

Tecnicatura Superior en Ciencia de Datos e Inteligencia Artificial



CLASIFICACIÓN DE MELANOMAS – CARCINOMAS MEDIANTE IMÁGENES

Integrantes:

Gabriela Cáceres – gagabrielacaceres@gmail.com

Lilen Guzmán – lilenguzman2015@gmail.com

Silvia Carina Monzón - silviam261072@gmail.com

Nicolás González Da Silva – nmqds8@gmail.com

EVIDENCIA N°3 - COHORTE 2022

Año: 2024

Marco conceptual y oportunidad

Uno de los campos de mayor expectativa para la aplicación del Procesamiento de Imágenes se encuentra en el ámbito médico, específicamente en la detección temprana de los primeros síntomas o señales de diversas enfermedades. La vasta literatura y casuística médica, ya dispone de muchos de estos datos, pero el potencial de los mismos se ve mermado por las limitaciones naturales del entendimiento humano, como pueden ser la limitada abstracción, generalización o la dificultad de detectar patrones profundos. El Procesamiento de Imágenes mediante métodos de Machine Learning abre nuevas posibilidades ofreciendo su asistencia libre de aquellas limitaciones intrínsecas al ser humano.

Puntualmente, el procedimiento habitual para diagnosticar el cáncer de piel y otras patologías relacionadas se basa en la inspección visual de las lesiones, evaluando parámetros como el tamaño, color y forma. El cáncer de piel es el más común de todos los tipos de cáncer. Aunque el melanoma representa apenas el 1% de los casos, es uno de los más mortales, y su evolución depende en gran medida de un diagnóstico temprano. Un sistema de diagnóstico automatizado que pueda detectar el cáncer en sus primeras etapas, cuando es difícil de distinguir de otras lesiones cutáneas benignas, sería crucial para salvar miles de vidas. La supervivencia a cinco años de un paciente con melanoma alcanza el 93% si se detecta en sus primeras fases.

Dada la importancia de una detección temprana, es necesario diseñar e implementar sistemas de ayuda al diagnóstico de melanomas para reducir costos y disminuir la carga de trabajo de los profesionales de la dermatología. Por lo tanto, el desafío abordado consiste en agilizar los procesos de diagnóstico temprano y asistir en la gestión de la atención prioritaria, a través del diseño y la implementación de modelos predictivos basados en el análisis por imágenes. Esto constituye un desafío debido al bajo contraste entre el lunar y la piel que lo rodea, las similitudes entre un melanoma y un no-melanoma, la presencia de artefactos en la imagen (vello corporal), la forma irregular y poco definida de los bordes.

Objetivo General

Asistir a los profesionales de la salud en el diagnóstico temprano de melanomas o carcinomas, a través del diseño y la implementación de modelos predictivos basados en el análisis por imágenes

Objetivo Específico

Desarrollar un sistema automatizado de clasificación binaria de lesiones cutáneas, a través del análisis de imágenes, logrando determinar si estas son lesiones "benignas" o

“malignas” mediante la utilización de técnicas y algoritmos de Aprendizaje Automático.

Objetivos Técnicos

1. Recopilación, entendimiento y limpieza de datos.
2. Transformación y adaptación de los mismos.
3. Comparación, selección y entrenamiento del Modelo.
4. Generación de informes y visualización de datos para comunicar resultados de manera efectiva, aspecto a tener en cuenta

Normativa

Las normativas sobre el uso de imágenes de lunares en personas para detectar cáncer de piel varían según el país, aspectos a tener en cuenta:

- Consentimiento Informado
- Protección de Datos Personales
- Uso Ético

Obtención y Descripción de Datos

Los datos utilizados en el proyecto fueron adquiridos de la web de Kaggle, mientras que las imágenes específicas fueron recopiladas por el autor en el banco de imágenes del International Skin Imaging Collaboration (ISIC). El Dataset consiste en 1800 imágenes de lesiones cutáneas benignas y 1497 imágenes de lesiones malignas, separadas en carpetas según su clasificación, realizada esta por personal capacitado.

Todas las imágenes son de tipo JPEG (.jpg) de 224 x 224 píxeles, con 3 canales (BGR) y una profundidad de bits de 24. La calidad de las mismas es alta, capturadas mediante fotografía dermatoscópica. También es alta la calidad de datos en el dataset. Es decir, no se encuentran imágenes corruptas o que presenten anomalías de características.

<https://www.kaggle.com/datasets/fanconic/skin-cancer-malignant-vs-benign>

<https://www.isic-archive.com/>

Exploración y Entendimiento de los Datos

Se realizó una visualización superficial de los datos para comprender las características distintivas de cada clase, Luego, una serie de análisis más profundos:

- **Histograma de intensidad de píxeles:** La distribución de los píxeles en el histograma perteneciente a cada categoría nos ayuda a generalizar las diferencias en rangos de intensidad y frecuencias. Comparando las *figuras 1 y 2* se observa que las imágenes malignas poseen picos de menor intensidad y mucha mayor frecuencia. Esto indica lunares más oscuros y de mayor superficie que los benignos.

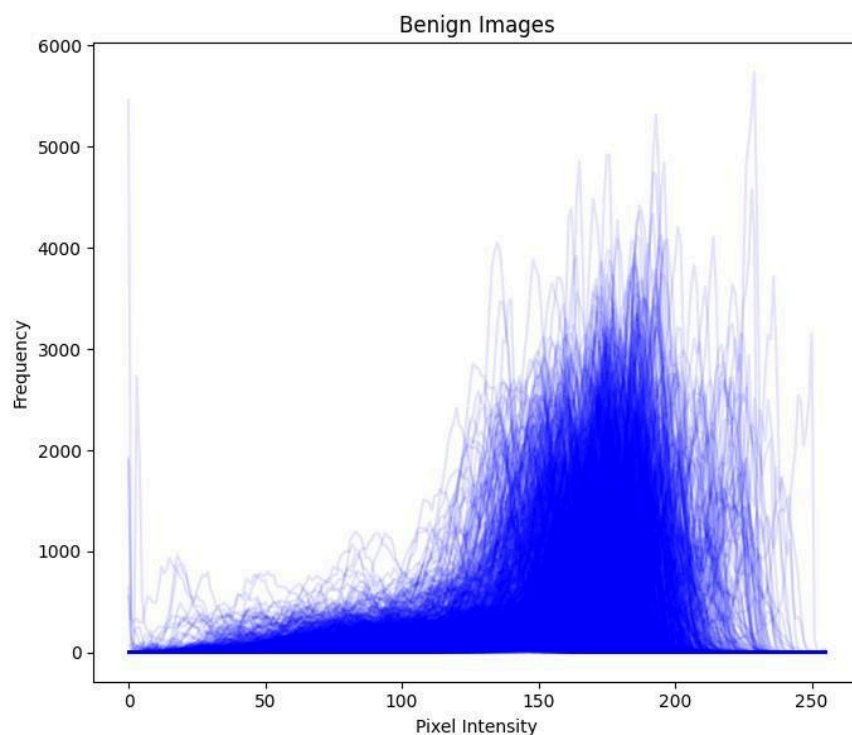


Figura 1: Histograma de Intensidad de píxeles para imágenes benignas

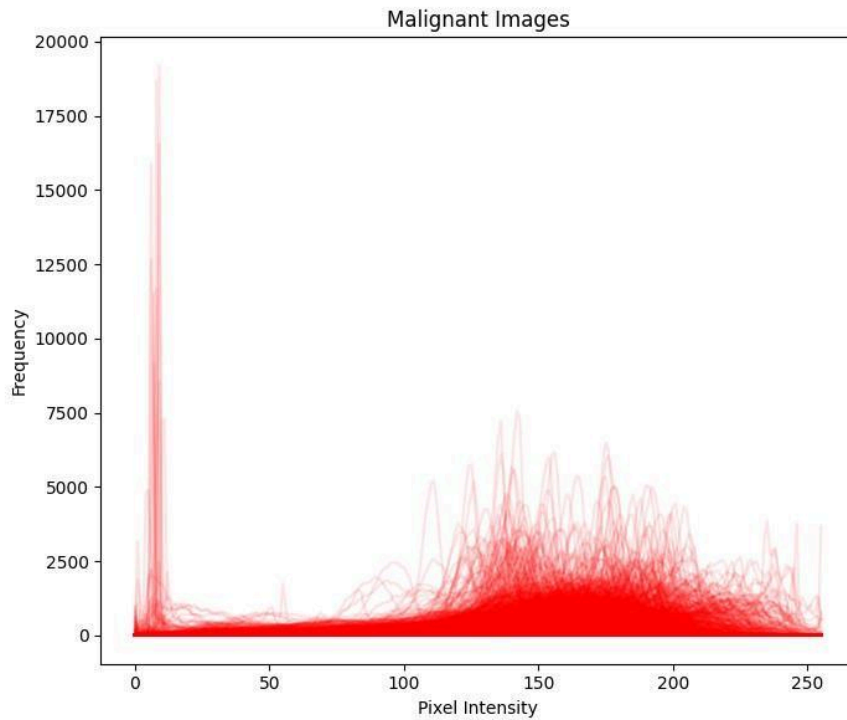


Figura 2: Histograma de Intensidad de píxeles para imágenes malignas

- Media y Varianza de Intensidad:** Complementando el histograma anterior, esta información nos permite comprender la variabilidad mayor o menor de intensidad que puede presentar una clase sobre otra. Se halló que las imágenes malignas presentan una media de intensidad de píxeles menor, con una varianza mayor. Es decir, que en promedio, son imágenes más oscuras y con menor homogeneidad de color.
- Análisis de Textura:** Mediante la matriz de co-ocurrencia de píxeles, se logró comparar las características de textura entre ambas clases de imágenes. Se analizó la media sobre el contraste, la disimilitud y la homogeneidad para cada clase, entre otras características. En la *Figura 3* se puede apreciar la gran diferencia de contraste entre ambas clases. Esto indica que en las imágenes malignas presentan coloraciones más extremas.

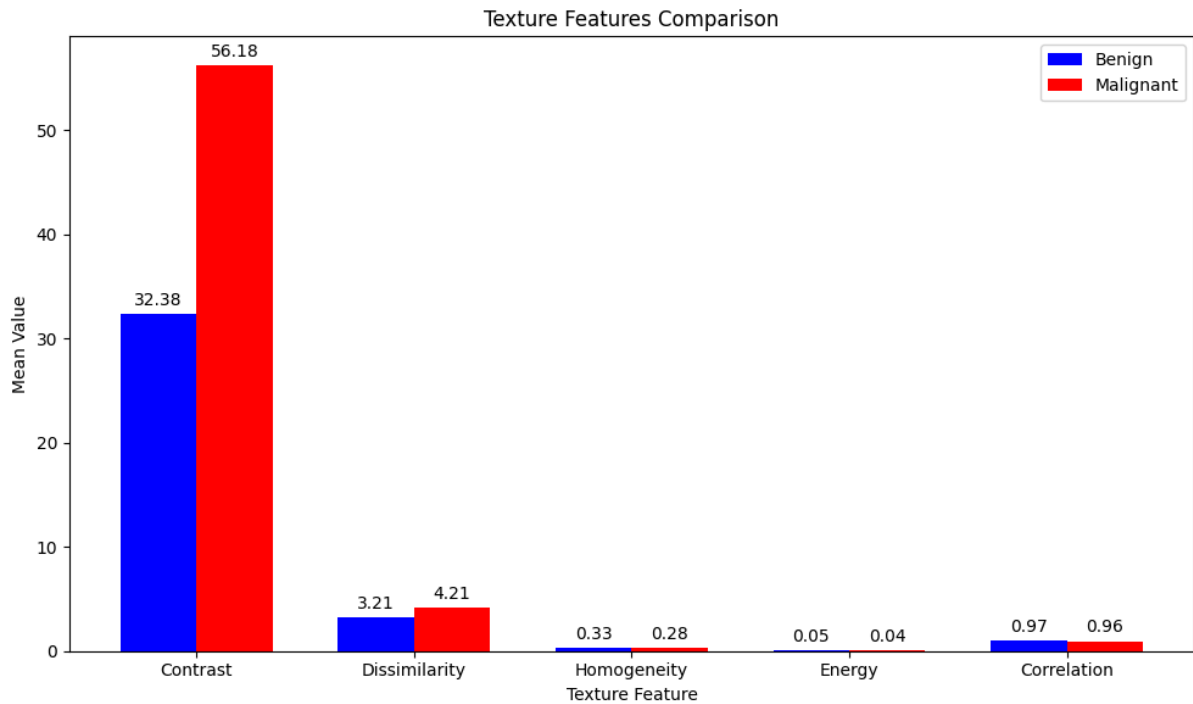


Figura 3: Gráfico de comparativa de características de textura

- Análisis Morfológico:** Mediante el análisis de la media de características morfológicas de interés como el área, perímetro, solidez y excentricidad, se realizó una comparativa entre estas para cada clase de imágenes. En la *figura 4* se observa la enorme diferencia hallada en el área para cada tipo de lesión.

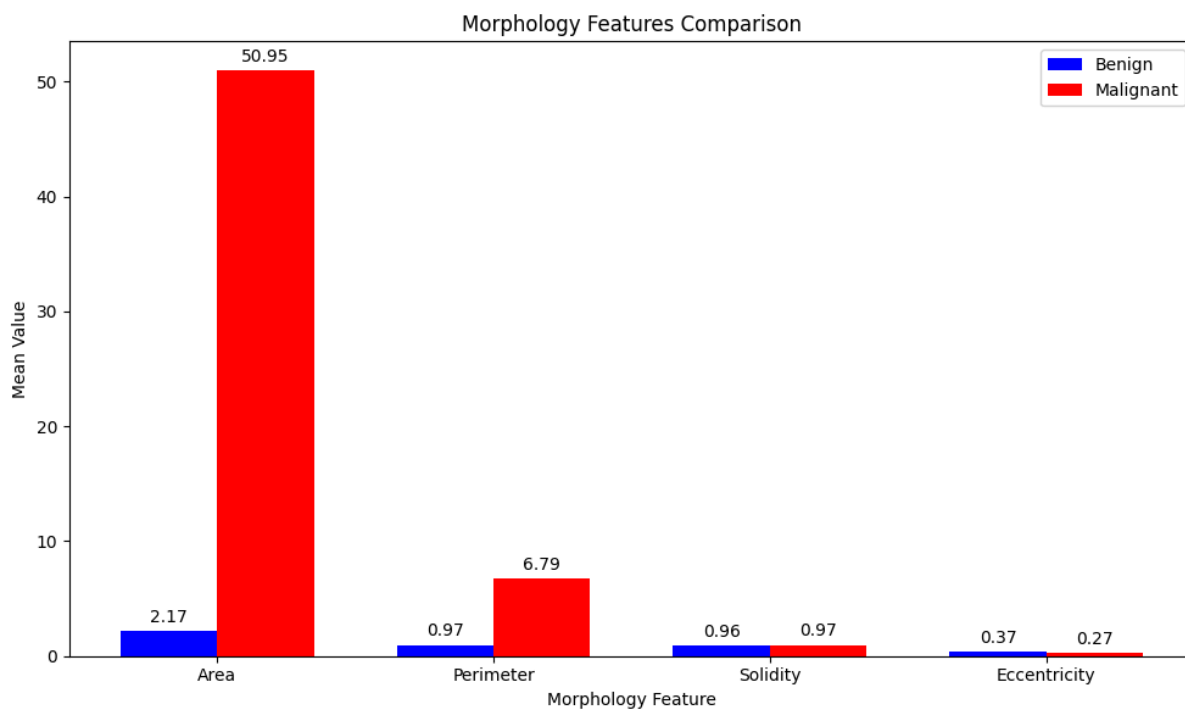


Figura 4: Gráfico de comparativa de características morfológicas.

- Histogramas de color:** Se realizó un nuevo análisis de intensidad de píxeles pero esta vez separando los datos por canal, es decir, por color. Esto puede ayudar a detectar nuevas características de interés. En la *figuras 5* se observa que las imágenes benignas poseen una alta frecuencia de píxeles en intensidades altas correspondientes al canal rojo.

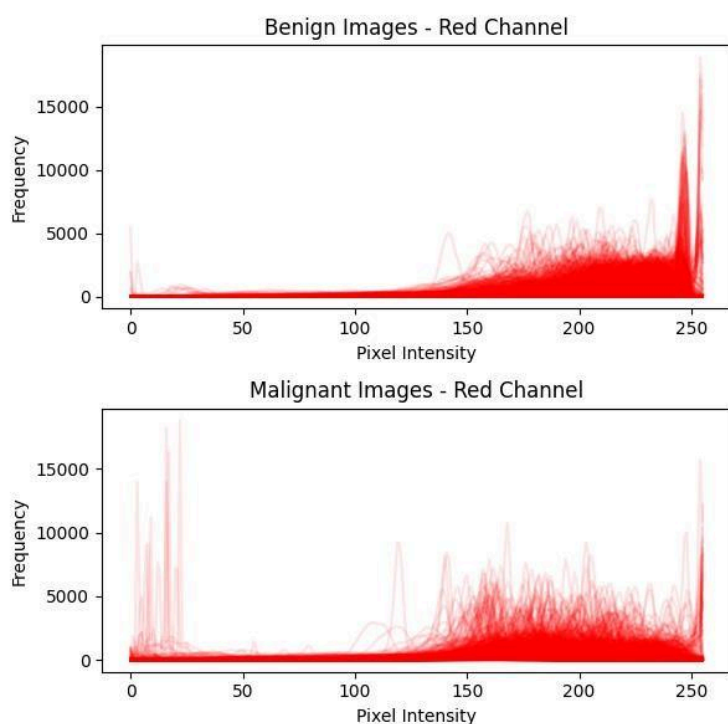


Figura 5: Histograma de intensidad de píxeles para el canal rojo.

- **Distribución y balance de clases:** Por último, se analizó la distribución de imágenes en cada clase, para corroborar que ambas estén balanceadas y evitar el riesgo de sesgo a la hora de entrenar el modelo.

Preparación de los Datos

Una vez entendidos los datos, para asegurar su calidad e idoneidad en el desarrollo del modelo de clasificación, se llevaron a cabo una serie de técnicas previo a la construcción y alimentación del modelo.

- **Segmentación de Región de Interés (ROI):** Esta técnica extrae la región con el contorno más grande presente en cada imagen y elimina el fondo no relevante. Para ello aplica una umbralización y búsqueda de contornos. Esto mejora la precisión del modelo, al centrarse en características relevantes.
- **Aumentación de los datos:** Se utilizaron técnicas de aumentación, como rotación, cambio de escala y recorte, para incrementar la variabilidad del conjunto de datos y mejorar así, la capacidad del modelo para generalizar.
- **Preprocesamiento:** Por último, se aplica un preprocesamiento a cada imagen que se encarga de implementar la segmentación ROI, de redimensionar la imagen a la medida deseada, realizar una transformación de color a RGB (necesaria para algunos modelos a entrenar) y normalizar la misma.

Modelado

Luego de la etapa Preparación de los Datos, y siguiendo la metodología CRISP-DM de minería de datos se procedió a desarrollar *modelos de redes neuronales convolucionales (CNN)*, para la clasificación de imágenes de lunares benignos y malignos.

Las redes neuronales artificiales convolucionales son una de las categorías de redes neuronales más conocidas y usadas, especialmente para datos con alta dimensionalidad, como es el caso de las imágenes y los videos. Ellas operan de manera similar a las redes neuronales estándar. Una de las diferencias claves, es que cada unidad en una capa es un filtro de 2 dimensiones (o superior) que es convolucionado con la entrada de dicha capa. La que es esencial para casos donde se quieren aprender patrones de entradas de altas dimensiones, por ejemplo, las imágenes o videos. Las CNN son un tipo útil de modelo para paradigmas de aprendizajes supervisados y no supervisados.

Un mecanismo de aprendizaje supervisado es aquel en el que se conocen la entrada de la red y las salidas requeridas (etiquetas verdaderas), aprendiendo de esta manera el "mapeo" entre los dos.

Para el desarrollo del proyecto, utilizamos el algoritmo de red neuronal convolucional (CNN). Creamos tres modelos con diferentes niveles de complejidad para la clasificación de imágenes de lunares benignos y malignos, de la siguiente manera:

→ **Modelo 1: Modelo Simple Secuencial**

Es un modelo secuencial simple, que apila las capas una sobre otra en orden secuencial con una arquitectura de Red Neuronal Convolucional (CNN), diseñada para clasificar imágenes de lunares como benignos o malignos.

Es el modelo más básico con capas convolucionales, de pooling, de aplanamiento y densas.

1. Descripción del Modelo Secuencial

Se utiliza la clase Sequential de Keras para crear un modelo secuencial. Esto significa que las capas del modelo se añaden una tras otra.

2. Capas Convolucionales y de Pooling

- Primera Capa Convolucional
- Segunda Capa Convolucional
- Tercera Capa Convolucional

3. Capa de Aplanamiento y Densas

- Capa de Aplanamiento
- Capa Densa

- Capa de Salida

En resumen, éste modelo simple secuencial es una CNN básica que consta de tres bloques de capas convolucionales seguidos de capas de pooling, una capa de aplanamiento, y dos capas densas. Es adecuado para tareas de clasificación binaria y proporciona una buena base para comparar el rendimiento de modelos más complejos como VGG16 o ResNet-50.

→ **Modelo 2: Modelo con VGG16**

Este modelo utiliza una red pre entrenada de redes neuronales convolucionales (VGG16) como base y añade capas densas personalizadas para la clasificación de imágenes de lunares benignos y malignos.

1. Descripción del Modelo Basado en VGG16:

- Base Pre Entrenada del Modelo VGG16, que utiliza arquitectura VGG16 como base, cargando los pesos pre entrenados de ImageNet y se define la forma de la entrada de las imágenes 224 x 224 y 3 canales (RGB).
- Congelación de capas: Congela todas las capas de la base VGG16 para que no se actualicen durante el entrenamiento. Esto es útil para transferir las características aprendidas en ImageNet a nuestro problema sin modificar estas características.

2. Capas del Modelo:

1. Base del Modelo:

La arquitectura VGG16 preentrenada sin las capas superiores.

2. Global Average Pooling

GlobalAveragePooling2D(): Reduce cada mapa de características 2D a un solo valor promedio. Esta capa convierte las salidas de la ResNet-50 a un vector unidimensional.

3. Capa Densa

Dense(512, activation='relu'): Añade una capa densamente conectada con 512 unidades o neuronas y función de activación ReLU. Esta capa introduce no linealidad y permite aprender combinaciones complejas de las características extraídas.

4. Dropout

Dropout(0.5): Apaga aleatoriamente el 50% de las unidades durante el entrenamiento, ayudando a prevenir el sobreajuste (overfitting).

5. Capa de Salida

Dense(1, activation='sigmoid'): Añade una capa de salida con una sola unidad y función de activación sigmoide, es adecuada para la clasificación binaria.

En resumen, éste modelo combina la capacidad de extracción de características de VGG16 pre entrenada con una red neuronal completamente conectada personalizada para la tarea de clasificación binaria (lunares benignos vs. malignos). La arquitectura global es adecuada para aprovechar el aprendizaje transferido y ajustar el modelo a los datos específicos del problema con capas adicionales.

→ Modelo 3: Modelo con ResNet-50

Este modelo utiliza la arquitectura ResNet-50 pre entrenada de redes neuronales convolucionales con los pesos del conjunto de datos ImageNet.

1. Descripción del Modelo Basado en ResNet-50:

- Base Preentrenada del Modelo (ResNet-50), que utiliza arquitectura ResNet50 como base, cargando los pesos preentrenados de ImageNet y se define la forma de la entrada de las imágenes 224 x 224 y 3 canales (RGB).
- Congelación de capas: Congela todas las capas de la base ResNet50 para que no se actualicen durante el entrenamiento. Esto es útil para transferir las características aprendidas en ImageNet a nuestro problema sin modificar estas características.

2. Capas del Modelo:

1. Base del Modelo:

base_model: La arquitectura ResNet-50 preentrenada sin las capas superiores.

2. Global Average Pooling

`GlobalAveragePooling2D()`: Reduce cada mapa de características 2D a un solo valor promedio. Esta capa convierte las salidas de la ResNet-50 a un vector unidimensional.

3. Capa Densa

`Dense(512, activation='relu')`: Añade una capa densamente conectada con 512 unidades o neuronas y función de activación ReLU. Esta capa introduce no linealidad y permite aprender combinaciones complejas de las características extraídas.

4. Dropout

`Dropout(0.5)`: Apaga aleatoriamente el 50% de las unidades durante el entrenamiento, ayudando a prevenir el sobreajuste (overfitting).

5. Capa de Salida

`Dense(1, activation='sigmoid')`: Añade una capa de salida con una sola unidad y función de activación sigmoide, es adecuada para la clasificación binaria.

En resumen, éste modelo utiliza la arquitectura ResNet 50 pre entrenada para extraer características de las imágenes. Las capas adicionales agregadas permiten adaptar estas características a la tarea de clasificación binaria. El modelo está configurado para evitar el sobreajuste mediante la congelación de las capas de la base ResNet 50 y la aplicación de dropout. Además, se utiliza un optimizador Adam con una tasa de aprendizaje baja para ajustar los pesos del modelo durante el entrenamiento.

Este enfoque combina la capacidad de ResNet50 para aprender representaciones profundas de imágenes con capas adicionales para una clasificación binaria efectiva, asegurando que el modelo sea capaz de generalizar bien a datos nuevos.

Entrenamiento del modelo:

- Se compila el modelo con el optimizador Adam, una tasa de aprendizaje de 0.0001, que es popular por su eficiencia y capacidad de adaptarse a diferentes problemas. Con función de pérdida de entropía cruzada binaria y métrica accuracy, para evaluar el rendimiento del modelo durante el entrenamiento y la validación.
- Callbacks:
 - `ModelCheckpoint`: Guarda el modelo con los mejores resultados en la pérdida de validación.

- EarlyStopping: Detiene el entrenamiento si no hay mejora en la pérdida de validación después de 10 épocas y restaura los pesos del mejor modelo.

Los tres modelos compilan con los parámetros indicados en la compilación.

Pero el Modelo 2 y 3 incluye callbacks adicionales (Model Checkpoint y Early Stopping) para mejorar la gestión del entrenamiento.

Modelo 2 y 3 son enfoques que podría ser más efectivo porque usa:

Transfer Learning: Usa características pre entrenadas que ya captaron patrones visuales útiles.

Capas Personalizadas: Añadimos capas específicas para ajustar las características pre entrenadas al problema particular.

Regularización: Usa Dropout para reducir el riesgo de sobreajuste.

Análisis individual de cada modelo

Interpretar los resultados del entrenamiento de los modelos nos permite entender su funcionamiento y determinar qué ajustes podrían ser necesarios para mejorar su rendimiento. A continuación, se detallan los resultados de cada modelo.

Modelo 1: Modelo Simple Secuencial

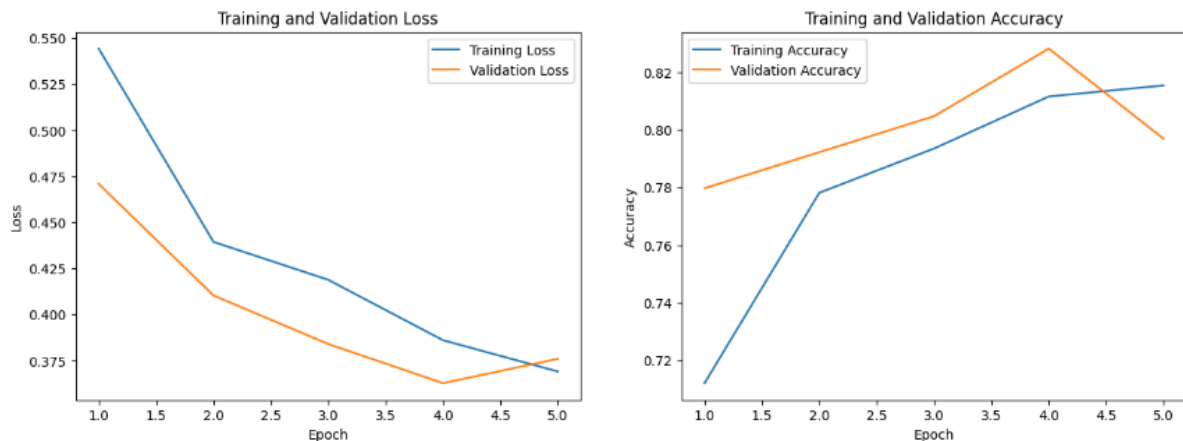
Métricas para evaluar el modelo:

- ❖ Loss (Pérdida): Indica qué tan mal está funcionando el modelo. Una pérdida baja indica que el modelo está haciendo buenas predicciones.
- ❖ Accuracy (Precisión): Proporción de predicciones correctas sobre el total de predicciones realizadas.
- ❖ Precision, Recall, y F1-Score: Métricas adicionales que son útiles especialmente en conjuntos de datos desbalanceados.

Métricas de Entrenamiento y Validación por Epoch:

Modelo Simple Secuencial con Capas Convolucionales y Densas				
Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
<u>1</u>	0.5442	0.7121	0.4708	0.7797
<u>2</u>	0.4393	0.7781	0.4103	0.7922
<u>3</u>	0.4188	0.7935	0.3839	0.8047
<u>4</u>	0.3860	0.8115	0.3627	0.8281
<u>5</u>	0.3690	0.8154	0.3759	0.7969

Gráficas de Pérdida y Precisión durante el Entrenamiento y la Validación:



Análisis

- ❖ **Precisión del Modelo:** La precisión del modelo mejora con cada epoch, alcanzando un 81.54% en el conjunto de entrenamiento y un 79.69% en el conjunto de validación al final de la quinta época.
- ❖ **Pérdida del Modelo:** La pérdida disminuye continuamente tanto para el conjunto de entrenamiento como para el de validación, indicando que el modelo está aprendiendo de manera efectiva.

Resultados

El modelo entrenado ha mostrado un rendimiento sólido, con una precisión de validación que supera el 79%. La disminución continua de la pérdida sugiere que el modelo está aprendiendo y generalizando bien a partir de los datos de

entrenamiento y validación. Sin embargo, se observa una ligera subida en la pérdida de validación en la última época, lo cual podría indicar un inicio de sobreajuste. Ajustar hiperparámetros o aplicar técnicas de regularización podría mejorar aún más el rendimiento del modelo.

Modelo 2: Modelo con VGG16

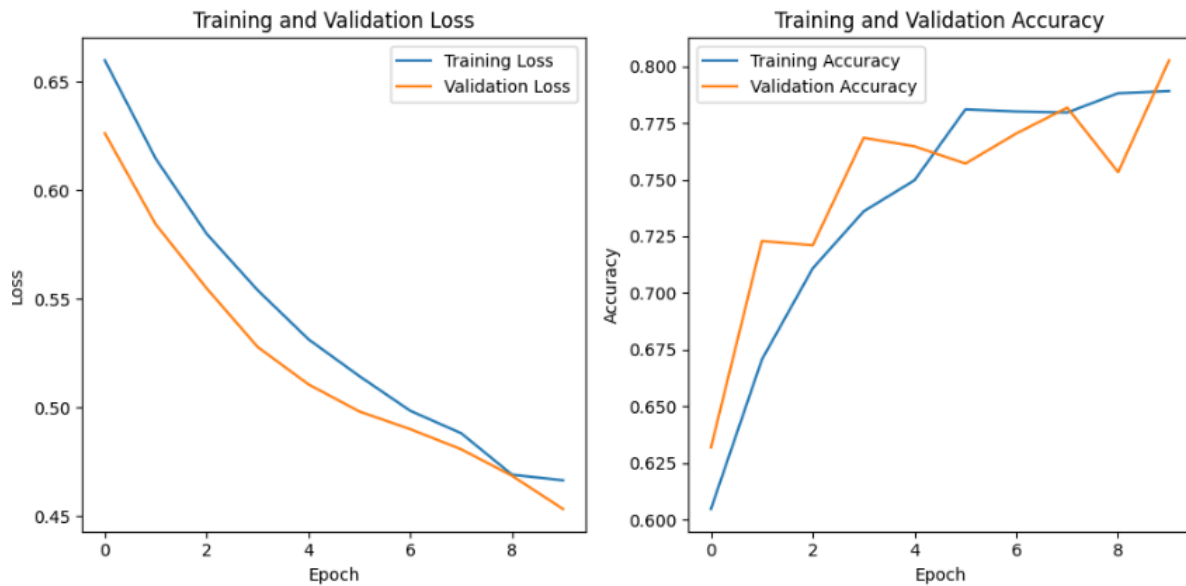
Métricas para evaluar el modelo:

- ❖ Precisión (Accuracy): Mide la proporción de verdaderos positivos y verdaderos negativos sobre el total de casos evaluados.
- ❖ Loss (Pérdida): Indica qué tan mal está funcionando el modelo. Una pérdida baja sugiere que el modelo está realizando predicciones más precisas.

Métricas de Entrenamiento y Validación por Epoch:

<i>Modelo VGG16</i>				
Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	0.6599	0.6047	0.6263	0.6319
2	0.6145	0.6706	0.5843	0.7230
3	0.5800	0.7109	0.5549	0.7211
4	0.5540	0.7360	0.5279	0.7685
5	0.5314	0.7498	0.5106	0.7647
6	0.5144	0.7810	0.4981	0.7571
7	0.4985	0.7801	0.4900	0.7704
8	0.4881	0.7796	0.4807	0.7818
9	0.4690	0.7882	0.4685	0.7533
10	0.4665	0.7891	0.4533	0.8027

Gráficas de Pérdida y Precisión durante el Entrenamiento y la Validación:



Métricas de Rendimiento:

En el conjunto de prueba, el modelo obtuvo:

- ❖ Test Loss (Pérdida en el conjunto de prueba): 0.4277
- ❖ Test Accuracy (Precisión en el conjunto de prueba): 0.8197

Análisis

- ❖ Precisión del Modelo: La precisión del modelo mejoró con cada época, alcanzando un 81.97% en el conjunto de prueba.
- ❖ Pérdida del Modelo: La pérdida disminuyó continuamente tanto en el conjunto de entrenamiento como en el de validación, indicando que el modelo está aprendiendo eficazmente y generalizando bien.

Resultados

El modelo VGG16 mostró una mejora continua en la precisión y una disminución constante en la pérdida durante el entrenamiento y la validación. Alcanzó una precisión del 81.97% en el conjunto de prueba, lo que indica que es efectivo para la clasificación de imágenes en este caso. Para mejorar aún más, se pueden considerar ajustes adicionales en los hiperparámetros y la aplicación de técnicas de regularización.

Modelo 3: Modelo con ResNet-50

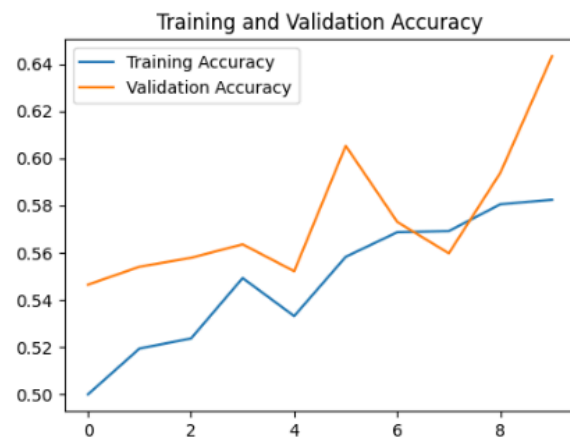
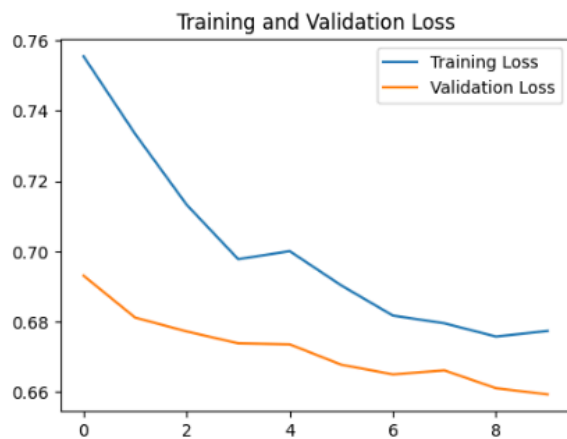
Métricas para evaluar el modelo:

- ❖ **Loss (Pérdida):** Mide la discrepancia entre las predicciones del modelo y las etiquetas reales. Una menor pérdida indica un mejor rendimiento del modelo.
- ❖ **Accuracy (Precisión):** Proporción de predicciones correctas sobre el total de predicciones realizadas. Una mayor precisión indica un modelo más efectivo.

Métricas de Entrenamiento y Validación por Epoch:

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	7.555	5.000	6.931	5.465
2	7.334	5.194	6.812	5.541
3	7.133	5.237	6.773	5.579
4	6.978	5.493	6.739	5.636
5	7.001	5.332	6.736	5.522
6	6.903	5.583	6.678	6.053
7	6.818	5.687	6.651	5.731
8	6.796	5.692	6.662	5.598
9	6.758	5.806	6.611	5.939
10	6.774	5.825	6.594	6.433

Gráficas de Pérdida y Precisión durante el Entrenamiento y la Validación:



Métricas de Rendimiento:

En el conjunto de prueba, el modelo obtuvo:

- ❖ Test Loss (Pérdida en el conjunto de prueba): 0.6605
- ❖ Test Accuracy (Precisión en el conjunto de prueba): 0.6530

Análisis

- ❖ Precisión del Modelo: La precisión del modelo muestra una mejora gradual, alcanzando un 65.30% en el conjunto de prueba.
- ❖ Pérdida del Modelo: La pérdida disminuye continuamente, tanto en el conjunto de entrenamiento como en el de validación, lo que sugiere que el modelo está aprendiendo y generalizando adecuadamente.

Resultados

En resumen, el modelo ResNet50 entrenado en este conjunto de datos ha demostrado una mejora en la precisión a lo largo de las epochs, alcanzando una precisión de prueba de 65.30%. La disminución de la pérdida a lo largo de las epochs indica que el modelo está aprendiendo y ajustándose adecuadamente a los datos.

Comparación del Rendimiento de los Modelos:

Comparación de Rendimiento						
Modelo	Precisión de Entrenamiento	Precisión de Validación	Precisión de Prueba	Pérdida de Entrenamiento	Pérdida de Validación	Pérdida de Prueba
Simple	81.54%	79.69%	-	0.3690	0.3759	-
VGG16	78.91%	80.27%	81.97%	0.4665	0.4533	0.4277
ResNet50	58.25%	64.33%	65.30%	0.6774	0.6594	0.6605

- ❖ VGG16 se destaca como el mejor modelo en términos de precisión y pérdida tanto en entrenamiento como en validación y prueba. Es especialmente adecuado para tareas donde la precisión es crucial.

- ❖ Modelo Simple Secuencial muestra un rendimiento razonable pero carece de datos de prueba para una comparación completa. Aun así, tiene una alta precisión de entrenamiento y validación.
- ❖ ResNet50 tiene un rendimiento menor en comparación con los otros modelos, pero aún puede ser útil en contextos donde se necesite un balance entre precisión y otros factores como la arquitectura más avanzada para aprender características más complejas.

Posibles Mejoras:

- **Más datos:** Recopilar más datos de entrenamiento puede ayudar a mejorar la generalización del modelo.
- **Ajuste de Hiper Parámetros:** Experimentar con la tasa de aprendizaje, el número de épocas, y el tamaño del lote puede mejorar el rendimiento.
- **Aumentación de Datos:** Usar técnicas de aumentación de datos más agresivas puede ayudar a prevenir el sobreajuste.
- **Capas y neuronas:** Ajustar el número de capas y neuronas en el modelo puede mejorar su capacidad de aprendizaje.

Mejoras o Escalabilidad a Futuro:

Infraestructura en la Nube: Utilizar servicios en la nube como AWS, Google Cloud, o Azure para desplegar el modelo.

Incorporar técnicas de procesamiento del habla: Estas incorporaciones pueden hacer al sistema más interactivo, accesible y útil para una variedad de usuarios y contextos médicos.

1. Asistente de Voz para la Interacción con el Usuario: Implementar un asistente de voz para guiar a los usuarios a través del proceso de análisis de imágenes.
2. Análisis de Opiniones de Pacientes: Utilizar técnicas de procesamiento del lenguaje natural para analizar los comentarios de los pacientes sobre el sistema.
3. Integración con Sistemas Médicos: Facilitar la integración del sistema de detección con otros sistemas médicos mediante la entrada y salida de voz.

Links:

- GitHub: https://github.com/gagabriela2787/Grupo16_ISPC.git
- Trello: <https://trello.com/b/bIQ6Q96K/ISPC>
- Drive: https://drive.google.com/drive/folders/1MGtcoDCaC7B-fdwKJ0ITMvAKlQq4PvH?usp=drive_link

Bibliografía y fuentes

- ❖ <https://www.kaggle.com/>
- ❖ The Process of Knowledge Discovery on Databases, Timarón-Pereira,
- ❖ Guía paso a paso de Minería de Datos, Pete Chapman (NCR),
- ❖ Introduction to Machine Learning with Python, Andreas C. Müller & Sarah Guido
- ❖ Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow - Aurelien Geron
- ❖ SCRUM Y XP DESDE LAS TRINCHERAS , Henrik Kniberg