**TABLA DE CONTENIDO**

[I. INTRODUCCIÓN 5](#_Toc170678961)

[II. RESUMEN 7](#_Toc170678962)

[III. DISEÑO TEÓRICO 8](#_Toc170678963)

[3.1. Antecedentes 8](#_Toc170678964)

[3.2. Bases Teórica 11](#_Toc170678965)

[*3.2.1.* *Agentes Inteligentes.* 11](#_Toc170678966)

[*3.2.2.* *Sistema inteligente.* 11](#_Toc170678967)

[*3.2.3.* *Deep Learning.* 11](#_Toc170678968)

[*3.2.4.* *Modelos de Deep Learning* 12](#_Toc170678969)

[IV. DISEÑO METODOLÓGICO 15](#_Toc170678970)

[4.1. Tipificación de la investigación 15](#_Toc170678971)

[4.2. Tabla de operacionalización de variable 15](#_Toc170678972)

[4.3. Población y muestra 17](#_Toc170678973)

[4.4. Técnicas, instrumentos, equipos y materiales 18](#_Toc170678974)

[V. RESULTADOS 20](#_Toc170678975)

[5.1. Análisis del conjunto de Imágenes 20](#_Toc170678976)

[*5.1.1.* *Descripción del conjunto de datos SICAPv2* 20](#_Toc170678977)

[5.2. Procesamiento de Datos 23](#_Toc170678978)

[5.3. Seleccionar un modelo óptimo de Deep Learning 25](#_Toc170678979)

[5.3.1. Descripción de los modelos utilizados 25](#_Toc170678980)

[5.3.2. Resultados de los modelos aplicados 30](#_Toc170678981)

[5.3.3. Elección del modelo 30](#_Toc170678982)

[5.4. Desarrollar el sistema inteligente 32](#_Toc170678983)

[5.4.1. Requisitos y preparación 32](#_Toc170678984)

[5.4.2. Preparación del entorno de trabajo 32](#_Toc170678985)

[*5.4.3.* *Desarrollo del sistema* 33](#_Toc170678986)

[VI. DISCUSIÓN DE RESULTADOS 48](#_Toc170678987)

[VII. CONCLUSIONES 50](#_Toc170678988)

[VIII. RECOMENDACIONES 52](#_Toc170678989)

[IX. BIBLIOGRAFÍA 53](#_Toc170678990)

**DEDICATORIA**

**------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

Jahir Santos Santamaria Santisteban

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Luis Felipe Siesquen Valdivia

**AGRADECIMIENTO**

**------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

Jahir Santos Santamaria Santisteban

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Luis Felipe Siesquen Valdivia

# INTRODUCCIÓN

El cáncer representa una de las mayores preocupaciones de salud a nivel mundial, siendo responsable de millones de defunciones cada año. Entre los tipos de cáncer más prevalentes se encuentra el cáncer de próstata, que afecta principalmente a la población masculina y puede tener un impacto significativo en la calidad de vida y la supervivencia de los pacientes. En el contexto peruano, el cáncer es la principal causa de mortalidad por grupo de enfermedad para el género masculino, lo que genera un importante impacto económico y social debido a su diagnóstico tardío y las limitaciones en el acceso a cuidados integrales.

Con el objetivo de abordar esta problemática, el Ministerio de Salud del Perú ha formulado el Plan Nacional de Cuidados Integrales del Cáncer 2020-2024, con el fin de mejorar el acceso a la atención del cáncer y reducir su mortalidad en el país. Sin embargo, la detección temprana sigue siendo un desafío importante en el manejo de esta enfermedad.

En este contexto, surge la necesidad de implementar tecnologías innovadoras que puedan mejorar la detección precoz del cáncer de próstata. La presente investigación se centra en el desarrollo de un sistema de inteligencia artificial basado en redes neuronales convolucionales, con el fin de predecir el diagnóstico de cáncer de próstata de manera eficiente.

El presente estudio aborda la situación problemática del cáncer de próstata, destacando su impacto a nivel mundial y en el Perú. Se destaca la importancia del Plan Nacional de Cuidados Integrales del Cáncer 2020-2024 como estrategia para mejorar la atención y reducir la mortalidad por esta enfermedad. Además, se resalta la relevancia social y clínica de implementar un sistema de inteligencia artificial para la detección temprana del cáncer de próstata.

El problema central de esta investigación se formula en torno a la eficiencia de un sistema inteligente basado en Deep Learning para predecir el diagnóstico de cáncer de próstata. Se plantea como hipótesis que la evaluación del rendimiento determinará la eficiencia de este sistema en el diagnóstico de la enfermedad.

Los objetivos de este estudio comprenden el desarrollo de un sistema inteligente basado en Deep Learning para predecir el diagnóstico de cáncer de próstata. Esto incluye el análisis y procesamiento de imágenes de cáncer de próstata, la elaboración de un modelo óptimo de Deep Learning, el desarrollo del sistema inteligente y la evaluación de su rendimiento en la detección de la enfermedad.

En resumen, esta investigación busca contribuir a mejorar la detección temprana del cáncer de próstata mediante la implementación de tecnologías de inteligencia artificial, lo que podría tener un impacto significativo en la atención y supervivencia de los pacientes afectados por esta enfermedad.

# RESUMEN

La presente investigación se desarrolló considerando la base de datos del SICAPv2 que contiene imágenes de histología de próstata con anotaciones de puntuaciones de Gleason y calificaciones a nivel de ruta, en la cual las imágenes estaban clasificadas por pacientes de acuerdo con sus exámenes realizados.

Se construyó un modelo para la detección de cáncer de próstata, basándose en el aprendizaje profundo aplicando redes neuronales convolucionales con procesamiento de imágenes utilizando un total de 13,827 imágenes divididas en 2 estados, con y sin cáncer.

El modelo se ejecutó obteniendo una tasa de acierto de 95.1%, con una función de pérdida de 0.176%, se comprobó con una muestra de 50 imágenes aleatorias existentes extraídas del repositorio las cuales hacen referencia a 3 pacientes registrados. Finalmente, la aplicación web y el modelo de red neuronal convolucional funcionaron correctamente.

Palabras claves: Aprendizaje Profundo, cáncer de próstata, procesamiento de imágenes, aplicación web.

# DISEÑO TEÓRICO

## Antecedentes

Guajin, Jeremy Yuen-Chun, & Kup-Sze, (2018) Desarrollaron un sistema para diagnosticar cáncer de próstata a una población china. Obtuvieron una base de datos de biopsias de próstata guiadas por ecografía transrectal (TRUS) en un hospital de Hong Kong la cual consta de 1625 registros de pacientes chinos. Se usaron cuatro métodos de aprendizaje automático Support Vector Machine (SVM), Least Squares Support Vector Machine (LS-SVM), Artificial Neural Network (ANN) y Random Forest (RF) en el cual ANN destacó logrando la más alta precisión de 95.27%.

Faez (2018) utilizó datos recopilados del hospital Imam Reza (Teherán). Este conjunto de datos es simplemente acerca de quiénes dieron positivo los resultados de sus pruebas y si sufrían de cáncer de próstata. Aplicaron el uso de Deep Learning y tres ANN para mejorar el diagnóstico respectivo, Deep Learning logró una precisión de clasificación del 86,3 % siendo está la más cercana al resultado deseado.

Lal, y otros (2019) desarrollaron una aplicación para asociar características morfológicas del cáncer de próstata con imágenes por resonancia magnética (MRI). Utilizaron redes bayesianas donde nodos y arcos muestran una relación de causa y efecto a través de gráficos acíclicos dirigidos (DAG). Gracias a la asociación entre Área y equidiámetro la aplicación pudo obtener una probabilidad de 98.94%.

Forero Cuellar (2019) utilizó un algoritmo de aprendizaje en datos “Prueba y puntuación" del programa informático para realizar minería de datos y análisis predictivo denominado "Orange Data Mining Fruit and Fun", el proceso de diagnóstico se realiza por medio del médico experto que evalúa las variables, apoyándose en las imágenes de resonancia magnética y otras técnicas tradicionales y muchas veces el diagnóstico es difícil y demorado, la novedad es que se puede tener un diagnóstico más rápido y acertado como complemento del diagnóstico clínico. Finalmente se encontró que la regresión logística y la red neuronal tienen un rendimiento mejor que bosques aleatorios y árbol de decisión, para la clasificación de las categorías de evaluación PI-RADS con una precisión del 83,1% para regresión logística y 79,5% para el árbol de decisión.

Yasser Mk, Fahima A., & Abdelrahman I. (2019) tuvieron como objetivo predecir fármacos y posibles candidatos a fármacos que interactúen con los receptores de adenosina (detiene el crecimiento de células tumorales incluido el carcinoma de próstata) para ellos se utilizó el algoritmo Random Forest - colección de árboles de decisión. DT - árbol de decisión para el conjunto de datos se obtuvieron de un estudio realizado por Saad et al., era una combinación de dos conjuntos de datos donde el primer conjunto de datos era un conjunto de datos de medicamentos extraído de la central de medicamentos la cuál contribuía a una simulación que se basa en información de la molécula, como el número de átomos y los enlaces entre estos átomos. Luego, esta información se utiliza para generar huellas dactilares codificadas (bits binarios); los resultados mostraron que RF obtuvo la mayor precisión en la clasificación de fármacos que interactúan con el receptor de adenosina A2a 75,09%.

Mohamed , Mansoor , Mansour , Hari Kiran, & P. (2020) utilizaron la técnica de clasificación de regresión logística modificada (MLR) la cual se realiza en la identificación del cáncer de próstata a partir de un conjunto de datos de Zhou W. Para las pruebas se utilizaron Imágenes obtenidas por la resonancia magnética (IRM) en específicos multiparamétricos (mpMRI) y en los resultados generales se vió reflejado que el nivel de sensibilidad (Se) en un 96% y Especificidad (Sp) en un 98 %, lo que mejora la precisión del clasificador en al menos un 3 % en comparación con los métodos existentes.

Carneiro Neto (2020) utilizó una arquitectura de red neuronal profunda, XmasNet, para realizar dos grandes experimentos sobre la clasificación de lesiones las imágenes se obtuvieron de un conjunto de datos IMPROD (Improved Prostate Cancer Diagnosis - Combination of Magnetic Resonance Imaging and Biomarkers) se originó a partir de un ensayo clínico realizado por el Hospital Universitario de Turku y la Universidad de Turku. El ensayo incluyó a 175 pacientes entre 40 y 85 años con sospechas de cáncer de próstata respaldadas por resultados de detección. se obtuvo por resultado que la clasificación binaria del significado clínico de las lesiones y la detección y segmentación de la próstata ya consiguen los resultados deseados 0,870 AUC y 0,915 dice score respectivamente.

## Bases Teórica

### *Agentes Inteligentes.*

Son sistemas que perciben su entorno, razonan y toman acciones de tal forma que maximizan sus oportunidades de éxito. Este paradigma permite que los investigadores estudien problemas complejos y busquen soluciones que son al mismo tiempo útiles y verificables. La tecnología de los agentes inteligentes conforma la base de una nueva generación de sistemas computacionales. Las aplicaciones incluyen sistemas desarrollados para búsqueda de información masivamente distribuida en Internet, sistemas de información móviles, sistemas de workflow inteligentes, e infraestructura de información de apoyo a las operaciones y decisiones corporativas (Gamboa, 2014).

### *Sistema inteligente.*

Se denomina sistema inteligente a una entidad capaz de percibir, razonar, aprender, adaptarse, tomar decisiones y actuar racionalmente para satisfacer sus metas, en un determinado entorno. (Gamboa, 2014).

Son parte de las ciencias de computación, que cubren una serie de tópicos teóricos – experimentales que sirven de base para la ingeniería de aplicaciones inteligentes.

### *Deep Learning.*

Deep Learning es un subconjunto de [machine Learning](https://www.ibm.com/co-es/analytics/machine-learning) (que a su vez es parte de la [inteligencia artificial](https://www.ibm.com/co-es/cloud/learn/what-is-artificial-intelligence)) donde las redes neuronales, algoritmos inspirados en cómo funciona el cerebro humano, aprenden de grandes cantidades de datos.

Los algoritmos de Deep Learning realizan una tarea repetitiva que ayuda a mejorar de manera gradual el resultado a través de ‘’deep layers’’ lo que permite el aprendizaje progresivo. Este proceso forma parte de una familia más amplia de métodos de machine Learning basados en redes neuronales (IBM, 2020).

### *Modelos de Deep Learning*

Según Sandoval, (2021)**:**

**Modelos lineales:** Estos tratan de encontrar una línea que se “ajuste” bien a la nube de puntos que se disponen. Aquí destacan desde modelos muy conocidos y usados como la regresión lineal (también conocida como la regresión de mínimos cuadrados), la logística (adaptación de la lineal a problemas de clasificación.

**Modelos de árbol:** Son modelos precisos, estables y más sencillos de interpretar básicamente porque construyen unas reglas de decisión que se pueden representar como un árbol.

**Redes neuronales:** Las redes artificiales de neuronas tratan, en cierto modo, de replicar el comportamiento del cerebro, donde tenemos millones de neuronas que se interconectan en red para enviarse mensajes unas a otras.

Según Graph, (2020):

**Algoritmos de Deep Learning**.

Existen diferentes tipos de algoritmos y estos son aplicables dependiendo del tipo de Deep Learning en el cual el algoritmo vaya a funcionar.

**Algoritmos de regresión**

Los algoritmos de regresión se aplican en modelos de Deep Learning que buscan estimar y determinar la existencia de relaciones entre variables que forman parte del objeto de estudio.

**Algoritmos Bayesianos**

Los algoritmos de Deep Learning conocidos como bayesianos reciben esta denominación porque están basados en el teorema de bayes. Dentro de su funcionamiento realizan clasificaciones de cada valor como independiente de cualquier otro. Esto permite predecir con mucha efectividad una clase o categoría dentro de un conjunto dado de características mediante modelos probabilísticos.

**Algoritmos de agrupación**

Dentro de los algoritmos de Deep Learning que son implementados para aprendizaje no supervisado se encuentran los algoritmos de agrupación. Gracias a ellos podemos establecer categorías dentro datos no etiquetados, es decir, podemos ordenar datos que pertenecen a grupos indefinidos.

**Algoritmos de árbol de decisión**

Los árboles de decisión son implementados como algoritmos de Deep Learning debido a que, al ser similares a un diagrama de flujo, utilizan un método determinado de cruce para representar los posibles resultados que ocasiona la toma de una decisión.

**Algoritmos de redes neuronales**

Dentro de los tipos de algoritmos de Deep Learning más interesantes se encuentran los que conocemos como redes neuronales. Estos comprenden unidades dispuestas en capas. Cada una de estas capas posee una conexión con las capas anexas. Su funcionamiento pretende emular el comportamiento de procesamiento de información del cerebro humano.

**Algoritmos de Aprendizaje Profundo**

Los algoritmos de aprendizaje profundo o de Deep Learning son aquellos que ejecutan datos a través de múltiples capas de redes neuronales. Estas pasan a la capa siguiente una representación simplificada de los datos analizados.

La variable dependiente e independiente fueron basadas por el libro “Para la Profundizando en la escala de puntuación de Gleason: Un sistema automático de extremo a extremo para histología de próstata, clasificación y detección de patrones cribiformes” (Silva, Colomer, Salesc, Molina, & Naranjo, 2020).

# DISEÑO METODOLÓGICO

## Tipificación de la investigación

* Cuantitativa o cualitativa: Cuantitativa
* Experimental o no experimental: No experimental
* Alcance de la investigación (exploratorio, descriptivo, correlacional o explicativo): Exploratorio
* Básica o aplicada: Aplicada

## Tabla de operacionalización de variable

**Tabla 1**

*Tabla operacional según Indicadores*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Variable | | Dimensión | Indicadores | Medida |
| Variable dependiente  Evaluación del rendimiento. | Performance | | VP  VN  FP  FN | Porcentaje |
| Exactitud | Porcentaje |
| Variable independiente  Sistema inteligente basado en Deep Learning para el diagnóstico de cáncer de próstata. |  | | Usabilidad | |

Fuente: Elaboración propia

1. Verdaderos Positivos (TP): cuando la clase real del punto de datos era 1 (Verdadero) y la predicha es también 1 (Verdadero).
2. Verdaderos Negativos (TN): cuando la clase real del punto de datos fue 0 (Falso) y el pronosticado también es 0 (Falso).
3. Falsos Positivos (FP): cuando la clase real del punto de datos era 0 (False) y el pronosticado es 1 (True).
4. Falsos Negativos (FN): Cuando la clase real del punto de datos era 1 (Verdadero) y el valor predicho es 0 (Falso).

Se definen los siguientes indicadores:

**Exactitud.** La exactitud o accuracy es la proporción de resultados verdaderos (tanto verdaderos positivos como verdaderos negativos) en la población. La exactitud es definida a partir de la fórmula siguiente.

Interfaz de usuario gráfica, Texto, Aplicación  Descripción generada automáticamente

Interfaz de usuario gráfica, Texto  Descripción generada automáticamente**Precisión**: el valor de la precisión se define como la proporción de verdaderos positivos contra todos los resultados positivos, y es definida por la fórmula.

Interfaz de usuario gráfica, Texto, Aplicación  Descripción generada automáticamente**La sensibilidad.** La sensibilidad o recall mide la proporción de “verdaderos positivos” que son correctamente identificados como tales.

**La especificidad** (Especificity). La especificidad se refiere a la capacidad de la prueba para identificar los resultados negativos. Mide la proporción de “verdaderos negativos” que se han identificado correctamente.

Interfaz de usuario gráfica, Texto, Aplicación  Descripción generada automáticamente

Los indicadores fueron basados por el libro “Importancia del cálculo de la sensibilidad, la especificidad y otros parámetros estadísticos en el uso de las pruebas del diagnóstico clínico y de laboratorio” (Vizcaíno-Salazar, 2017).

La usabilidad se basará bajo la ISO25010 (iso25000, 2022).

## Población y muestra

La muestra se utilizará para el entrenamiento del sistema son 13,827 imágenes, de las cuáles fueron agrupadas mostrando diferentes grados de clasificación Gleason extraídas de la base de datos de SICAPv2 (Imágenes de diapositivas completas de próstata con anotaciones de grados de Gleason), para ello se cuenta con la autorización de hacer uso de las imágenes.

La data de SICAPv2 cuenta con la interpretación de la imagen correspondiente a analizar indicando a que grado de Gleason pertenece, la data SICAPv2 fue extraída de la web de (Rodríguez, 2020).

La técnica de muestreo que se utilizó fue la de **Aleatorio estratificado ya que se escogió al azar de cada grado de la clasificación Gleason.**

Se tomarán en cuenta 5 estratos cada uno de ellos definidos por la clasificación de Gleason de los cuáles los grupos de grado 1 (células bien diferenciadas), grupo 2 (moderadamente diferenciadas) y grado 3, 4 ,5 (poco diferenciadas).

## Técnicas, instrumentos, equipos y materiales

Las técnicas para la recolección de datos que se utilizarán en la presente investigación son: análisis documentario y la observación para las pruebas realizadas.

* **Análisis documentario:** La información será obtenida de los diferentes documentos, artículos, libros, revistas, publicaciones, gráficos, etc. Los cuales presentan una serie de algoritmos, modelos y conjunto de imágenes.

Este análisis documental serpa utilizado para la búsqueda del conjunto de imágenes y para la selección de modelos de Deep Learning.

* **La observación:** La observación es un elemento fundamental de todo proceso de investigación; en ella se apoya el investigador para obtener el mayor número de datos, en este caso se aplicará para la interpretación de las imágenes.

La observación será utilizada en la etapa de evaluación del rendimiento del sistema inteligente, registrando los resultados en una ficha de observación.

Los instrumentos, que se utilizará para verificar la fiabilidad de los resultados del sistema web.

* **Estadístico:** Las herramientas estadísticas que se utilizarán para la determinación de la hipótesis
* **Fichas de observación:** Se utilizarán progresivamente de acuerdo con las mejoras que se vayan obteniendo en el proceso de desarrollo.

Los materiales que se utilizarán son 2 computadoras portátiles; una core i7 3.20 GHz, 16 GB RAM y una core i7 2.80 GHz, 16 GB RAM respectivamente utilizando Python como lenguaje de programación con el framework Django.

Con respecto a los resultados serán guardados en base de datos MySQL y se utilizará un sistema de control de versiones para el proyecto (Git y GitHub).

# RESULTADOS

## Análisis del conjunto de Imágenes

### *Descripción del conjunto de datos SICAPv2*

El conjunto total de datos SICAPv2 (Silva-Rodríguez, 2020), contiene imágenes histológicas completas de próstata, con anotaciones de puntuaciones globales de Gleason y grados de Gleason a nivel de patrón. Está diseñado para estudios de clasificación y detección de patrones en histología prostática, específicamente para la detección de patrones cribiformes y la clasificación automática según la escala de Gleason.

La base de datos presentada en este artículo, SICAPv2, es, hasta donde saben los autores, la colección pública más grande de biopsias de próstata H&E con anotaciones a nivel local de grados de Gleason. SICAPv2 es una extensión de la base de datos introducida en [17] y estará disponible públicamente después de la publicación de este artículo.

Contiene 13,827 imágenes que son divididas entre los 2 estados, entre ellas tenemos: con cáncer de próstata y sin cáncer de próstata. El conjunto de imágenes está dividido en 2 carpetas llamadas train y val. Train contiene las imágenes para entrenamiento las cuales son un total de 10,957 imágenes y val contiene las imágenes para validación con un total de 2870 imágenes.

**Tabla 2**

*Distribución de imágenes en dataset*

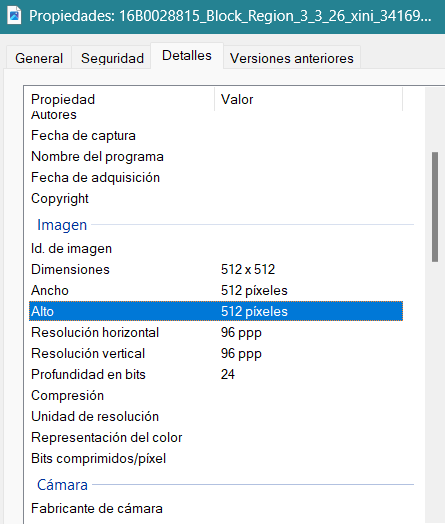
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dataset | Entrenamiento(train) | Validación(val) | Total | Porcentaje |
| ConCancer | 7,367 | 2,043 | 9,410 | 68% |
| SinCancer | 3,590 | 827 | 4,417 | 32% |

Fuente: Elaboración propia.

Las características principales de cada imagen del dataset son dimensiones de 512 x 512 píxeles, una resolución horizontal y vertical de 96 ppp (píxeles por pulgada), y una profundidad de color de 24 bits. Estos datos indican que la imagen tiene una resolución moderada y una profundidad de color estándar.

**Figura 1**

*Propiedad de imagen de dataset SICAPv2.*

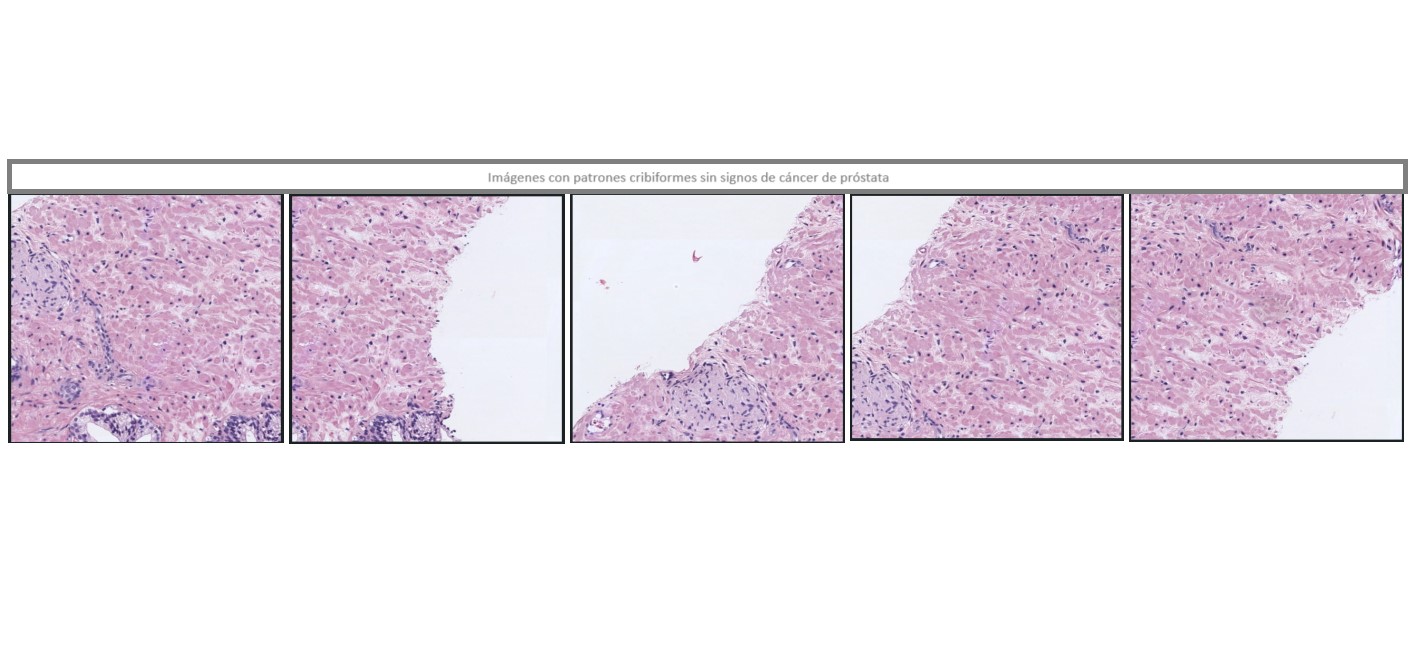


Fuente: Elaboración Propia

Se muestran 5 imágenes sin cáncer de próstata las cuales tienen una consistencia uniforme en las células, sin signos de crecimiento irregular, no se observan áreas de células agrupadas de manera anormal o formando patrones cribiformes.

**Figura 2**

*Imágenes con patrones cribiformes, no contiene signos de cáncer de próstata*

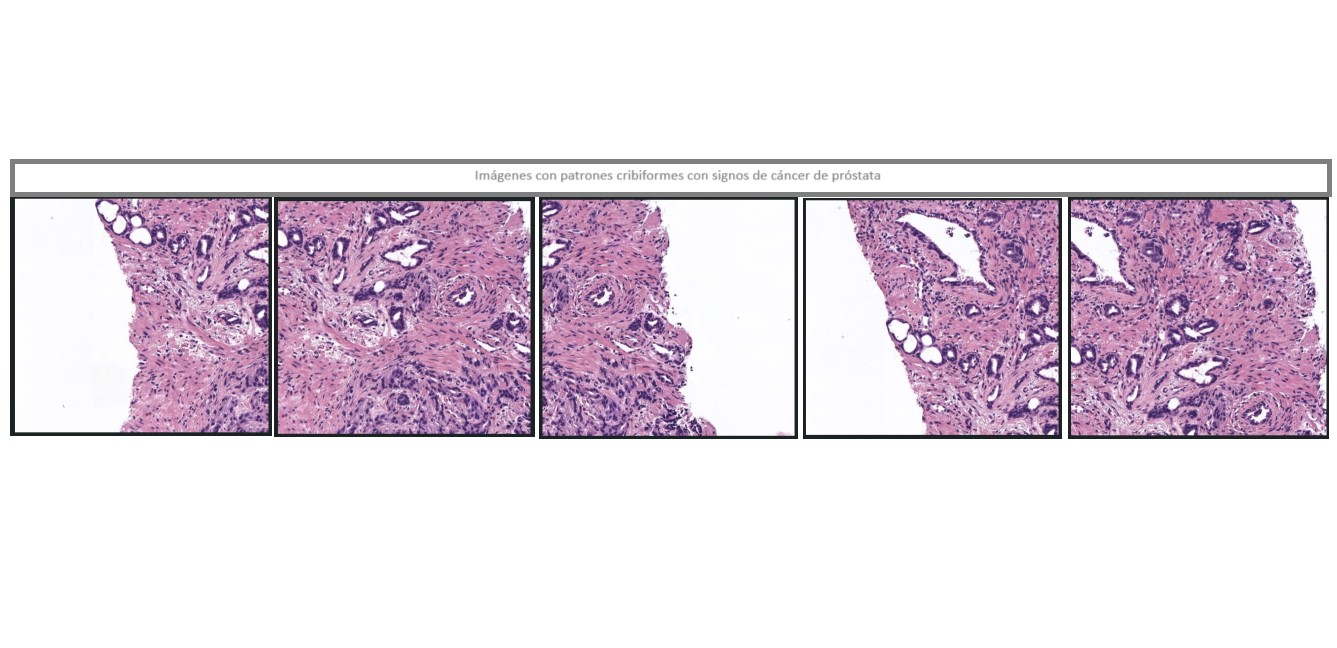


Fuente: SICAPV2

Se presentan 5 imágenes con cáncer de próstata, en las cuales se observan características distintivas. Las células muestran un crecimiento desordenado y agrupaciones irregulares en forma de criba, el cual es indicativo de la enfermedad. Además, se observan núcleos agrandados y patrones cribiformes, señales claras de malignidad evidenciando la gravedad del cáncer.

**Figura 3**

*Imágenes con patrones cribiformes contiene signos de cáncer de próstata*



Fuente: SICAPV2

## Procesamiento de Datos

Los datos procesados contienen una gran variedad de imágenes de estudios de cáncer de próstata para la detección de dicha enfermedad.

El conjunto de datos se divide en dos partes: entrenamiento y validación, ambas partes contiene 2 subcarpetas las cuales están divididas en imágenes con cáncer y sin cáncer respectivamente.

**Definición de variables y parámetros**

Primero, se define la ruta donde se encuentran las imágenes de entrenamiento y las categorías que se van a clasificar: "ConCancer" y "SinCancer". Además, se establece el tamaño del lote (batch size) para el entrenamiento del modelo y las dimensiones a las que se redimensionarán las imágenes, que en este caso es de 512x512 píxeles, en nuestro caso las imágenes originales no se verán afectadas ya que se trabajará con su tamaño original.

**Carga y procesamiento de imágenes**

Se crea una lista vacía que almacenará las imágenes que han sido cargados para ser analizadas junto con sus etiquetas. La función se encarga de recorrer cada categoría y procesar todas las imágenes dentro de cada categoría. Para cada imagen, se realiza lo siguiente:

* Se lee la imagen desde el disco.
* Se redimensiona la imagen a las dimensiones especificadas (512x512 píxeles).
* Se almacena la imagen redimensionada y su etiqueta (0 para "ConCancer" y 1 para "SinCancer") en la lista.

Esta función se ejecuta para cargar y procesar todas las imágenes del conjunto de datos.

Luego de procesar las imágenes, se le asignaron las etiquetas correspondientes como se ve en las siguientes imágenes:

**Figura 4**

*Procesamiento de una imagen con cáncer*

**IMAGEN ORIGINAL IMAGEN PROCESADA**

**Un dibujo de una persona

Descripción generada automáticamente con confianza bajaGráfico, Gráfico de dispersión

Descripción generada automáticamente**

Fuente: Elaboración propia

**Figura 5**

*Procesamiento de una imagen sin cáncer*

**IMAGEN ORIGINAL IMAGEN PROCESADA**

Gráfico

Descripción generada automáticamente con confianza mediaGráfico

Descripción generada automáticamente con confianza media

Fuente: Elaboración propia

En el procesamiento la imagen original no es redimensionada ya que el parámetro inicial es de 512x512 y se posterior a ello se procede a clasificar las imágenes con una etiqueta ya sea “SinCancer” o “ConCancer”.

## Seleccionar un modelo óptimo de Deep Learning

### Descripción de los modelos utilizados

Se evaluaron 3 algoritmos de deep Learning que son VGG16, VGG19 y ResNet50 que se describen a continuación:

#### VGG-16

Quispe, (2020) esta arquitectura está constituida de 13 capas convolucionales. Cada grupo es seguido de una capa de max pooling. A estas le siguen 3 capas totalmente conectadas, de ahí su nombre incluye 16, y finalmente un clasificador softmax. Esta red se caracteriza por contar con varios parámetros para entrenar, lo que puede hacer que su entrenamiento dure más que el resto de las redes. Asimismo, tiene la ventaja de contar con pesos pre entrenados disponibles del dataset de Imagenet, que podrían ser utilizados para inicializar el entrenamiento de la red propuesta como comenta.

**Figura 6**

*Arquitectura de VGG-16 separado por 16 capas convolucionales.*

Diagrama

Descripción generada automáticamente

Fuente: (Quispe, 2020)

**Figura 7**

*Modelo lógico de Arquitectura VGG-16 con 13 capas convolucionales y 3 densas*

Tabla, Escala de tiempo

Descripción generada automáticamente

Fuente: (Quispe, 2020)

#### VGG-19

Jian, Jia, Shaozhong, & Bilong, (2020) este modelo tiene seis estructuras principales, cada una de las cuales se compone principalmente de múltiples conexiones capas convolucionales y capas completamente conectadas. El tamaño del núcleo convolucional es 3\*3, y el tamaño de entrada es 224\*224\*3, el número de capas generalmente se concentra en 16 y 19.

**Figura 8**

*Modelo de red VGG-19*

Escala de tiempo

Descripción generada automáticamente

Fuente: (Jian, Jia, Shaozhong, & Bilong, 2020)

**Figura 9**

*Marco de entrenamiento del modelo de detección de uso de máscara.*

Diagrama

Descripción generada automáticamente

Fuente: (Jian, Jia, Shaozhong, & Bilong, 2020)

Utiliza una estructura alterna de múltiples capas convolucionales y capas de activación no lineales, que es mejor que una sola convolución. La estructura de capas puede extraer mejor las características de la imagen, usar Maxpooling para reducir la resolución y modificar la unidad lineal (ReLU) como función de activación, es decir, seleccione el valor más grande en el área de la imagen como valor conjunto del área. La capa de reducción de resolución se utiliza principalmente para mejorar la capacidad anti-distorsión de la red a la imagen, manteniendo las características principales de la muestra y reduciendo el número de parámetros.

#### ResNet50

Nishant & Shrivastava, (2021) ResNet-50 es un modelo prediseñado que ha sido entrenado en el conjunto de datos ImageNet para identificar diferentes imágenes de 1000 clases. El tamaño de la entrada de imagen de la red neuronal es de 224 x 224, Los pesos previamente entrenados de ImageNet se proporcionaron como pesos iniciales para la red neuronal profunda propuesta. Las capas residuales presentes en ResNet50 juega un papel importante para transferir grandes valores de gradiente a sus capas adyacentes anteriores.

**Figura 10**

*Arquitectura de ResNet50*

Gráfico, Diagrama, Gráfico de cajas y bigotes

Descripción generada automáticamente

Fuente: (Qingge, He, Yankui, & Jie, 2019)

Gracias a esta capa, el modelo puede extraer eficazmente patrones complejos y relevantes y resolver los problemas. problema de gradiente de desaparición. En nuestra configuración experimental, todas las capas del modelo previo al entrenamiento se mantienen abierto para aprender nuevas características de las imágenes de biopsia. Las matrices de características, adquiridas de capas CNN, se suministraron a la capa FC ajustada, donde la función sigmoidea se usó en la capa de salida.

### Resultados de los modelos aplicados

#### *Resultado del modelo VGG16*

A continuación, se muestran los resultados del siguiente modelo considerando los resultados finales de loss y accuracy respectivamente:

**Figura 11**

*Resultado modelo VGG-16Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente*

Tabla

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

Fuente: Elaboración propia

Según los resultados luego de entrenar el modelo con el algoritmo VGG16 podemos observar la precisión de los datos de entrenamiento se mantiene durante las últimas épocas con un resultado de 98.32 %, mientras que los datos de precisión de validación se mantienen y finalizan con un resultado de 90.10%.

La función de pérdida de los datos de entrenamiento y los datos de pérdida de validación en las últimas épocas muestra resultados constantes poco significativo por lo cual se consideró finalizar la ejecución.

#### *Resultado del modelo VGG19*

**Figura 12**

*Resultado modelo VGG-19*

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

Fuente: Elaboración propia

Según los resultados luego de entrenar el modelo con el algoritmo VGG19 podemos observar la precisión de los datos de entrenamiento se mantiene durante las últimas épocas con un resultado de 98.22%, mientras que los datos de precisión de validación se mantienen y finalizan con un resultado de 94,11%.

La función de pérdida de los datos de entrenamiento disminuye linealmente hasta llegar a 0.18% lo cual es un resultado favorable, mientras para los datos de pérdida de validación muestra resultados constantes por lo cual se consideró finalizar la ejecución.

#### *Resultado del modelo ResNet50*

**Figura 13**

Resultado modelo ResNet50

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

*Fuente: Elaboración propia*

Según los resultados luego de entrenar el modelo con el algoritmo ResNet90 podemos observar la precisión de los datos de entrenamiento se mantiene durante las últimas épocas con un resultado de 98.97%, mientras que los datos de precisión de validación se mantienen y finalizan con un resultado de 93,33%.

La función de pérdida de los datos de entrenamiento disminuye linealmente hasta llegar a 0.16% lo cual es un resultado favorable, mientras para los datos de pérdida de validación muestra resultados constantes por lo cual se consideró finalizar la ejecución.

### Elección del modelo

Después de entrenar el modelo utilizando el algoritmo VGG19, observamos que la precisión de los datos de entrenamiento se mantiene alta durante las últimas épocas con un resultado de 98.22%, mientras que la precisión de validación se mantiene y finaliza con un resultado de 94.11%. La función de pérdida de los datos de entrenamiento disminuye linealmente hasta llegar a 0.18%, lo cual es un resultado favorable. La pérdida de validación muestra resultados constantes, por lo cual se consideró adecuado finalizar la ejecución.

Probamos tres modelos diferentes y, después de evaluar sus desempeños, determinamos que el VGG19 era el mejor. La precisión y la pérdida de los otros modelos no lograron alcanzar ni mantener los niveles obtenidos con el VGG19. Específicamente, el VGG19 mostró una mejor estabilidad en las métricas de precisión y pérdida tanto en el conjunto de entrenamiento como en el de validación.

Comparado con los otros modelos, el VGG19 demostró una superior capacidad de generalización y adaptación a nuevos datos. Por estas razones, el VGG19 fue seleccionado como el mejor modelo para nuestro estudio, destacando por su alta precisión y baja pérdida, así como por su consistencia durante el proceso de entrenamiento y validación. Estos resultados indican que el VGG19 es más efectivo y fiable para la tarea de clasificación de imágenes en este contexto.

Como conclusión considerando la siguiente tabla adjunta:

**Tabla 3**

*Resultados algoritmos aplicados*

|  |  |  |  |
| --- | --- | --- | --- |
| Algoritmo | Cantidad de épocas | Accuracy | Loss |
| VGG16 | 20 | 90.10%. | 0.34% |
| VGG19 | 20 | 94,11% | 0.18% |
| ResNet50 | 20 | 93,33% | 0.16% |

Fuente: Elaboración propia

Podemos indicar que el modelo con mayor precisión y menor valor de pérdida se obtuvo aplicando el algoritmo VGG19 con una precisión de 84,11% dentro del rango de aceptación en la predicción para esta investigación.

## Desarrollar el sistema inteligente

### *Requisitos y preparación*

La presente investigación se desarrolló en la plataforma Anaconda con un lenguaje de programación de Python en su versión 3.10 y TensorFlow en su versión de 2.10 Para el entrenamiento de este modelo se adquirió una laptop con un procesador Ryzen 7 13th Gen y una tarjeta gráfica Nvidia GeForce RTXStudio para el procesamiento de imágenes.

### *Preparación del entorno de trabajo*

Para el desarrollo de la presente investigación se utilizó el lenguaje de programación Python, y la plataforma de desarrollo Anaconda con la siguiente estructura en nuestro data set alojado localmente en el explorador de archivos.

**Figura 14**

*Directorios de trabajo*

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Fuente: Elaboración propia*.*

**Figura 15**

*Estructura Principal*

Captura de pantalla de un celular

Descripción generada automáticamente

Fuente: Elaboración propia.

El directorio data del proyecto se dividió en 2 subdirectorios los cuales son train (entrenamiento) el val (validación), tienen data de la clasificación de Gleason que determina si el paciente tiene cáncer de próstata, siendo subdivididas en carpetas que tienen los siguientes nombres: ConCancer y SinCancer.

### *Desarrollo del sistema*

Se desarrolló un sistema inteligente en el lenguaje de programación Python en la versión 3.11.5 usando el framework Django. Este sistema ha sido diseñado para ayudar en el ámbito médico: la detección precoz y precisa del cáncer de próstata.

Para almacenar y gestionar los datos de manera eficiente, se ha optado por utilizar MySQL como sistema de gestión de bases de datos. Con MySQL garantizamos la integridad y la seguridad de los datos médicos sensibles.

El desarrollo consiste en la capacidad de realizar predicciones precisas utilizando un modelo de aprendizaje automático previamente entrenado. Este modelo ha sido entrenado con un amplio conjunto de datos de imágenes médicas de próstatas.

Cuando un usuario carga una imagen médica en el sistema, los datos son transmitidos al backend, donde se aplican técnicas de preprocesamiento para preparar la imagen para su análisis. Luego el modelo de aprendizaje automático entra en acción, realizando una evaluación detallada de la imagen y generando una predicción sobre la presencia o ausencia de cáncer de próstata.

Los resultados de la predicción son luego presentados al usuario a través de la interfaz de usuario y también pueden ser descargados mediante un reporte en formato PDF.

A continuación, se dará información más detallada acerca de la creación del sistema inteligente:

#### *Modelo de estructura para la base de datos*

La base de datos utilizada fue Mysql, para crear nuestras tablas lo hicimos desde Django con las migraciones al definir nuestros modelos se autogeneran como tablas en nuestra base de datos y esto hace que la creación no sea un proceso manual si no automático.

#### *Desarrollo para el backend para el software*

Para el desarrollo del backend se utilizó Python en la versión 3.11.5 usando el framework Django en la versión 4.0.4.

La implementación consistió principalmente en conectar nuestro backend con nuestro modelo de predicción que ha sido entrenado anteriormente además de conectar con el frontend para recibir los datos enviados. La utilizada fue la de Modelo - Vista - Controlador.

**Figura 16**

*Estructura de carpetas del backend*

Texto

Descripción generada automáticamente

Fuente: Elaboración propia

Esta estructura ayuda a mantener un código limpio y organizado, facilita la reutilización de código y promueve la escalabilidad de las aplicaciones.

Utilizamos Firebase Authentication que nos proporcionó una solución completa y segura para manejar el proceso de autenticación de usuarios en nuestra aplicación.

**Figura 17**

*Cuenta enlazada al proyecto*

Captura de pantalla de computadora

Descripción generada automáticamenteFuente: Elaboración propia

Finalmente se utilizó el API de la RENIEC para obtener los datos de un usuario a través de su Dni.

**Figura 18**

*Resultados enlace de cuenta*

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Fuente: Elaboración propia

El código del sistema fue almacena en GitHub y el enlace para acceder al repositorio es <https://github.com/FSIce1/prediction_cancer_prostata>, este es un repositorio público así que cualquiera puede acceder al código.

**Figura 19**

*Repositorio de Github*

Captura de pantalla de computadora

Descripción generada automáticamente

Fuente: Elaboración propia

#### *Funciones*

Se detalla mediante una tabla las funciones elaborados con una descripción explicando el funcionamiento de cada punto elaborado.

**Tabla 4**

*Descripción de funciones*

|  |  |
| --- | --- |
| Función | Descripción |
| crear\_usuario | Nos sirve para registrar el usuario para poder loguearse posteriormente. |
| Inicio | Redirige al sistema en caso de que el usuario se haya autentificado. |
| analisis\_imagen | Realiza la predicción a través del modelo. |
| cerrar\_sesion | Cierra la sesión actual en el sistema y redirige a la pestaña del login. |
| buscar\_por\_dni | Consulta el api de la RENIEC y nos devuelve los datos del usuario que se consultó con su dni. |
| resultado\_imagen | Nos lleva a la vista final que es mostrar el resultado. |
| realizar\_analisis | En esta función se consume el modelo de predicción. |
| historial\_analisis | Historial de todos los análisis realizados en el sistema. |
| pacientes | Registra los pacientes que fueron consultados por la RENIEC |
| generar\_pdf | Reporte final de cada análisis realizado |

Fuente: Elaboración propia

#### *Desarrollo para el frontend para el software*

Para desarrollar el frontend de la aplicación utilizamos HTML, Css y JavaScript para integrarlo con el motor de plantillas de Django, tenemos las vistas de Login, Registro, Análisis, Resultado, Historial y Pacientes.

**Figura 20**

*Menú Login del sistema*

Imagen que contiene interior, techo, persona, tabla

Descripción generada automáticamente

Fuente: Elaboración propia

Para la siguiente prueba se adjuntó una imagen extraída al azar del dataset en la opción “seleccionar archivo”, posterior a ello se digita el número de DNI de la persona que está realizando la consulta, la cual genera una búsqueda a través del API de reniec los datos personales del usuario:

**Figura 21**

*Análisis de resultados*

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Fuente: Elaboración propia

A continuación, el sistema te arroja los siguientes resultados:

**Figura 22**

*Resultados de la evaluación*

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Fuente: Elaboración propia

Adicional al resultado obtenido, el sistema nos permite exportarlo con un formato PDF para emitirle físicamente el resultado al paciente evaluado.

**Figura 23**

*Imagen que contiene Texto

Descripción generada automáticamenteExportable en formato PDF*

Fuente: Elaboración propia

El sistema también nos permite visualizar un reporte de todos los pacientes que realizaron una evaluación en el sistema con su resultado obtenido considerando parámetros como Dni, Nombre y Apellidos, Resultado, Porcentaje de precisión y el tiempo que demoró en realizarse la consulta.

**Figura 24**

*Resumen de análisis ejecutados en el sistema*

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

*Fuente: Elaboración propia*

#### *Evaluación del rendimiento del sistema inteligente*

Primero, cargamos el modelo que había sido previamente entrenado. Luego, se preparó los datos de prueba, que son imágenes organizadas en carpetas según sus categorías (Sin Cáncer y Con Cáncer). Se utilizó la herramienta TensorFlow para cargar estas imágenes y agruparlas en lotes, lo que facilita su procesamiento.

El modelo realizó predicciones sobre estos datos de prueba. Cada predicción es una suposición del modelo sobre a qué categoría pertenece cada imagen. Se convirtió estas predicciones a etiquetas que representan dichas categorías.

Luego, se comparó las predicciones del modelo con las etiquetas reales de las imágenes para evaluar su precisión. Se utilizó una matriz de confusión para ver cuántas predicciones fueron correctas y cuántas no.

También se generó un informe detallado que muestra métricas como precisión, recall y F1-score para cada categoría. Estas métricas nos ayudan a entender el funcionamiento el modelo en general y en cada categoría específica.

Por último, se identificaron las clases de las imágenes (Sin Cáncer y Con Cáncer), estas se utilizaron para analizar los resultados y hacer conclusiones sobre el rendimiento del modelo.

Explicación del modelo de rendimiento

**Figura 25**

*Conversión de las probabilidades predichas*

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Fuente: Elaboración Propia

Se carga el modelo previamente entrenado (modelo vgg19) y se aplica sobre un conjunto de datos de prueba llamado “test\_ds” el cuál contiene imágenes organizadas para las categorías “Sin Cáncer” y “Con Cáncer”

**Figura 26**

*Decisión final de categoría*

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Fuente: Elaboración Propia

Luego de que el modelo realiza las predicciones sobre las imágenes del conjunto de prueba, obtenemos un conjunto de probabilidades para cada imagen, indicando la confianza del modelo en cada categoría posible. Sin embargo, necesitamos convertir estas probabilidades en decisiones finales sobre la categoría de cada imagen. Para hacer esto, usamos np.argmax, que selecciona la categoría con la mayor probabilidad para cada imagen. Esto nos da las etiquetas de clase predichas, que podemos comparar con las etiquetas reales para evaluar el rendimiento del modelo.

**Figura 27**

*Etiquetas por lote*

Patrón de fondo

Descripción generada automáticamente

Fuente: Elaboración Propia

Aquí se itera todo el conjunto de datos de prueba “test\_ds”, extrayendo lotes de imágenes y sus correspondientes etiquetas reales. La instrucción print(y\_reales) dentro del bucle imprime las etiquetas reales de cada lote de imágenes.

Esto significa que cada vez que el bucle itera, y\_reales contiene las etiquetas reales de un lote de imágenes del conjunto de datos de prueba. Este tipo de iteración es útil para verificar qué etiquetas se están utilizando como datos de prueba y para asegurarse de que las etiquetas reales se correspondan correctamente con las imágenes que el modelo está evaluando.

**Figura 28**

*Almacenamiento de etiquetas reales y predictivas*

Texto

Descripción generada automáticamente

Fuente: Elaboración Propia

En este segmento de código, se inicializan dos listas vacías, y\_pred y y\_real, donde la primera almacenará las predicciones del modelo y la segunda almacenará las etiquetas reales de los datos de prueba. Luego, se inicia un bucle que itera sobre el conjunto de datos de prueba test\_ds, obteniendo en cada iteración un lote de imágenes (img\_batch) y sus correspondientes etiquetas reales (y\_batch). Las etiquetas reales de cada lote se añaden a la lista y\_real, y el modelo modelo\_vgg se utiliza para predecir las clases de las imágenes en el lote, almacenando las predicciones resultantes en la variable predicciones. Para cada lote de predicciones, se determina la clase con la mayor probabilidad usando np.argmax(predicciones, axis=1), lo que convierte las probabilidades de salida en etiquetas de clase que se añaden a la lista y\_pred. Después de completar el bucle y haber procesado todos los lotes de datos, se concatenan las listas y\_real y y\_pred en tensores únicos utilizando tf.concat, donde y\_reales contiene todas las etiquetas reales concatenadas y y\_predicciones contiene todas las etiquetas predichas concatenadas.

**Figura 29**

*Evaluación del desempeño del modelo*

Captura de pantalla de computadora

Descripción generada automáticamente

Fuente: Elaboración Propia

Se definen los datos de etiquetas reales, predicciones y nombres de categorías. Genera una matriz de confusión que compara las etiquetas reales con las predichas, y finalmente, dibuja un gráfico de la matriz de confusión mostrando las proporciones de clasificaciones correctas e incorrectas, permitiéndote evaluar el desempeño de tu modelo de clasificación.

**Figura 30**

*Matriz de confusión*

Gráfico

Descripción generada automáticamente

Fuente: Elaboración Propia

La matriz de confusión de las dos clases: 'ConCancer' y 'SinCancer'. La interpretación de los valores en la matriz es:

**Clase 'ConCancer' (etiqueta real)**:

Verdaderos Positivos (TP): 1433 (0.96)

Falsos Negativos (FN): 72 (0.10)

**Clase 'SinCancer' (etiqueta real)**:

Falsos Positivos (FP): 57 (0.04)

Verdaderos Negativos (TN): 630 (0.90)

Los valores entre paréntesis representan la proporción de las predicciones respecto al total de predicciones para cada clase. Por ejemplo, 0.96 indica que el 96% de las predicciones para 'ConCancer' son correctas.

# DISCUSIÓN DE RESULTADOS

Nuestro modelo implementado en esta investigación VGG-19 superó a los otros dos modelos evaluados, con una precisión del 94.11% y una pérdida mínima, dentro del rango de aceptación para esta investigación. Esta precisión es comparable o incluso superior a la obtenida en otros estudios similares, lo que sugiere la efectividad de la arquitectura VGG-19 en este contexto.

En cuanto a la optimización del tiempo de diagnóstico, si bien no fue el enfoque principal de este estudio, la rapidez y eficiencia computacional de los modelos de Deep Learning son aspectos que deben considerarse en futuras investigaciones para garantizar su viabilidad clínica.

En relación con la literatura previa, nuestros resultados son consistentes con estudios que demuestran la eficacia de las redes neuronales convolucionales en el diagnóstico de enfermedades a partir de imágenes médicas. Además, la alta precisión alcanzada respalda la hipótesis inicial de que la implementación de un sistema inteligente basado en Deep Learning puede mejorar significativamente el diagnóstico del cáncer de próstata.

Es importante señalar que, si bien nuestros resultados son prometedores, aún existen desafíos a considerar, como la generalización del modelo a diferentes conjuntos de datos y la interpretación de las decisiones del modelo por parte de los profesionales de la salud. Estos aspectos deben abordarse en futuras investigaciones para garantizar la aplicabilidad clínica y el impacto real en la práctica médica.

En resumen, este estudio demuestra que el uso de modelos de Deep Learning, especialmente la arquitectura VGG19, puede ser una herramienta efectiva para el diagnóstico de cáncer de próstata a partir de imágenes médicas, contribuyendo así a mejorar la precisión y eficiencia en el proceso de detección de esta enfermedad.

# CONCLUSIONES

1. Este estudio demuestra la viabilidad y efectividad del uso de modelos de Deep Learning, especialmente la arquitectura VGG16, VGG19 y ResNet50, en el diagnóstico de cáncer de próstata a partir de imágenes médicas. La precisión alcanzada por el modelo VGG19, del 94,11%, es significativa y muestra un rendimiento prometedor en la detección de esta enfermedad.
2. La utilización del dataset SICAPv2, que contiene 13,827 imágenes segmentadas por la clasificación de Gleason de las cuáles terminaron siendo divididas entre 2 categorías finales: Sin Cáncer y Con Cáncer, que permitió entrenar y evaluar los modelos de manera exhaustiva, lo que garantiza la representatividad y la calidad de los resultados obtenidos.
3. La aplicación de los algoritmos VGG16, VGG19 y ResNet50 en este contexto confirma la importancia de seleccionar cuidadosamente la arquitectura del modelo para obtener resultados óptimos. Aunque todos los modelos mostraron buen desempeño, el VGG19 destacó como el más preciso y eficiente en esta tarea específica.
4. La capacidad de identificar y clasificar el cáncer de próstata mediante este sistema inteligente brinda una herramienta valiosa para los profesionales de la salud, lo que puede mejorar la precisión y la eficiencia en el diagnóstico y tratamiento de esta enfermedad.
5. La aplicación de Deep Learning en el campo del diagnóstico médico tiene el potencial de revolucionar la atención médica al proporcionar herramientas más precisas y eficientes para la detección temprana de enfermedades como el cáncer de próstata. Sin embargo, se requiere una validación adicional en entornos clínicos para su implementación exitosa.

En resumen, este trabajo destaca el potencial del Deep Learning para el diagnóstico de cáncer de próstata a partir de imágenes médicas, ofreciendo una herramienta precisa y eficaz que puede mejorar significativamente la detección y tratamiento de esta enfermedad, contribuyendo así a la mejora de la atención médica y la calidad de vida de los pacientes.

# RECOMENDACIONES

Se recomienda utilizar imágenes de alta calidad y resolución HD para garantizar una precisión óptima en el diagnóstico. Las imágenes deben ser tomadas en un entorno con buena iluminación para obtener resultados más confiables.

Para obtener una predicción precisa del cáncer de próstata, es importante que las imágenes médicas sean capturadas con corrección y que muestren claramente la región de interés. Se sugiere que los pacientes se sometan a un protocolo de captura de imágenes estandarizado para garantizar la consistencia y calidad de los datos.

Además de las imágenes médicas, sería beneficioso considerar variables adicionales como la edad del paciente, los antecedentes médicos familiares, los niveles de antígeno prostático específico (PSA) y otros marcadores relevantes para mejorar la precisión del diagnóstico.

Como trabajo futuro, se recomienda explorar la posibilidad de ampliar el alcance del sistema inteligente para diagnosticar otras enfermedades relacionadas con la próstata, como las infecciones prostáticas, incorporando más datos y características específicas de cada enfermedad.

Además, sería importante considerar la integración de factores ambientales y de estilo de vida, como la dieta, el ejercicio y los hábitos de salud, para una evaluación más completa del riesgo y la progresión del cáncer de próstata.

# BIBLIOGRAFÍA

Forero Cuellar, O. (2019). Clasificadores supervisados del cáncer de próstata a partir de imágenes de resonancia magnética en secuencias T2.

Alvaro Daniel Quispe Carbonel. (2020). *Predicción de edad ósea con red basada en VGG-16 y Transfer Learning.* Lima.

Carneiro Neto, P. (05 de 06 de 2020). Análisis basado en aprendizaje profundo del cáncer de próstata de MP-MRI.

Faez, K. (2018). Mejorar el diagnóstico temprano del cáncer de próstata mediante el uso de Redes Neuronales Artificiales y Aprendizaje Profundo. *IEEE Xplore*, 4.

Gamboa, H. A. (2014). *Inteligencia Artificial - Principios y Aplicaciones.*

Guajin, W., Jeremy Yuen-Chun, T., & Kup-Sze, C. (2018). Diagnóstico de cáncer de próstata en una población china mediante el uso de métodos de aprendizaje automático. *IEEE Xplore*, 4.

*IBM*. (2020). Obtenido de https://www.ibm.com/co-es/cloud/deep-learning

*iso25000*. (3 de 10 de 2022). Obtenido de https://iso25000.com/index.php/normas-iso-25000/iso-25010/23-usabilidad

Jian, X., Jia, W., Shaozhong, C., & Bilong, L. (2020). Application of a Novel and Improved VGG-19. *Conference Series*, 7.

Lal, H., Amjad, A., Saima, R., Sharjl, S., Adnan, I., Muhammad Usama, U., . . . Doug Young, S. (2019). Aplicación del enfoque de red bayesiana a determinar la asociación entre características morfológicas extraídas de imágenes de cáncer de próstata. 16.

Minsa. (2021). *El plan nacional de cuidados integrales de Cáncer.* Lima: Ministerio de Salud.

Mohamed , I., Mansoor , A., Mansour , T., Hari Kiran, V., & P., R. (2020). Una técnica de clasificación de aprendizaje automático para predicción del cáncer de próstata.

Nishant, B., & Shrivastava, M. (2021). ResNet50-Based Effective Model for Breast Cancer Classification Using Histopathology Images. *Computer Modeling in*, 17.

Organización Mundial de la Salud. (01 de 02 de 2022). págs. https://www.who.int/es/news-room/fact-sheets/detail/cancer.

Qingge, J., He, W., Yankui, S., & Jie, H. (2019). Optimized Deep Convolutional Neural Networks for Identification of Macular Diseases from Optical Coherence Tomography Images. *MDPI*, 12.

Rodríguez, J. S. (10 de 2020). *Mendeley Data*. Obtenido de Prostate Whole Slide Images with Gleason Grades Annotations: https://data.mendeley.com/datasets/9xxm58dvs3/2

Silva, J., Colomer, A., Salesc, M., Molina, R., & Naranjo, V. (2020). Profundizando en la escala de puntuación de Gleason: Un sistema automático de extremo a extremo para histología de próstata, clasificación y detección de patrones cribiformes.

Silva-Rodríguez, J. (20 de 07 de 2020). *SICAPv2 - Prostate Whole Slide Images with Gleason Grades Annotations*. Obtenido de https://data.mendeley.com/datasets/9xxm58dvs3/1

Vázquez, J. R. (2020). Epidemiología del cáncer de próstata. *OF NEGATIVE & NO POSITIVE RESULTS*, 12.

Vizcaíno-Salazar, G. J. (2017). Importancia del cálculo de la sensibilidad, la especificidad y otros parámetros estadísticos en el uso de las pruebas del diagnóstico clínico y de laboratorio. En G. J. Vizcaíno-Salazar. Obtenido de https://docs.bvsalud.org/biblioref/2018/05/883697/importancia-calculo-sensibilidad-y-especifidad.pdf

Yasser Mk, O., Fahima A., M., & Abdelrahman I., S. (09 de 10 de 2019). Predicción de la interacción farmacológica con adenosina Receptores que utilizan técnicas de aprendizaje automático y SMOTE. *IEEAccess*.