-cancer-de-piel>
- Stolte, C., Hanrahan, P., & Chabot, C. (2022). *Tableau*. Obtenido de <https://www.tableau.com/>
- Weinstock, M. A. (2022). Dysplastic Nevus as a Precursor to Melanoma: Evidence and Controversies. *Journal of the American Academy of Dermatology*, 86(2), 235-245.
- Zimmermann, H.-J. (2021). *Fuzzy Set Theory – and Its Applications*. Cham, Suiza: Springer.