



**Evaluación del impacto de la computación cuántica en la clasificación de
imágenes mediante modelos de CNN**

Johan Arbey Vargas Mejía

Monografía para optar al título de especialista en analítica y ciencia de datos

Asesor

Javier Fernando Botia Valderrama, PhD

Universidad de Antioquia

Departamento de Ingeniería de Sistemas

Especialización en Analítica y Ciencia de Datos

Medellín

2023

Cita	Vargas Mejía, 2023 [1]
Referencia	[1] Vargas Mejía J. A. “Evaluación del impacto de la computación cuántica en la clasificación de imágenes mediante modelos de CNN”, Trabajo de grado especialización, Universidad de Antioquia, Medellín, 2023.
Estilo IEEE (2020)	



Especialización en Analítica y Ciencia de Datos, Cohorte IV.



Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.

Decano/Director Julio Cesar Saldarriaga Molina.

Jefe departamento: Diego José Luis Botia Valderrama.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Dedicatoria

A la curiosidad, que nos impulsa a crecer.

Agradecimientos

Agradezco profundamente a mi familia por su amor incondicional, a mis profesores por su guía, y a mis compañeros por el viaje compartido en esta aventura intelectual.

Índice

RESUMEN	7
ABSTRACT	8
I. Descripción del problema	9
A. Problema de negocio.	9
B. Aproximación desde la analítica de datos	10
C. Origen de los datos	11
D. Métricas de desempeño	12
II. Objetivos.	15
A. Objetivo general	15
B. Objetivos específicos	15
III. Datos	16
A. Datos originales.	16
B. Datasets	17
C. Analítica descriptiva.	18
IV. Proceso de analítica	23
A. Pipeline principal	23
B. Preprocesamiento	25
C. Modelos	26
D. Métricas	31
V. Metodología	32
A. Baseline	32
B. Validación	32
VI. Resultados y discusión.	34
VII. CONCLUSIONES	40
VIII. RECOMENDACIONES	42
REFERENCIAS	43

LISTA DE TABLAS

Tabla I	Resultados de los experimentos con los modelos TensorFlow, PyTorch y Híbrido	41
---------	---	----

LISTA DE FIGURAS

Fig. 1	Distribución de categorías. Elaboración propia.	18
Fig. 2	Muestra de retratos. Elaboración propia.	19
Fig. 3	Muestra de otras categorías. Elaboración propia.	19
Fig. 4	Intensidad de brillo. Elaboración propia.	20
Fig. 5	Análisis de dos componentes PCA. Elaboración propia.	21
Fig. 6	Análisis de dos componentes t-SNE. Elaboración propia.	21
Fig. 7	Pipeline. Elaboración propia.	23
Fig. 8	Arquitectura CNN con Tensorflow. Elaboración propia.	26
Fig. 9	Arquitectura CNN con PyTorch. Elaboración propia.	28
Fig. 10	Arquitectura CNN híbrida. Elaboración propia.	29
Fig. 11	Evolución de la Precisión en CNN de TensorFlow.	34
Fig. 12	Análisis de la Pérdida de Entrenamiento con Regularización L2.	35
Fig. 13	Matriz de confusión del modelo de TensorFlow.	36
Fig. 14	Evolución de la Precisión en CNN de PyTorch.	37
Fig. 15	Matriz de confusión del modelo de PyTorch.	37
Fig. 16	Evolución de la Precisión en CNN híbrida.	38
Fig. 17	Matriz de confusión del modelo híbrido.	39

RESUMEN

Este estudio se centra en la evaluación del impacto que la computación cuántica tiene en la clasificación de imágenes, empleando modelos de Redes Neuronales Convolucionales (CNN). El objetivo principal es comparar el desempeño de modelos tradicionales de CNN desarrollados con TensorFlow y PyTorch contra modelos que integran un circuito cuántico como elemento distintivo de procesamiento. La metodología involucra la aplicación de un conjunto de datos estandarizado de imágenes de retratos, donde cada modelo CNN, uno basado en TensorFlow y otro en PyTorch, es entrenado y evaluado en su capacidad de clasificar las imágenes de forma binaria. Posteriormente, se introduce un circuito cuántico con el propósito de explorar las ventajas potenciales que la superposición y el entrelazado cuánticos pueden ofrecer en el proceso de clasificación.

Los resultados obtenidos sugieren que la integración de circuitos cuánticos en la arquitectura de las redes neuronales convolucionales no solo es viable, sino que también produce resultados competitivos en comparación con los enfoques clásicos establecidos. Este rendimiento es notable dado que la computación cuántica, aún en su naciente integración con técnicas de aprendizaje profundo, logra igualar casi a la perfección los resultados de su homólogo clásico. Esta paridad de rendimiento destaca el potencial significativo de los circuitos cuánticos en enriquecer y potenciar las capacidades de las CNN en tareas de clasificación de imágenes.

Palabras clave — Computación Cuántica, Clasificación Binaria de Imágenes, Redes Neuronales Convolucionales (CNN), TensorFlow (TF), PyTorch.

ABSTRACT

This study focuses on assessing the impact that quantum computing has on image classification, employing Convolutional Neural Network (CNN) models. The main goal is to compare the performance of traditional CNN models developed with TensorFlow and PyTorch against models that integrate a quantum circuit as a distinctive processing element. The methodology involves applying a standardized dataset of portrait images, where each CNN model, one based on TensorFlow and the other on PyTorch, is trained and evaluated on its ability to classify the images in a binary manner. Subsequently, a quantum circuit is introduced with the purpose of exploring the potential advantages that quantum superposition and entanglement can offer in the classification process.

The obtained results suggest that the integration of quantum circuits into the architecture of convolutional neural networks is not only viable but also yields competitive results compared to established classical approaches. This performance is remarkable given that quantum computing, even in its nascent integration with deep learning techniques, manages to match almost perfectly the results of its classical counterpart. This performance parity highlights the significant potential of quantum circuits to enrich and enhance the capabilities of CNNs in image classification tasks.

Keywords — Quantum Computing, Binary Image Classification, Convolutional Neural Networks (CNN), TensorFlow (TF), PyTorch

I. DESCRIPCIÓN DEL PROBLEMA

El interrogante central de esta investigación se concentra en determinar el grado de mejora que la computación cuántica podría aportar a los sistemas de clasificación binaria de imágenes, específicamente en la identificación de retratos de personas. En un mundo donde la cantidad de datos visuales crece exponencialmente, la necesidad de clasificar y procesar estas imágenes con alta precisión y eficiencia se ha vuelto imperativa. Este estudio pretende explorar cómo los avances en computación cuántica pueden ser aplicados para potenciar las Redes Neuronales Convolucionales (CNN), utilizando frameworks como TensorFlow y PyTorch. Se busca comprender si la integración de circuitos cuánticos en las CNN puede ofrecer un salto cualitativo en la precisión de la clasificación y una optimización en el uso de recursos computacionales.

A. Problema de negocio

El problema de negocio se centra en la capacidad de mejorar y optimizar los procesos mediante la clasificación de imágenes, una tarea esencial para la segmentación de clientes, la personalización de contenidos y la eficiencia en la publicidad dirigida. La utilización de la visión por computadora y el aprendizaje profundo enfrenta desafíos específicos al entrar en producción, tales como la adquisición de datos de alta calidad, la gestión de expectativas de rendimiento del modelo, la pragmática en la vanguardia de la tecnología de red, y la planificación de la ingesta de datos, almacenamiento, seguridad y la infraestructura en general [1].

El concepto emergente de las redes neuronales convolucionales cuánticas (QCNN) representa una innovadora intersección entre la inteligencia artificial y la computación cuántica. Los QCNN buscan superar las limitaciones de los modelos de aprendizaje profundo tradicionales mediante la incorporación de principios cuánticos, como la superposición y el entrelazamiento. Estos modelos prometen mejoras en la eficiencia y la capacidad de procesamiento, especialmente en la clasificación de imágenes, aprovechando el paralelismo inherente a los sistemas

cuánticos. Este enfoque tiene el potencial de transformar fundamentalmente la manera en que las máquinas aprenden y procesan información a gran escala [2].

B. Aproximación desde la analítica de datos

Los modelos predictivos clásicos en la analítica de datos, han sido el soporte principal de la toma de decisiones basada en datos en muchas industrias. Estos modelos de aprendizaje profundo, son cruciales para interpretar y analizar imágenes, lo que permite desde el reconocimiento facial en seguridad hasta la detección de anomalías. Aunque carecen de la velocidad y el potencial de procesamiento paralelo de los modelos cuánticos, los modelos clásicos están bien establecidos, comprendidos y son accesibles, lo que les permite ser desplegados con relativa facilidad en entornos de producción. Además, ofrecen la ventaja de ser probados y verdaderos en escenarios donde la interpretación precisa de datos es vital, como en el seguimiento de objetos, inspección de calidad en la manufactura y análisis de comportamiento del consumidor en el marketing.

En el ámbito de la salud, por ejemplo, los modelos **QCNN** podrían emplearse para el diagnóstico médico automatizado, analizando imágenes de resonancias magnéticas o tomografías computarizadas para detectar patologías con mayor rapidez que los métodos tradicionales. En la industria automotriz, podrían ser clave para el desarrollo de sistemas avanzados de asistencia al conductor, mejorando la capacidad de los vehículos autónomos para interpretar su entorno y tomar decisiones en tiempo real. En el sector de la seguridad, los modelos ofrecerían sistemas de vigilancia mejorados capaces de identificar amenazas potenciales con una precisión muy superior, reduciendo así los falsos positivos y permitiendo una respuesta más efectiva. Para el marketing y el comercio, la analítica avanzada de imágenes permitiría una personalización más profunda de la experiencia del cliente, identificando patrones de comportamiento y preferencias a partir de datos visuales.

C. Origen de los datos

Kaggle es una plataforma online ampliamente reconocida que ofrece competiciones de ciencia de datos, hospeda conjuntos de datos públicos y permite a los usuarios publicar y encontrar datos para proyectos de investigación y aplicaciones prácticas. Es un recurso valioso tanto para científicos de datos en formación como para profesionales experimentados, proporcionando un espacio para compartir, colaborar y competir en el análisis de datos y el desarrollo de modelos predictivos [3].

El conjunto de datos “Human Portrait or Not” alojado dentro de esta plataforma, representa una colección de imágenes binarias y en color RGB, con una resolución de 128x128 píxeles, destinadas a tareas de clasificación binaria para determinar si una imagen es un retrato de una persona o no [4]. Este es el resultado de un meticuloso proceso de recopilación de una variedad de fuentes especializadas en imágenes de diversas categorías. Las imágenes de automóviles se extrajeron del Stanford Car Dataset, un compendio de vehículos diseñado para facilitar la investigación en el campo de la visión por computadora [5]. En cuanto a las representaciones de mascotas domésticas, tanto felinas como caninas, se recurrió al ampliamente reconocido desafío Dogs vs. Cats disponible en Kaggle, el cual proporciona un extenso conjunto de imágenes para entrenar algoritmos de clasificación [6].

Para las imágenes de flora, se seleccionaron recursos de ImageNet, uno de los repositorios más exhaustivos para la investigación de reconocimiento de imágenes, ofreciendo una amplia gama de especímenes botánicos capturados en imágenes [7]. La categoría de frutas fue enriquecida con imágenes del Kaggle Fruits Dataset, que ofrece imágenes detalladas de frutas, esenciales para estudios de clasificación y reconocimiento de patrones alimentarios [8]. Asimismo, las imágenes de motocicletas provienen del desafío de Clases de Objetos Visuales de Oxford, que proporciona datos para tareas de reconocimiento de objetos en contextos de tráfico y movilidad [9].

Por otro lado, el análisis de imágenes culinarias se apoyó en el Food Recognition Benchmark proporcionado por AICrowd, un desafío que pone a prueba la habilidad de los algoritmos para identificar y clasificar una gran variedad de platos y alimentos [10]. Para la categoría de fauna, se utilizaron las imágenes de caras de animales del dataset de Animal Faces en Kaggle, que incluye una colección de representaciones de diversas especies animales [11]. Finalmente, las imágenes de retratos humanos se obtuvieron del proyecto de segmentación de retratos alojado en GitHub, que es una iniciativa dedicada a perfeccionar las técnicas de segmentación y clasificación de retratos humanos en imágenes [12].

Cada fuente fue elegida no solo por la calidad y claridad de las imágenes que proporciona sino también por su relevancia y utilidad potencial en el entrenamiento de modelos de aprendizaje profundo. La diversidad de estas imágenes asegura un conjunto de datos amplio y variado, que facilita la investigación y el desarrollo de modelos de clasificación binaria, con aplicaciones prácticas que van desde la identificación personal hasta la automatización de procesos en diversos sectores industriales.

D. Métricas de desempeño

En la evaluación de modelos de aprendizaje automático, las métricas desempeñan un papel crucial al proporcionar una visión cuantitativa de su rendimiento. Estas métricas, como la precisión, la pérdida, el recall y el F1-score, permiten una comparación objetiva y una comprensión más profunda de cómo el modelo se comporta en tareas específicas, como la clasificación [13]:

- **Exactitud (Accuracy):** Proporción de predicciones correctas sobre el total. Es útil cuando las clases están equilibradas.

$$\text{Accuracy} = \frac{\text{Número de predicciones correctas}}{\text{Número total de predicciones}} \quad (1)$$

- **Precisión (Precision):** De todas las predicciones positivas, cuántas son realmente positivas.

$$\text{Precision} = \frac{\text{Verdaderos Positivos (VP)}}{\text{Verdaderos Positivos (VP)} + \text{Falsos Positivos (FP)}} \quad (2)$$

- **Recall (Sensibilidad):** De todos los positivos reales, cuántos fueron identificados correctamente.

$$\text{Recall} = \frac{\text{Verdaderos Positivos (VP)}}{\text{Verdaderos Positivos (VP)} + \text{Falsos Negativos (FN)}} \quad (3)$$

- **Puntuación F1 (F1 Score):** Media armónica entre la precisión y el recall, útil cuando se busca un equilibrio entre estas métricas.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

- **AUC-ROC:** Área bajo la curva del gráfico de Característica Operativa del Receptor, útil para clasificadores binarios.

$$\text{Tasa de FP (TFP)} = \frac{\text{Falsos Positivos (FP)}}{\text{Falsos Positivos (FP)} + \text{Verdaderos Negativos (VN)}} \quad (5)$$

En cuanto a las métricas de negocio, pueden incluir:

- **Tiempo de procesamiento:** Rapidez con la que el modelo procesa una imagen.
- **Costos de implementación y operación:** Cuánto cuesta mantener el modelo en producción.
- **Tasa de adopción del usuario:** En aplicaciones de usuario final, qué tan a menudo se utiliza el modelo.
- **Satisfacción del cliente:** Medida cualitativa de cómo la solución impacta la experiencia del cliente.

El valor mínimo aceptable de las métricas para la clasificación de imágenes en la tarea de identificar retratos humanos dependerá de las exigencias y consecuencias del error en el contexto específico en el que se aplique el modelo. En contextos de seguridad, como el reconocimiento en aeropuertos o sistemas de vigilancia, un alto recall sería esencial para minimizar el riesgo de no identificar a una persona de interés. Por otro lado, en aplicaciones de consumo, como la clasificación de fotos en redes sociales, una alta precisión podría ser preferible para mejorar la experiencia del usuario y evitar marcaciones incorrectas. Además, el tiempo de procesamiento debe ser rápido para facilitar una experiencia de usuario fluida, y los costos operativos deben alinearse con el retorno de inversión esperado. Las métricas y umbrales específicos deben ser cuidadosamente seleccionados para equilibrar la usabilidad, la seguridad y la rentabilidad del modelo.

II. OBJETIVOS

A. Objetivo general

Evaluar el impacto de la computación cuántica en la mejora de la precisión y eficiencia de la clasificación binaria de imágenes utilizando modelos de Redes Neuronales Convolucionales implementados con TensorFlow y PyTorch, e integrando circuitos cuánticos.

B. Objetivos específicos

- Comparar la precisión de la clasificación binaria de retratos utilizando modelos CNN estándar desarrollados con TensorFlow y PyTorch.
- Implementar un circuito cuántico en modelos de CNN para determinar su efectividad en la clasificación de imágenes.
- Medir la eficiencia en términos de tiempo de entrenamiento y convergencia de modelos CNN tradicionales frente a modelos con componentes cuánticos.
- Analizar la capacidad de generalización de los modelos cuánticos y convencionales de CNN en el conjunto de datos de imágenes de retratos.
- Identificar y documentar los desafíos técnicos y limitaciones encontradas al integrar circuitos cuánticos en modelos de aprendizaje profundo para la clasificación de imágenes.

III. DATOS

A. Datos originales

El conjunto de datos utilizado en este trabajo es ‘‘Human Portrait or Not (128×128 Binary and RGB)’’ [4]. Este conjunto de datos está diseñado para una tarea de clasificación binaria, donde el objetivo es distinguir entre imágenes que contienen retratos humanos (‘‘Human Portrait’’) y aquellas que no (‘‘Not’’). Las imágenes han sido procesadas para tener un tamaño uniforme de 128x128 píxeles, lo que facilita su uso en algoritmos de aprendizaje automático y deep learning sin la necesidad de redimensionamiento adicional.

El conjunto de datos incluye dos versiones de las imágenes:

- **Binario:** Imágenes en escala de grises o binarizadas, donde los píxeles tienen solo dos posibles intensidades. Este formato es útil para algoritmos que solo necesitan información básica de forma y estructura.
- **RGB:** Imágenes a color con tres canales (rojo, verde y azul). El formato RGB es adecuado para modelos que pueden aprovechar la información de color para mejorar la precisión de la clasificación.

Como se mencionó en las secciones anteriores, el conjunto de datos es una compilación obtenida de diversas fuentes y abarca una amplia gama de categorías. En concreto, se cuenta con 21 770 imágenes de diversas categorías etiquetadas como ‘‘others’’:

- Imágenes de aviones provienen de la base de datos de objetos visuales de la Universidad de Oxford.
- Imágenes de automóviles se obtuvieron del conjunto de datos de automóviles de la Universidad de Stanford.
- Imágenes de gatos y perros se descargaron de la competencia de Kaggle ‘‘Dogs vs. Cats’’.
- Imágenes de flores se adquirieron de ImageNet, una amplia base de datos de imágenes organizadas según la jerarquía de WordNet.

- Imágenes de frutas se recopilaron de un conjunto de datos en Kaggle que contiene imágenes de frutas.
- Imágenes de motocicletas se obtuvieron también de la base de datos de objetos visuales de la Universidad de Oxford.
- Imágenes de comida provienen del desafío de reconocimiento de alimentos de la plataforma AICrowd en 2022.
- Imágenes de animales se tomaron de un conjunto de datos de rostros de animales en Kaggle.

Y un total de 22 438 imágenes etiquetadas como “portrait”:

- Imágenes de retratos humanos se extrajeron de un proyecto de segmentación de retratos en GitHub.

B. Datasets

Para la ejecución de este trabajo, se utilizaron las imágenes en escala de grises. Estas imágenes están organizadas en dos carpetas: `other` y `portrait`. Además, se dispone de dos archivos CSV, `train.binary.csv` y `test.binary.csv`, que contienen dos columnas: la ruta de la imagen y la categoría correspondiente. En estas columnas, el valor 1 representa la categoría `portrait` y el valor 0, en caso contrario.

El proceso establecido en el flujo de trabajo de `ingenieria de características` involucra varias etapas esenciales para preparar un conjunto de datos para tareas de aprendizaje automático. El conjunto de datos se organiza dividiéndolo en tres subconjuntos distintos: entrenamiento, validación y prueba. Esto es crucial para preparar los datos de manera que se pueda entrenar un modelo de manera efectiva, ajustar sus parámetros y evaluar su rendimiento de forma imparcial. Cada uno de estos subconjuntos se compila y almacena en archivos separados, lo que permite un acceso fácil y sistemático para los pasos de entrenamiento y evaluación. Este enfoque estructurado y metódico asegura que el flujo de trabajo de aprendizaje automático esté bien fundamentado en datos limpios y bien organizados, facilitando la reproducibilidad y la eficiencia en todas las fases de modelado y análisis.

C. Analítica descriptiva

En total se cuenta con 44 208 imágenes, separadas en 22 438 retratos y 21 770 de otras categorías, lo cual comprueba que el conjunto de datos está balanceado como se observa en la figura 1.

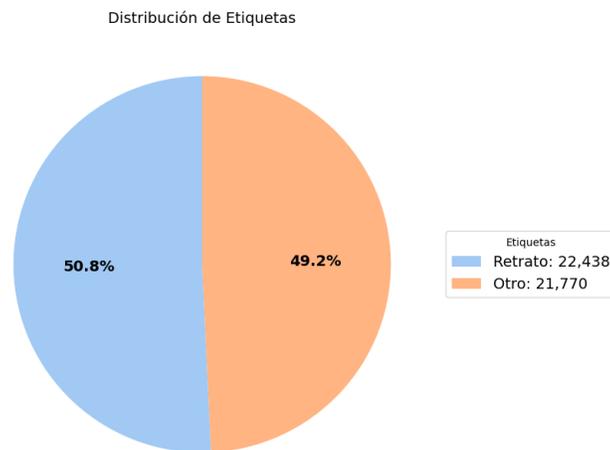


Fig. 1. Distribución de categorías. Elaboración propia.

Adicionalmente, se realiza una búsqueda de anomalías, para comprobar la estructura de los datos. Se encuentra que todas las imágenes son 128x128 y el rango de pixeles de 0 a 255. La muestra de los retratos se evidencia en la figura 2.



Fig. 2. Muestra de retratos. Elaboración propia.

Las imágenes de otras categorías, varían entre vehículos, animales y objetos en general, como se muestra en la figura 3.

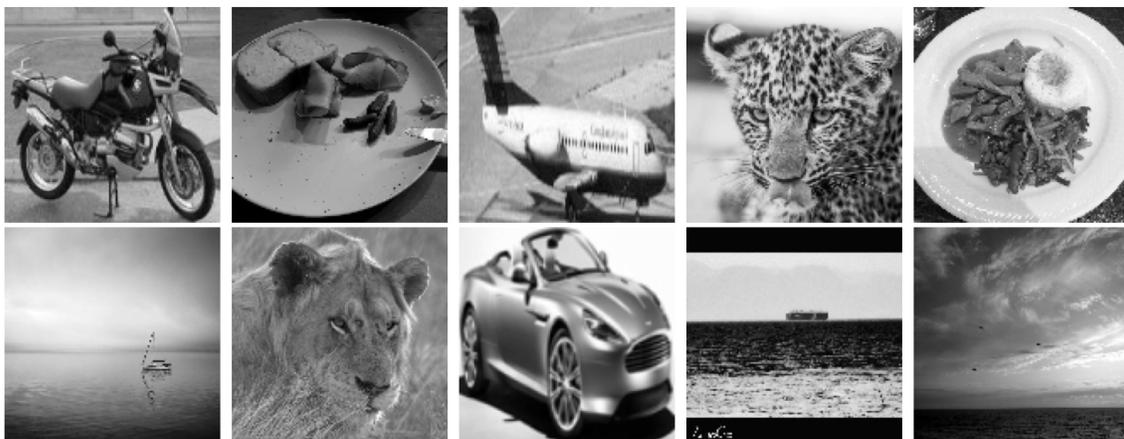


Fig. 3. Muestra de otras categorías. Elaboración propia.

Seguidamente, se analiza las estadísticas de brillo de las imágenes en escala de grises y muestra su histograma 4. De lo que se concluye que todas las muestras siguen una misma distribución de brillo.

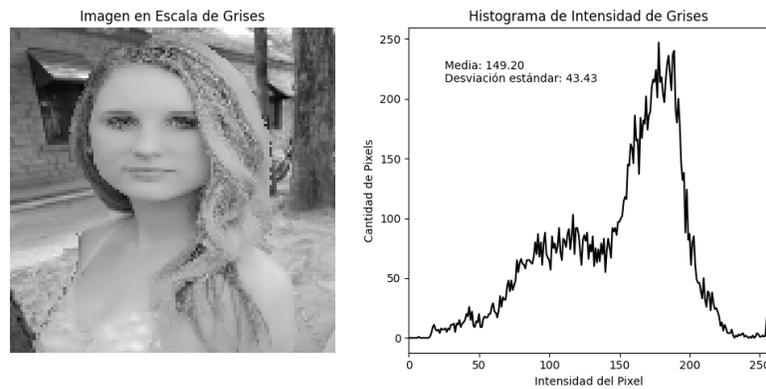


Fig. 4. Intensidad de brillo. Elaboración propia.

La visualización de las dos primeras componentes principales mediante PCA (Fig. 5) no demostró una separación lineal entre las clases del conjunto de datos de imágenes, lo que sugiere la posibilidad de una complejidad intrínseca y patrones no lineales en los datos. Este fenómeno motiva a considerar la posibilidad de que las características distintivas de cada clase puedan requerir métodos de reducción de dimensionalidad que aborden la no linealidad, como t-SNE. La calidad del preprocesamiento y la necesidad de técnicas de modelado más avanzadas también entran en juego, sugiriendo que podría ser beneficioso explorar enfoques más sofisticados, como redes neuronales profundas o máquinas de vectores de soporte con kernels no lineales, para mejorar la capacidad de diferenciación y clasificación dentro del conjunto de datos.

La aplicación de t-SNE (Fig. 6) al conjunto de datos, a pesar de ser una técnica avanzada de reducción de dimensionalidad capaz de capturar relaciones no lineales complejas, no ha logrado revelar una separación clara entre las categorías en una representación bidimensional. Este resultado sugiere que las características distintivas entre las categorías pueden estar entrelazadas en formas que trascienden la separación espacial simple.

Se han identificado la variabilidad y las características esenciales de las imágenes que influyen en la clasificación, así como los desafíos inherentes a la distinción entre retratos humanos y

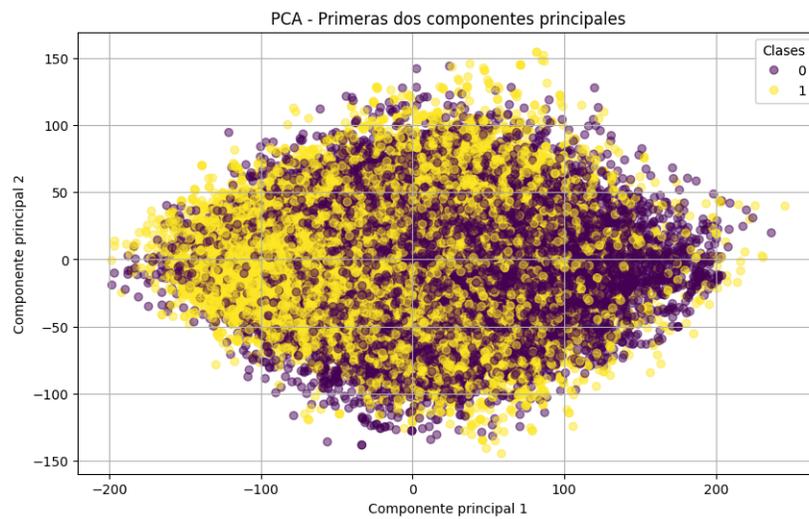


Fig. 5. Análisis de dos componentes PCA. Elaboración propia.

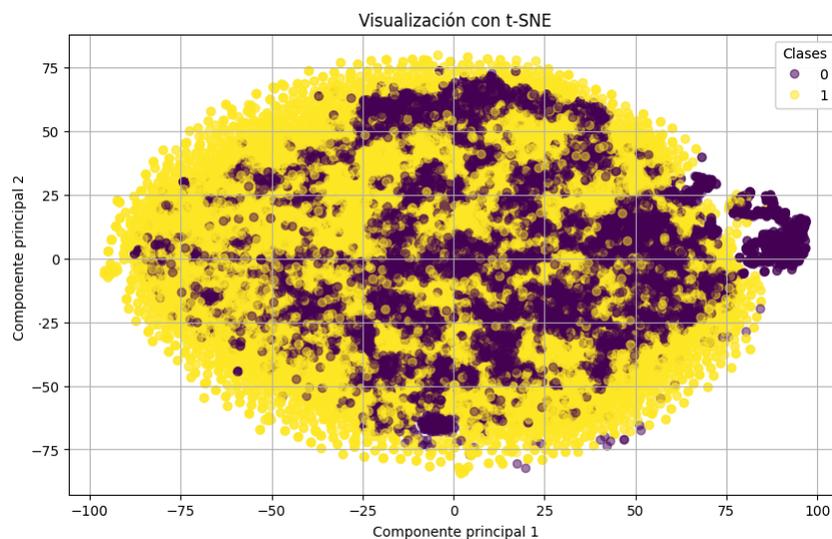


Fig. 6. Análisis de dos componentes t-SNE. Elaboración propia.

otros tipos de imágenes. Esta comprensión detallada sienta una base sólida para el desarrollo posterior de modelos de clasificación avanzados. Con las métricas de rendimiento bien definidas y alineadas con los objetivos del proyecto, estamos en una posición favorable para proceder a la fase de modelado, anticipando que los modelos creados contribuirán significa-

tivamente a mejorar las capacidades de clasificación en el contexto deseado.

IV. PROCESO DE ANALÍTICA

A. Pipeline principal

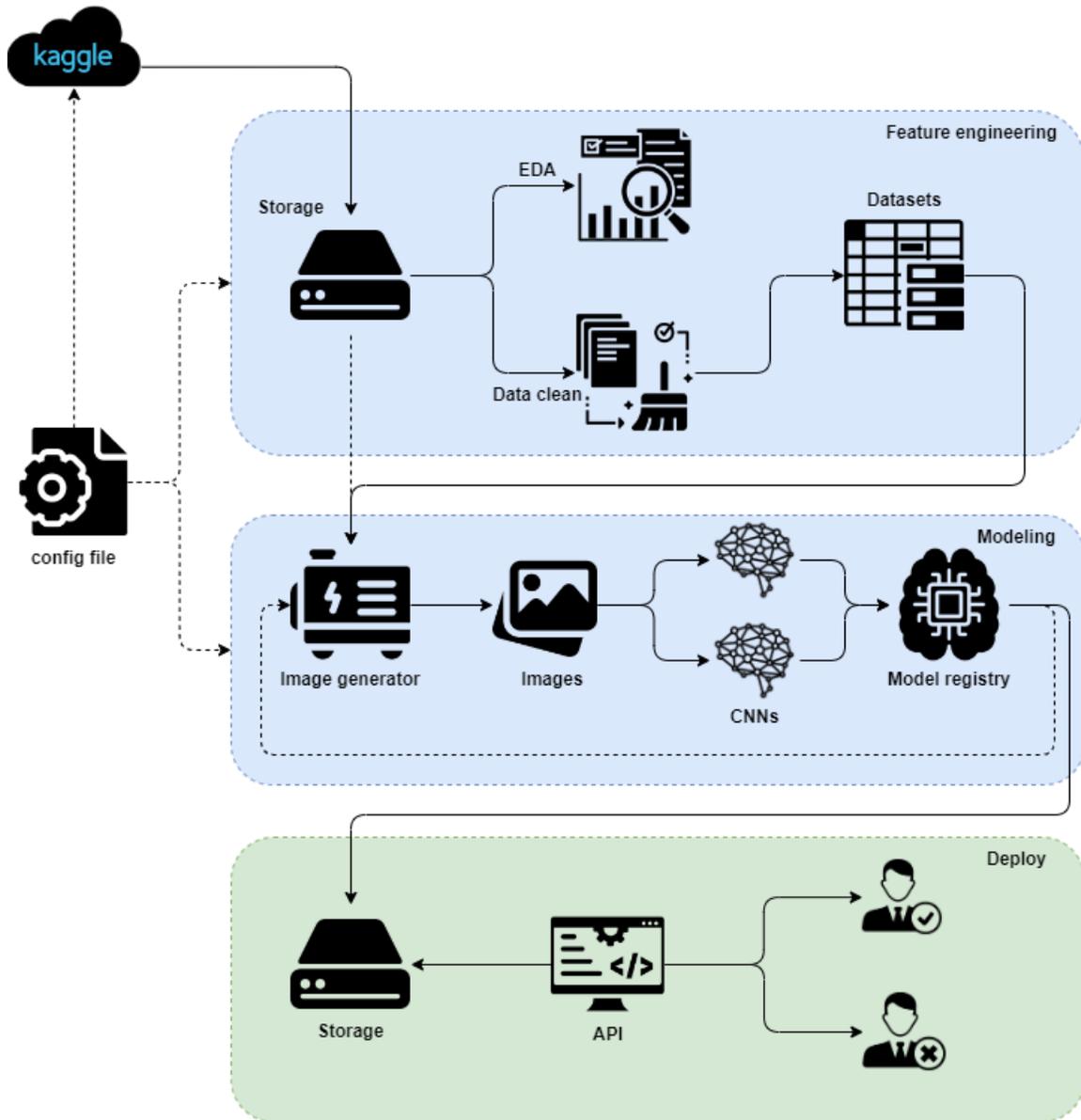


Fig. 7. Pipeline. Elaboración propia.

El pipeline mostrado en la figura 7 muestra el flujo de trabajo utilizado en este proyecto, en el cual todo está parametrizado bajo un archivo de configuración y se explica a

continuación:

1. **Configuración inicial:** Todo comienza con un archivo de configuración que define los parámetros y las variables del entorno del proyecto. Este archivo es esencial para garantizar la reproducibilidad y la consistencia a lo largo del desarrollo y la implementación del modelo.
2. **Almacenamiento de datos:** Los datos están almacenados y organizados en la plataforma kaggle, los cuales se acceden a través del archivo de configuración, donde se parametriza la conexión al API de la plataforma y permite realizar la descarga de datos para su respectivo almacenamiento.
3. **Análisis exploratorio y limpieza de datos (EDA):** En esta etapa se examinan los datos para entender su estructura, contenido y las relaciones entre las variables.
4. **Ingeniería de características y preparación de datasets:** Se transforman los datos crudos en características que los modelos de aprendizaje automático pueden utilizar, lo cual implica la normalización de las imágenes, la extracción de características significativas y la división de datos en conjuntos de entrenamiento, validación y prueba.
5. **Generación de imágenes:** Utilizando generadores de imágenes, se aplican técnicas de aumento de datos como rotación, cambio de escala y volteo para incrementar artificialmente la diversidad de los datos de entrenamiento y mejorar la capacidad del modelo para generalizar a datos no vistos.
6. **Modelado con CNN:** Se seleccionan y entrenan las redes neuronales convolucionales con los datos preparados. Estos modelos son capaces de captar características jerárquicas en las imágenes, lo que es fundamental para tareas de clasificación visual.
7. **Registro de modelos:** Los modelos entrenados se registran en un sistema de control de versiones, donde se pueden gestionar diferentes versiones y se facilita el seguimiento de su rendimiento a lo largo del tiempo.

8. **Despliegue:** Una vez entrenados y evaluados, los modelos seleccionados se despliegan a través de una API en un entorno de producción. Esto permite que aplicaciones cliente interactúen con el modelo, enviando nuevas imágenes y recibiendo predicciones en tiempo real.
9. **Interacción con el usuario final:** Los usuarios finales interactúan con el modelo a través de aplicaciones cliente que utilizan la API para obtener predicciones, lo que permite aplicar el modelo en escenarios reales como la autenticación de identidad o la catalogación de imágenes en plataformas en línea.

Este pipeline refleja un enfoque sistemático y estructurado para construir y desplegar un modelo de clasificación de imágenes, asegurando que cada etapa del desarrollo se realice de manera eficiente y controlada.

B. Preprocesamiento

Se definen dos funciones importantes para el procesamiento de las imágenes. La función `image_generator` se encarga de definir y configurar el aumento de datos, un proceso esencial para ampliar el conjunto de entrenamiento y mejorar la generalización del modelo. Según el modo especificado (“train” ó “test”), esta función aplica técnicas como la rotación, el cambio de escala, el volteo horizontal y ajustes de brillo a las imágenes de entrenamiento, mientras que para las imágenes de prueba, normalmente solo se realiza una reescalación para mantener la consistencia de los valores de los píxeles.

Por otro lado, `flow_generator` toma el conjunto de datos estructurado en un `DataFrame` y lo combina con las configuraciones definidas en `image_generator`. Esta función utiliza `flow_from_dataframe` para crear un generador que alimentará las imágenes al modelo durante el entrenamiento y la validación en lotes. Estos lotes son grupos de imágenes que se procesan juntas, lo que permite manejar eficientemente grandes conjuntos de datos y facilita la computación en paralelo durante el entrenamiento del modelo. Además, `flow_generator` asegura que cada imagen se ajuste al tamaño de entrada requerido por el modelo y que las

etiquetas sean procesadas en el formato correcto para la clasificación binaria.

El uso conjunto de estas dos funciones proporciona un flujo de trabajo automatizado y eficiente para el preprocesamiento de imágenes, lo que es fundamental para el rendimiento óptimo de cualquier modelo de clasificación de imágenes en aprendizaje profundo.

C. Modelos

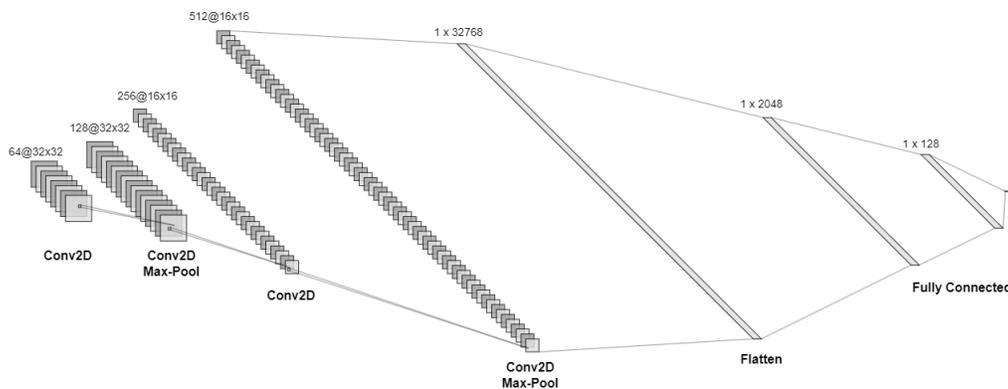


Fig. 8. Arquitectura CNN con TensorFlow. Elaboración propia.

Tensorflow clásico: La red neuronal convolucional (CNN) construida usando TensorFlow (Fig. 8) es una arquitectura secuencial diseñada para la clasificación de imágenes. Comienza con una capa Conv2D que tiene 64 filtros de 3x3, diseñada para extraer características de bajo nivel, como bordes y texturas, de la imagen de entrada, manteniendo el tamaño de la imagen gracias al relleno `same`. A continuación, la capa `BatchNormalization` estabiliza el aprendizaje normalizando las activaciones de la capa anterior, seguida de una capa `Activation` que introduce no linealidades al modelo usando la función `ReLU`, permitiendo a la red aprender patrones complejos.

Para mitigar el riesgo de sobreajuste, se introduce una capa `Dropout` que, al azar, pone a cero un conjunto de activaciones durante el entrenamiento. Este proceso, que utiliza un ratio

de 0.5, significa que la mitad de las “conexiones” se desactivan temporalmente, lo que obliga a la red a aprender patrones redundantes, aumentando su robustez.

A medida que la red se profundiza con más capas **Conv2D**, aumenta el número de filtros (128, 256, y 512), permitiendo a la red extraer características de alto nivel. Cada nueva capa convolucional está seguida de su propia normalización por lotes y activación **ReLU**. Después de ciertas capas convolucionales, se utiliza **MaxPooling2D** para reducir las dimensiones espaciales de las representaciones de características, lo que también contribuye a la reducción de la complejidad computacional y al control del sobreajuste.

Después de las capas convolucionales, la capa **Flatten** convierte las matrices tridimensionales de las características en vectores unidimensionales, preparando el terreno para el procesamiento a través de capas densas (completamente conectadas). Una gran capa **Dense** con 2048 unidades actúa como un clasificador que puede manejar la compleja combinación de características aprendidas, seguida de normalización y activación **ReLU**.

Finalmente, se añaden capas **Dense** más pequeñas para seguir refinando las decisiones de la red, cada una con su propia normalización y activación **ReLU**, y una capa final **Dense** con una sola unidad y una activación sigmoide, la cual produce un valor entre 0 y 1 para representar la probabilidad de que la imagen de entrada sea un retrato humano.

El modelo se compila con un optimizador específico, una función de pérdida adecuada para clasificación binaria y un conjunto de métricas para evaluar el rendimiento durante el entrenamiento y la validación. Este diseño estratificado permite al modelo aprender jerarquías de características y realizar clasificaciones precisas.

La incorporación de la regularización L2 en la arquitectura de la CNN desarrollada con TensorFlow es una estrategia deliberada para mejorar la generalización del modelo, haciéndolo más resistente al sobreajuste, especialmente cuando se manejan conjuntos de datos de gran

tamaño o con mucha variabilidad. La regularización L2 penaliza los pesos grandes en la función de pérdida durante el entrenamiento, promoviendo así pesos más pequeños y menos extremos. Este enfoque es esencial para que la red no dependa de características específicas del conjunto de entrenamiento, permitiendo que el modelo mantenga un alto rendimiento cuando se expone a nuevos datos. Al ajustar la tasa de regularización, se equilibra la complejidad del modelo con su capacidad de generalización, optimizando el rendimiento en datos no vistos y asegurando que el modelo sea robusto y confiable en su despliegue práctico.

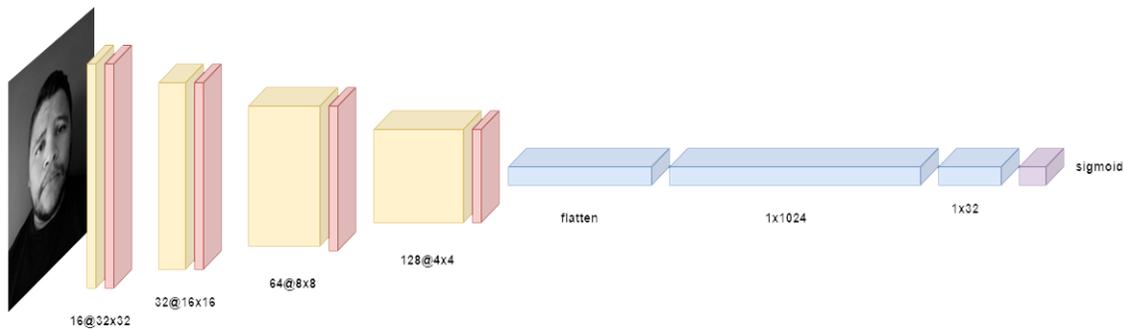


Fig. 9. Arquitectura CNN con PyTorch. Elaboración propia.

PyTorch clásico: La arquitectura del modelo convolucional presentada en la figura 9, plantea una arquitectura más sencilla que la de TensorFlow. La red inicia con una capa convolucional de 16 filtros, seguida por una normalización de lote y una capa de MaxPooling que reduce la dimensionalidad espacial a la mitad. Este patrón se repite, incrementando los filtros a 32 y 64, y finalmente 128, cada uno seguido por MaxPooling.

Después de la extracción de características, la red aplanar los mapas de características y los transfiere a través de una serie de capas lineales con unidades de Dropout intercaladas para reducir el riesgo de sobreajuste. La última capa lineal reduce la salida a un solo valor, procesado por una función sigmoide que determina la probabilidad de pertenencia a la clase de interés. Este diseño refleja un enfoque clásico en las CNN, donde la capacidad de generaliza-

ción se incrementa gradualmente a través de capas sucesivas, refinando la representación de la información visual para la tarea de clasificación.

Comparada con la arquitectura anterior en TensorFlow, que también empleaba una estructura de capas convolucionales y densas con Dropout y regularización L2, la red de PyTorch comparte principios similares en términos de diseño secuencial y estrategias de regularización. Sin embargo, las implementaciones pueden diferir en sintaxis y en algunas operaciones específicas de backend, propias de cada framework de aprendizaje profundo. Ambas buscan equilibrar la capacidad de captura de características complejas con la necesidad de generalizar bien a nuevos datos.

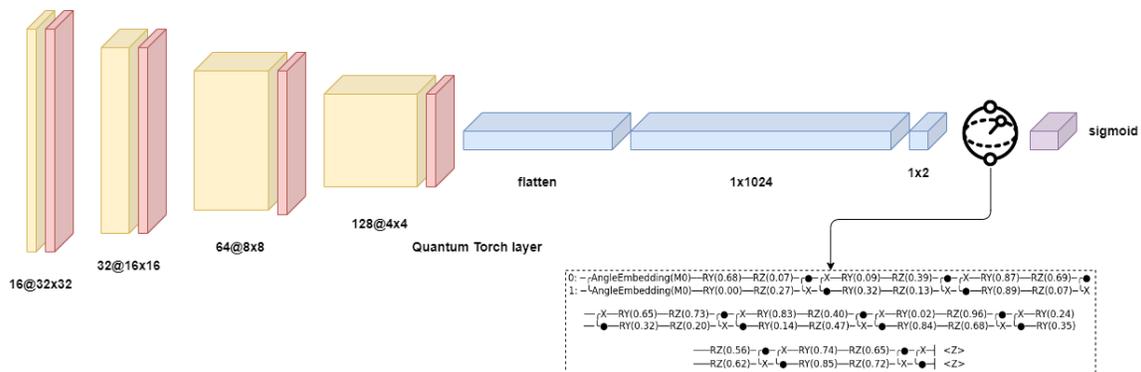


Fig. 10. Arquitectura CNN híbrida. Elaboración propia.

PyTorch híbrido: El modelo híbrido, se basa en la arquitectura del modelo de PyTorch, agregando un circuito cuántico en la última capa oculta construida con PennyLane. La parte de la CNN del modelo consta de varias capas convolucionales y de normalización por lotes, seguidas por capas de max-pooling y dropout. Estas capas están estructuradas para procesar datos de entrada, típicamente imágenes o matrices bidimensionales, extrayendo características relevantes a través de la convolución y reduciendo gradualmente la dimensión espacial de los datos.

Tras procesar los datos a través de la CNN, la información se pasa a un circuito cuántico parametrizado. Este circuito, utiliza técnicas de codificación cuántica para mapear los datos procesados en un estado cuántico. Luego aplica una serie de operaciones cuánticas, incluyendo rotaciones controladas por los parámetros `weights RY` y `weights RZ`, y puertas CNOT para entrelazar los qubits.

La estructura del circuito se explica a continuación:

1. Codificación de Ángulo: Usa `qml.AngleEmbedding` para codificar los datos de entrada en los estados de los qubits. Esta técnica implica girar cada qubit por un ángulo proporcional a los valores del tensor de entrada.
2. Capas de Rotación: Aplica compuertas de rotación RY y RZ en cada qubit. Los pesos RY y RZ determinan los ángulos de estas rotaciones, permitiendo que el circuito se ajuste durante el entrenamiento de la red.
3. Entrelazamiento: Introduce compuertas CNOT entre pares sucesivos de qubits para entrelazarlos. El entrelazamiento es una propiedad clave de los sistemas cuánticos y permite que los qubits influyan entre sí, lo que puede ser crucial para la capacidad de procesamiento del circuito.
4. Medidas: Finalmente, el circuito mide la expectativa de la observable `PauliZ` en cada qubit. Estos valores de expectativa se utilizan como salida del circuito cuántico y se integran en la red neuronal híbrida.

El circuito cuántico funciona en dos qubits y se repite a través de varias capas (ocho en este caso). Cada capa consiste en rotaciones cuánticas específicas seguidas por operaciones de entrelazamiento, lo que permite al circuito explorar un espacio de soluciones complejo y potencialmente encontrar representaciones más eficientes de los datos para la tarea de clasificación.

D. Métricas

La precisión (accuracy) es una métrica particularmente adecuada para evaluar estos modelos de clasificación binaria de imágenes, especialmente dado que se está trabajando con un conjunto de datos balanceado. En este escenario, donde cada clase está representada por igual, la precisión ofrece una evaluación clara y directa del modelo, midiendo efectivamente la proporción de predicciones correctas frente al total de predicciones realizadas. Esta métrica proporciona una comprensión intuitiva del rendimiento del modelo, permitiendo una comparación objetiva entre los enfoques tradicionales y cuánticos en la clasificación de imágenes.

Para las métricas de negocio, se podrían medir aspectos como la velocidad de procesamiento de las imágenes, que es crítica para aplicaciones en tiempo real, y la eficiencia en el uso de recursos computacionales, relevante para el coste operativo. Estas métricas se derivan de la infraestructura y la logística de implementación, y son fundamentales para evaluar la viabilidad comercial del modelo. El seguimiento continuo de estas métricas durante la fase de pruebas proporciona una visión directa del impacto del modelo en los procesos de negocio relevantes.

V. METODOLOGÍA

A. Baseline

En la primera iteración del modelo de clasificación de imágenes, se estableció una línea base utilizando una arquitectura CNN clásica, de esta forma, tener una base para comparar con la introducción de circuito cuánticos. Los resultados mostraron una precisión de entrenamiento consistentemente alta que superaba el 98 %, mientras que la precisión de validación experimentaba una ligera variabilidad. Estos resultados al final son presentados en un API.

Entre los problemas técnicos abordados se encontró la necesidad de balancear la complejidad del modelo con la capacidad de generalización y la gestión de la alta variabilidad en la pérdida de validación. Para combatir estos desafíos, se implementaron técnicas de regularización y ajustes a los hiperparámetros, como la tasa de aprendizaje y el tamaño del lote, buscando un entrenamiento más estable y predecible. Estos cambios apuntaban a mejorar la robustez del modelo frente a datos no vistos y a reducir el sobreajuste indicado por la divergencia entre la precisión de entrenamiento y validación.

B. Validación

El proceso de validación del modelo de clasificación comenzó con la segregación cuidadosa de los datos disponibles en conjuntos especializados para entrenamiento, validación y pruebas. Inicialmente, se seleccionó un porcentaje específico de los datos de entrenamiento original para establecer un conjunto de validación. Esta separación tiene el propósito de proporcionar un conjunto de datos imparcial que se utilizará para evaluar la capacidad del modelo para generalizar más allá de la información con la que fue entrenado, permitiendo así ajustes y mejoras antes de la evaluación final.

En un esfuerzo por mantener la integridad de los datos de prueba y asegurar que reflejaran escenarios del mundo real, una parte de ellos se destinó también a la validación. Esta

práctica es crucial para mitigar cualquier sobreajuste que podría surgir durante la fase de entrenamiento del modelo. La división de los datos se llevó a cabo de forma aleatoria para garantizar la variabilidad y la representatividad en los conjuntos resultantes, evitando sesgos que podrían comprometer la fiabilidad del proceso de validación.

Además, se implementó un mecanismo para fijar la aleatoriedad de la partición de datos a través de una semilla de números aleatorios. Esto aseguró la reproducibilidad del experimento, un aspecto fundamental para validar el rendimiento del modelo de forma consistente en estudios futuros o intentos de replicación. Con estos pasos, se establecieron las bases para una validación meticulosa y se promovió la transparencia y la precisión en la evaluación del modelo.

En los circuitos cuánticos, las iteraciones de ajuste y mejora son fundamentales para optimizar el rendimiento del modelo. Cada iteración implica un delicado equilibrio de ajustar los parámetros del circuito, como las rotaciones y las operaciones de entrelazamiento, y evaluar el impacto de estos cambios en la precisión y eficiencia del modelo. El proceso iterativo no solo implica la modificación de los parámetros del circuito, sino también la experimentación con diferentes arquitecturas de circuito y estrategias de entrelazamiento. Esto es especialmente crítico en tareas como la clasificación y la predicción, donde una configuración óptima puede significar una mejora sustancial en la precisión. Además, el monitoreo y la adaptación a los problemas de convergencia y sobreajuste durante el entrenamiento son esenciales. Por lo tanto, el proceso iterativo en los circuitos cuánticos no solo es una búsqueda de la configuración óptima del circuito, sino también una exploración continua para comprender mejor la interacción entre la mecánica cuántica y los algoritmos de aprendizaje automático.

VI. RESULTADOS Y DISCUSIÓN

En la etapa de desarrollo de los modelos de clasificación binaria, se inició con las CNN de TensorFlow, donde su primera iteración fue planteada sin regularización. Este enfoque se adoptó para evaluar puramente la capacidad de la arquitectura de la red neuronal en el aprendizaje y generalización basado exclusivamente en los datos proporcionados. Para optimizar el proceso de entrenamiento, se establecieron puntos de control (checkpoints) que salvaguardaban los pesos del modelo en su estado óptimo, definido por el pico de precisión (accuracy) en el conjunto de validación. Adicionalmente, se integró una técnica de parada temprana (early stopping) diseñada para contrarrestar el sobreajuste, interrumpiendo el entrenamiento si no se observaba una mejora en la precisión de validación tras un número predefinido de iteraciones (épocas).

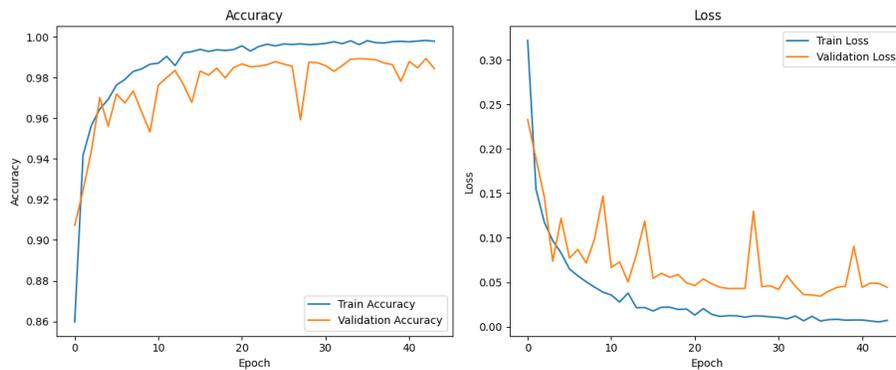


Fig. 11. Evolución de la Precisión en CNN de TensorFlow.

El modelo sin regularización mostró un rápido incremento en la precisión de entrenamiento (Fig. 11), alcanzando un valor cercano al 98% en las primeras épocas. La precisión en el conjunto de validación siguió una tendencia similar, estabilizándose aproximadamente en el 96%, lo que sugiere una alineación adecuada entre los datos de entrenamiento y validación. La pérdida en ambos conjuntos disminuyó significativamente en las primeras épocas, con la pérdida de validación mostrando fluctuaciones menores a medida que avanzaba el entrenamiento. La implementación de early stopping resultó efectiva para detener el entrenamiento

antes de que surgiera el sobreajuste, indicado por la estabilización de la precisión de validación y la pérdida.

En la etapa subsiguiente, se introdujo la regularización L2 en los modelos con el objetivo de examinar su efecto en la capacidad de generalización de la red. A pesar de la implementación de la regularización L2, las métricas de precisión y pérdida mantuvieron una tendencia consistente con las observaciones previas sin regularización como se puede ver en la figura 12. No obstante, se notaron fluctuaciones más pronunciadas en las curvas de validación, lo cual puede ser indicativo de una respuesta del modelo a la complejidad añadida por el término de regularización.

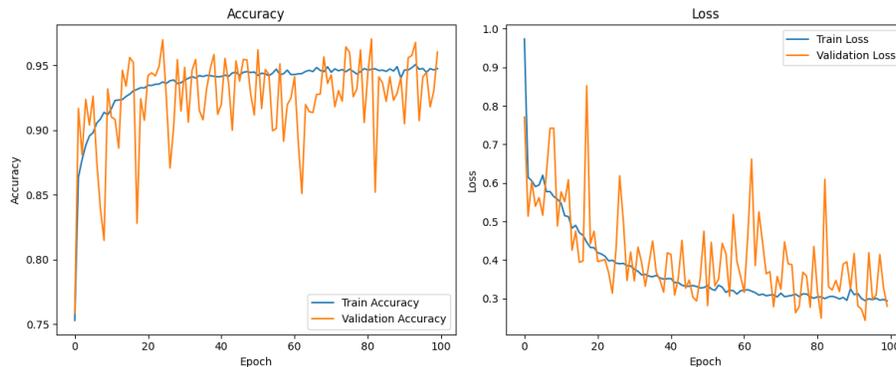


Fig. 12. Análisis de la Pérdida de Entrenamiento con Regularización L2.

La matriz de confusión, como se puede observar en la figura 13, obtenida refleja un desempeño notable del modelo en la tarea de clasificación binaria. Los elementos diagonales principales, que indican las clasificaciones correctas, muestran números significativamente altos tanto para la clase positiva como para la clase negativa, evidenciando una alta tasa de Verdaderos Positivos y Verdaderos Negativos. La presencia de un número reducido de Falsos Positivos y Falsos Negativos sugiere que el modelo ha conseguido un equilibrio adecuado entre sensibilidad y especificidad, lo que implica una buena capacidad de generalización y una baja tasa de error. Estos resultados, cuantitativamente robustos, avalan la efectividad del modelo

desarrollado.

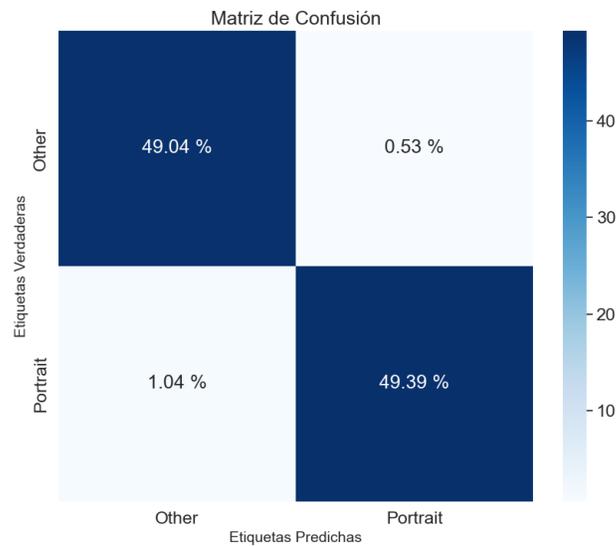


Fig. 13. Matriz de confusión del modelo de TensorFlow.

La red neuronal convolucional implementada en PyTorch muestra un perfil de entrenamiento prometedor, con una precisión que se estabiliza en torno al 98 % para el conjunto de entrenamiento, indicando un aprendizaje efectivo de las características. Por otro lado, la precisión en la validación se mantiene alrededor del 96 % con variaciones moderadas, lo que podría señalar un ajuste preciso pero no perfecto a los datos no vistos (Fig. 14). En cuanto a la pérdida, el modelo exhibe una disminución significativa y sostenida en la pérdida de entrenamiento, con una pérdida de validación que, aunque generalmente baja, muestra picos ocasionales. Estos podrían sugerir respuestas del modelo a anomalías específicas en el conjunto de validación o la necesidad de refinar el proceso de aprendizaje, como la optimización de la tasa de aprendizaje.

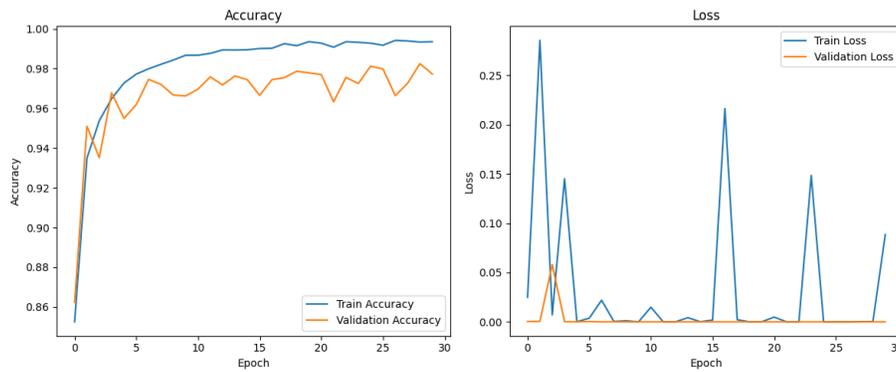


Fig. 14. Evolución de la Precisión en CNN de PyTorch.

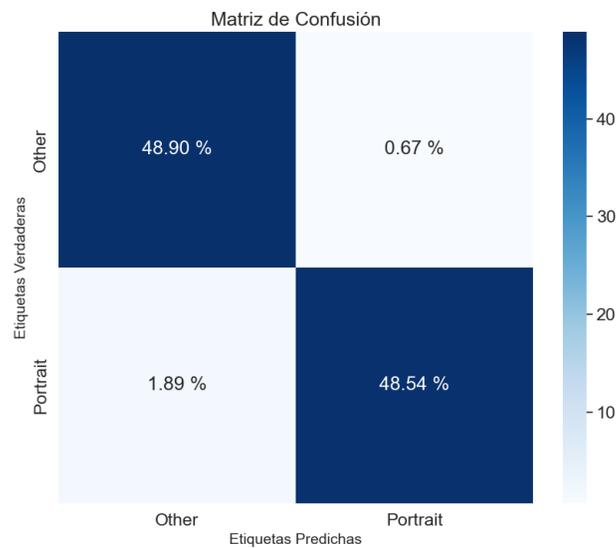


Fig. 15. Matriz de confusión del modelo de PyTorch.

La matriz de confusión (Fig. 15) generada para el modelo anterior muestra un fuerte desempeño en la identificación de ambas clases. La cantidad de verdaderos positivos y verdaderos negativos es considerablemente alta, lo que indica una alta tasa de clasificaciones correctas. Sin embargo, la presencia de falsos negativos y falsos positivos, aunque relativamente baja,

sugiere que hay margen de mejora en la precisión y la exhaustividad del modelo. Este análisis destaca la capacidad del modelo para distinguir efectivamente entre las clases en la mayoría de los casos, pero también subraya la importancia de continuar con el ajuste fino para reducir el número de clasificaciones erróneas.

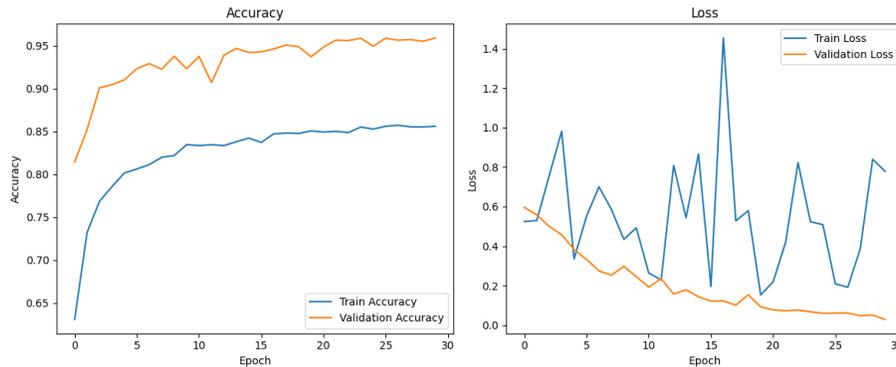


Fig. 16. Evolución de la Precisión en CNN híbrida.

La figura 16 muestra una curva de aprendizaje con tendencias comunes en el entrenamiento de redes neuronales. La línea azul, que representa la precisión de entrenamiento, asciende rápidamente durante las primeras épocas, indicando que el modelo está aprendiendo de manera efectiva del conjunto de entrenamiento. Posteriormente, esta curva se estabiliza, lo que es típico a medida que el modelo comienza a converger. Sin embargo, es atípico que la precisión de validación, mostrada por la línea naranja, sea consistentemente más alta que la de entrenamiento. Este fenómeno podría sugerir que el modelo está mejor optimizado para el conjunto de validación o que los datos de validación no reflejan adecuadamente la complejidad del conjunto de entrenamiento.

Por otro lado, el gráfico de pérdida presenta un comportamiento más errático, especialmente en la curva de validación. La pérdida de entrenamiento disminuye significativamente al principio, indicando un buen ajuste inicial del modelo a los datos de entrenamiento. Sin embargo, la pérdida de validación, aunque generalmente más baja que la de entrenamiento,

exhibe picos significativos y variabilidad a lo largo del entrenamiento. Esto podría ser una indicación de sobreajuste o inestabilidad en el proceso de aprendizaje, posiblemente debido a una tasa de aprendizaje mal seleccionada o falta de regularización adecuada. Es crucial abordar estos picos para mejorar la generalización del modelo, lo que podría implicar ajustar hiperparámetros, aplicar técnicas de regularización más robustas o revisar la distribución de los conjuntos de datos. Finalmente, la matriz de confusión del modelo se presenta en la figura 17.

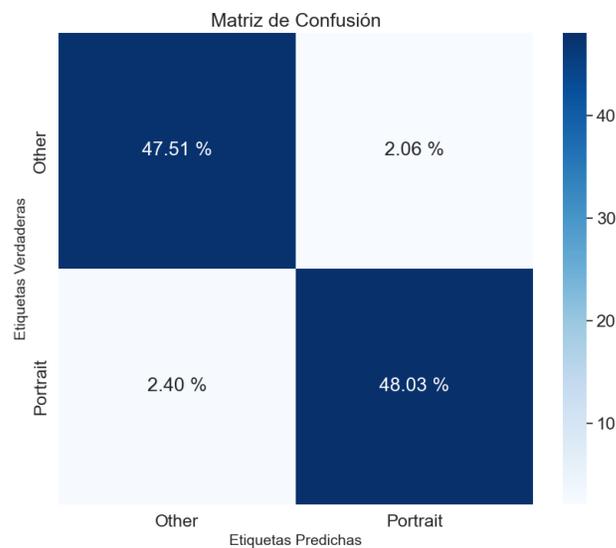


Fig. 17. Matriz de confusión del modelo híbrido.

Link repositorio: <https://github.com/jhanvargas/comparacion-cnn-clasico-cuatico>

VII. CONCLUSIONES

La conclusión de este estudio resalta que, aunque la incorporación de elementos cuánticos en las redes neuronales convolucionales no supera ampliamente a los modelos clásicos en términos de precisión, demuestra un potencial considerable. Los modelos TensorFlow y PyTorch muestran un rendimiento ligeramente superior, lo que refleja la madurez y optimización de las técnicas de aprendizaje profundo clásico. Sin embargo, el modelo híbrido cuántico muestra resultados prometedores, señalando el potencial emergente de la computación cuántica en la mejora de tareas complejas de clasificación de imágenes.

Además de los resultados obtenidos en este estudio, es importante tener en cuenta que la computación cuántica aún se encuentra en una fase temprana de desarrollo. A medida que los sistemas cuánticos se vuelvan más potentes y eficientes, es probable que la incorporación de elementos cuánticos en las redes neuronales convolucionales dé lugar a un rendimiento aún mayor.

En particular, los sistemas cuánticos podrían permitir a las redes neuronales convolucionales aprender patrones más complejos en los datos. Esto se debe a que los sistemas cuánticos pueden representar y procesar información de forma simultánea en múltiples dimensiones, lo que les permite capturar relaciones no lineales entre los datos. Este estudio sienta las bases para futuras investigaciones que podrían desbloquear y explotar plenamente las capacidades únicas de los sistemas cuánticos en el ámbito del aprendizaje automático [I](#).

TABLA I
RESULTADOS DE LOS EXPERIMENTOS CON LOS MODELOS TENSORFLOW, PYTORCH Y
HÍBRIDO

Modelo	Pérdida (Loss)	Precisión (Accuracy)
TensorFlow	0.0494	0.9843
PyTorch	0.0015	0.9744
Híbrido	0.1389	0.9554

VIII. RECOMENDACIONES

Para futuras investigaciones, se recomienda profundizar en el desarrollo de modelos híbridos avanzados que integren circuitos cuánticos con redes neuronales convolucionales, experimentando con conjuntos de datos más extensos y diversos para evaluar la escalabilidad y eficacia en diferentes escenarios. Es vital explorar aplicaciones prácticas específicas donde la computación cuántica pueda ofrecer ventajas únicas, junto con un enfoque en la optimización de circuitos cuánticos y la investigación sobre la robustez y la capacidad de generalización de estos modelos cuánticos y híbridos en condiciones variadas y desafiantes.

REFERENCIAS

- [1] O. Media, “Solving real-world business problems with computer vision,” O’Reilly Radar, 2023. [Online]. Available: <https://www.oreilly.com/radar/solving-real-world-business-problems-with-computer-vision/>
- [2] A. I. Magazine, “What is a Quantum Convolutional Neural Network?” 2020, accessed: fecha-de-acceso. [Online]. Available: <https://analyticsindiamag.com/what-is-a-quantum-convolutional-neural-network/>
- [3] “Kaggle: Your Home for Data Science,” Kaggle Inc., 2023. [Online]. Available: <https://www.kaggle.com>
- [4] “Human Portrait or Not (128x128 Binary and RGB),” Kaggle Dataset, accessed: 2023-11-06. [Online]. Available: <https://www.kaggle.com/datasets/kazushiadachi/human-portrait-or-not-128128-binary-and-rgb>
- [5] “Stanford Car Dataset,” Stanford Artificial Intelligence Laboratory. [Online]. Available: https://ai.stanford.edu/~jkrause/cars/car_dataset.html
- [6] “Dogs vs. Cats,” Kaggle. [Online]. Available: <https://www.kaggle.com/c/dogs-vs-cats>
- [7] “ImageNet,” ImageNet. [Online]. Available: <http://www.image-net.org>
- [8] “Fruit Images for Object Detection,” Kaggle. [Online]. Available: <https://www.kaggle.com/moltean/fruits>
- [9] “The PASCAL Visual Object Classes Homepage,” University of Oxford - Department of Engineering Science. [Online]. Available: <http://host.robots.ox.ac.uk/pascal/VOC>
- [10] “Food Recognition Benchmark 2022,” AICrowd. [Online]. Available: <https://www.aicrowd.com/challenges/food-recognition-benchmark-2022>
- [11] “Animal Faces,” Kaggle. [Online]. Available: <https://www.kaggle.com/datasets/andrewmvd/animal-faces>

- [12] A. Sathyan, “Portrait Segmentation,” GitHub repository. [Online]. Available: <https://github.com/anilsathyan7/Portrait-Segmentation>
- [13] A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, Inc., 2017.