Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Matemáticas y Física

Maestría en Ciencia de Datos

Detección de Imágenes DeepFake con Vision Transformers (ViT)

TESIS que se presenta para obtener el **GRADO** de
MAESTRO EN CIENCIA DE DATOS

Tesis presentada por:
Ing. Óscar Guillermo Retolaza Carlos

Asesor de Tesis:
Dr. Iván Esteban Villalón Turrubiates

Tlaquepaque, Jalisco. Mayo de 2025.

Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Matemáticas y Física

Formato de aprobación de la Maestría en Ciencia de Datos

Título de la Tesis: **Detección de Imágenes DeepFake con Vision Transformers (ViT)**

Autor: **Ing. Óscar Guillermo Retolaza Carlos**

Tesis aprobada para completar todos los requisitos de grado para la Maestría en Ciencia de Datos.

Director de Tesis, **Dr. Iván Esteban Villalón Turrubiates**

Codirector de Tesis, ——

Lector de Tesis, **Mtro. Víctor Hugo Martínez Sánchez**

Lector de Tesis, **Dr. Guillermo Luis Osuna González**

Asesor Académico, **Dra. Rocío Carrasco Navarro**

Tlaquepaque, Jalisco, Mayo de 2025.

Detección de Imágenes DeepFake con Vision Transformers (ViT)

Ing. Óscar Guillermo Retolaza Carlos

Resumen

La inteligencia artificial ha experimentado un crecimiento exponencial en los últimos años. Este avance, conlleva retos éticos y sociales porque el uso indebido de estas tecnologías podría propagar desinformación, crear sesgos o inclusive vulnerar la integridad de personas. En particular, este trabajo aborda el uso de Vision Transformers (ViT), un modelo de deep learning, con la finalidad de apostarle a la creación de herramientas que adviertan sobre la autenticidad del material, especialmente en redes sociales. Esto es crucial para un consumo responsable y seguro de la información.

Por ello, el presente trabajo se centra en evaluar Vision Transformers (ViT) como método de clasificación de imágenes reales frente a falsas, conocidas también como 'DeepFakes', analizando sus fortalezas, limitaciones y vías de mejora. Con ello se busca contribuir a soluciones más efectivas que promuevan un uso ético y confiable de los medios digitales.

Tabla de contenido

Página

Lista de acrónimos 13
1 Introducción 17
2 Justificación 21
2.1 Impactos ético y sociales 21
2.2 Elección del Modelo *Transformers* 21
3 Objetivos 23
3.1 Objetivo general 23
3.1.1 Objetivos específicos 23
4 Marco teórico 25
4.1 Conceptos clave 25
4.2 Descripción general del modelo utilizado. 28
4.2.1 Contexto: Visión por computadora (Computer Vision) 28
4.2.2 Historia de la visión por computadora 28
4.2.3 Modelos transformers 30
4.2.4 'Vision Transformers' (ViT) 31
4.3 Aplicación de ViT en la detección de DeepFakes . . . 33
5 Estado del Arte 35
5.1 Introducción general al estado del arte 35
5.2 Detección de DeepFakes: Actualidad 37
5.2.1 Redes Neuronales Convolucionales (CNN) . 37
5.2.2 Redes Neuronales Adversariales (GAN) . . . 40
5.2.3 Diferencias principales entre ViT y CNN . . . 41
5.3 Conclusión del capítulo 43
6 Desarrollo y Resultados 45
6.1 Métricas utilizadas 46
6.2 Análisis y descripción del conjunto de imágenes . . . 47
6.3 Proceso de implementación 49
6.3.1 Generación de muestras Data Augmentation 51
6.3.2 Preprocesamiento de imágenes con modelo de detección de rostros 51
6.3.3 Primer modelo de detección de rostros: Haarcascade 52

6.3.4 Segundo modelo de detección de rostros: YOLO 53
6.3.5 Ventajas y desventajas del uso de YOLOv8 . 55
6.3.6 Comparación entre ambos modelos de detección de rostros. Haarcascade y YOLOv8 57
6.3.7 Carga de rostros detectados y normalización de imágenes 58
6.3.8 División de imágenes en parches 59
6.3.9 Proyección lineal de cada parche 59
6.3.10 Codificación posicional de los vectores 60
6.3.11 Bloque Transformer: Mecanismo de autoatención 60
6.3.12 Feed Forward Networks (FNN) y conexiones residuales 62
6.3.13 Resumen de un bloque Transformer 63
6.4 Arquitectura e hiper parámetros 64
6.5 Resultados 64
7 Análisis de resultados y conclusiones 69
7.1 Trabajo a futuro 70
8 Bibliografía y Referencias 73

Lista de Figuras

Página

4.1 Inteligencia artificial y sus subáreas 27
4.2 'Vision Transformers' (ViT) 31

5.1 Diagrama de todos los modelos para detección de DeepFakes. 38
5.2 Distribución de los modelos y los números de estudios utilizados. 38
5.3 Diagrama de CNN para clasificación multiclase. 40
5.4 Diagrama de GAN para clasificación multiclase. 41

6.1 Ejemplos de imágenes de personas reales dentro del conjunto de datos. 48
6.2 Balanceo de clases del conjunto de datos. 48
6.3 Histograma de intensidad de color por píxel 49
6.4 Diagrama de flujo para carga y preprocesamiento de imágenes 50
6.5 Ejemplo de técnica de data augmentation 51
6.6 Ejemplo de detección de rostro con modelo Haar cascade de la librería OpenCV. 52
6.7 Ejemplo de detección de rostro con modelo Haar cascade de la librería OpenCV. 53
6.8 Ejemplo rostro de perfil. 53
6.9 Ejemplo rostro a contraluz. 53
6.10 Ejemplo 1 de detección con modelo YOLO 54
6.11 Ejemplo 2 de detección con modelo YOLO 55
6.12 Ejemplo 3 de detección con modelo YOLO 55
6.13 Dataset y su etiqueta. 58
6.14 Ejemplo ilustrativo de división de una imagen en parches. 59
6.15 Ejemplo ilustrativo de las operaciones matriciales 61
6.16 Modelos y sus hiperparámetros. 65
6.17 Modelo 3 y sus métricas 65
6.18 Modelo 3 y su matriz de confusión 66
6.19 Modelo 3 y sus aprendizajes 66

6.20 Modelo 3 y sus predicciones. Las imágenes sí eran "fake", "real", "fake" en ese orden. 66
6.21 Modelo 3 y sus predicciones. La imagen era "fake" . . . 67

Lista de Tablas

Página

5.1 Comparación entre CNN y Vision Transformer (ViT) . . 42

6.1 Comparativa entre Haarcascade y YOLOv8 para detección de rostros 57

Lista de acrónimos

AI: Artificial Intelligence. Inteligencia Artificial. Campo de estudio enfocado en el desarrollo de sistemas capaces de realizar tareas que requieren inteligencia humana.

CNN: Convolutional Neural Network. Red Neuronal Convolucional. Tipo de red neuronal diseñada para procesar datos con una estructura de grilla, como las imágenes.

DNN: Deep Neural Network. Red Neuronal Profunda. Red neuronal con múltiples capas ocultas que permite modelar funciones altamente complejas.

FFN: Feed Forward Network. Red de propagación directa. Arquitectura en la que los datos fluyen en una sola dirección sin ciclos.

GAN: Generative Adversarial Network. Red Generativa Antagónica o Adversarial. Modelo de aprendizaje no supervisado compuesto por una red generadora y una discriminadora en competencia.

GPU: Graphics Processing Unit. Unidad de procesamiento gráfico. Hardware especializado en operaciones paralelas, ideal para entrenamiento de modelos de aprendizaje profundo.

LR: Learning Rate. Tasa de aprendizaje. Hiperparámetro que controla la magnitud del ajuste de pesos durante el entrenamiento.

MHSA: Multi-Head Self-Attention. Autoatención Multicabeza. Mecanismo en Transformers que permite al modelo enfocarse en múltiples partes de la entrada simultáneamente.

MLP: Multi-Layer Perceptron. Perceptrón multicapa. Red neuronal totalmente conectada con al menos una capa oculta.

NLP: Natural Language Processing. Procesamiento del lenguaje natural. Área de la IA enfocada en la comprensión y generación de lenguaje humano.

RNN: Recurrent Neural Network. Red neuronal recurrente. Arquitectura que procesa secuencias utilizando retroalimentación para mantener memoria temporal.

ViT: Vision Transformer. Modelo de Transformer adaptado a visión por computadora mediante la división de imágenes en parches.

YOLO: You Only Look Once. Arquitectura de detección de objetos en tiempo real basada en una sola evaluación por imagen.

Con dedicatoria especial a mi familia, ya que su apoyo y palabras de aliento han sido soporte vital para continuar en mi desarrollo personal, profesional y espiritual. Mención con gran agradecimiento al Dr. Iván Villalón, que fue pilar en guiar este trabajo; agradecido por tener la dicha de ser su alumno y de continuar con mi desarrollo académico. Agradezco a PCE Paragon Solutions (Foxconn) y, especialmente, a mi gerente por brindarme respaldo y motivación para continuar mi formación tanto en el ámbito académico como en el laboral. "El héroe no se hace grande durante los períodos de confort. Las ilustres y nobles almas de nuestro mundo se hacen fuertes, valerosas y éticas cuando afrontan resueltamente los embates de la adversidad, la dificultad y la duda. Es, pues, en el momento en el que afrontan su más profunda debilidad cuando tienen la oportunidad de forjar sus mayores fuerzas. El verdadero poder no procede, por tanto, de una vida de comodidad, sino de la del esfuerzo intenso, de la abnegada disciplina y de la actuación exigente en la dirección que tu yo supremo sabe correcta. Para continuar hasta el momento en el que tu dolor cese. Para avanzar cuando deseas abandonar. Para persistir en el instante en el que sientes que desistir es renunciar a pertenecer al ámbito de los grandes guerreros y de los personajes honorables que llevaron a la humanidad a un lugar mejor, alcanzada la invencibilidad". - Robin S. Sharma

1 Introducción

La evolución creciente de la **inteligencia artificial (IA)** ha impactado de forma considerable varios y diversos sectores de la sociedad, facilitando o promoviendo el avance de tecnologías que intentan replicar ciertas habilidades humanas; inclusive algunas han logrado superarlas.

El continuo progreso sin descanso en el desarrollo tecnológico, con el aumento considerable en la capacidad de analizar volúmenes masivos de información, ha posibilitado la resolución de múltiples problemas en distintas áreas, todos referentes a la búsqueda de patrones o relaciones intrínsecas en un océano de datos. Estos problemas pueden abarcar desde la automatización de procesos repetitivos, la clasificación de información, hasta el poder conversar en tiempo real con una inteligencia artificial.

A primera vista, podría sonar como algo totalmente inofensivo; sin embargo, hay algo detrás de este tipo de tecnologías emergentes donde el debate ético toma un papel crucial y fundamental. La utilización inapropiada de estas tecnologías podría generar incertidumbre, desinformación e inclusive afectar a diversos sectores de la población.

La creación o alteración de imágenes de personas reales, suplantándolas parcial o totalmente mediante características falsas, abarca desde alteraciones simples, como el cambio del color de ojos, hasta la generación de videos en los que la persona parece realizar acciones que nunca ocurrieron. Estas alteraciones son comúnmente conocidas como imágenes *DeepFake*.

El incremento reciente de estas tecnologías ha sido impulsado en gran parte por la aparición de modelos generativos de texto como 'ChatGPT' de OpenAI. Estos modelos aprenden a generar respuestas estadísticamente probables basadas en datos históricos previos, o dicho de otra manera, adquieren la capacidad de generar la palabra más probable que sigue basado en todo su proceso de entrenamiento previo, donde pueden ver una inmensa cantidad de datos y así permitiéndoles

producir salidas nuevas pero probables. Es importante destacar que estos modelos no poseen capacidad de razonamiento como aparentan, sino que identifican patrones dentro de los datos, ya sean semánticos, posicionales o jerárquicos, por mencionar algunos, para después proporcionar una salida.

Como ya se mencionó, con la evolución de estas tecnologías generativas, su aplicación se ha expandido más allá de la generación de texto, abarcando también la generación de imágenes, videos e incluso audios. Y, desgraciadamente, estos avances no han sido utilizados únicamente con fines positivos.

La accesibilidad y facilidad de acceso a estos modelos ha promovido mucho la creación y distribución de contenido multimedia manipulado. La sofisticación de estos sistemas, combinada con la facilidad de uso y el acceso prácticamente a unos cuantos clics, ha permitido generar imágenes y videos híperrealistas, que pueden emplearse tanto para el entretenimiento y la producción artística, pero también para fines poco éticos. Ejemplos particularmente preocupantes incluyen la creación de videos pornográficos falsos, en los cuales se utilizan imágenes cotidianas de una víctima para generar cuerpos desnudos hiperrealistas, que luego son explotados comercialmente sin el conocimiento ni consentimiento de la persona afectada. Como se documenta en el artículo publicado por CRUCE ITESO recientemente: [1]

[1] A. Diana, "Deepfake: la violencia machista parece no tener límites - CRUCE," Marzo 2025

En un contexto tan digitalizado, de consumo masivo y de rápida difusión, las consecuencias pueden ser severas tanto para quien consume el contenido como para quien lo sufre. El impacto, al influir considerablemente en la percepción de la realidad de ciertos grupos, comunidades o poblaciones enteras, puede ser peligroso e incalculable.

Ahora bien, aunque la manipulación y edición de contenido no son fenómenos nuevos, la rapidez de difusión en redes sociales o en internet, sumado a la falta de un contexto adecuado y la urgencia prácticamente nula del usuario final por validar la información de manera inmediata, plantean desafíos nuevos y urgentes. En este sentido, el desarrollo de herramientas de detección de contenido manipulado es crucial para garantizar un consumo de información más seguro y confiable.

Por consiguiente, el estudio de estos modelos y su implementación de mecanismos que informen a los usuarios sobre la autenticidad del material que visualizan debería ser un estándar en todos los medios masivos de comunicación, haciendo especial énfasis en redes sociales. La implementación de estas herramientas facilitaría un análisis más

preciso del contenido, brindando a los usuarios mayores recursos para discernir su veracidad y tomar decisiones informadas antes de compartirlo.

Por ello, el propósito de este trabajo de obtención de grado es analizar y comparar diversas tecnologías diseñadas para detectar la manipulación de contenido digital, proporcionando al consumidor herramientas que le permitan evaluar la autenticidad del contenido.

Este trabajo en particular se enfocará en el análisis de una herramienta denominada 'Vision Transformers' para la clasificación de imágenes de personas reales y 'DeepFakes', examinando sus principales ventajas, limitaciones y posibles áreas de mejora. Se espera que este estudio contribuya al desarrollo de soluciones más eficaces, fomentando un consumo de contenido multimedia más ético y responsable.

2 *Justificación*

Contenido

2.1 Impactos ético y sociales 21
2.2 Elección del Modelo *Transformers* 21

2.1 *Impactos ético y sociales*

El uso irresponsable y no medido de las tecnologías generativas representa un riesgo serio para la integridad y seguridad de las personas, especialmente debido a la facilidad de acceso a imágenes personales en redes sociales y a la falta de mecanismos eficaces para proteger sus datos o de avisar al consumidor sobre la veracidad del contenido.

Es fundamental desarrollar sistemas o herramientas que adviertan al usuario cuando el contenido pueda ser falso o ilícito, a fin de prevenir la desinformación y fomentar un entorno digital más seguro.

Este trabajo se propone analizar modelos de detección de *'DeepFakes'* con el fin de evaluar su viabilidad como herramienta para proteger la integridad de las personas, mediante el uso de modelos 'Vision Transformers (ViT)'.

2.2 *Elección del Modelo Transformers*

Los modelos *Transformers* son una arquitectura de aprendizaje profundo propuesta por Vaswani et al. en 2017 [1], desarrollada originalmente para tareas de procesamiento del lenguaje natural. Su principal innovación es el mecanismo de auto-atención, que permite al modelo asignar de forma dinámica la importancia relativa de cada elemento de la entrada, independientemente del orden en que se presente.

Este mecanismo ha demostrado ser eficaz para capturar relaciones complejas y dependencias a largo plazo dentro de los datos. En

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," 6 2017

términos prácticos, el modelo puede identificar automáticamente las partes más relevantes de una secuencia sin verse afectado por la posición de cada elemento.

A diferencia de arquitecturas basadas en recurrencia, como las Redes Neuronales Recurrentes (RNN), los *Transformers* eliminan la dependencia secuencial, lo cual permite la paralelización de los cálculos y el procesamiento eficiente de grandes volúmenes de datos.

Este trabajo busca aprovechar dicha capacidad para la detección de *DeepFakes*, una tarea que requiere identificar patrones sutiles en imágenes sin perder contexto global de la imagen. Por esta razón, se emplearán modelos basados en *Transformers* aplicados a visión por computadora, con el objetivo de clasificar imágenes reales y generadas artificialmente, maximizando su desempeño en grandes conjuntos de datos y su habilidad para extraer características clave.

3 Objetivos

Contenido

3.1 Objetivo general 23
3.1.1 Objetivos específicos 23

3.1 Objetivo general

Analizar y evaluar el desempeño del modelo para clasificación de imágenes, para la detección de imágenes reales y falsas. Además, se expondrán sus ventajas y desventajas en términos de precisión, eficiencia computacional y capacidad de generalización, proponiendo híperparámetros de entrenamiento con el fin de optimizar su desempeño en entornos prácticos.

3.1.1 Objetivos específicos

1. Implementar un modelo basado en 'Vision Transformers' para la clasificación de imágenes reales y falsas, optimizando hiperparámetros de entrenamiento.
2. Implementar un modelo de detección de rostros como preprocesamiento del conjunto de imágenes.
3. Analizar y comparar el desempeño de los modelos mediante métricas cuantitativas como precisión, recall, F1-score, evaluando sus ventajas y desventajas en términos de exactitud, eficiencia computacional y capacidad de generalización.
4. Alcanzar una precisión de al menos 80% en la clasificación de imágenes.

4 Marco teórico

Contenido

4.1 Conceptos clave 25
4.2 Descripción general del modelo utilizado. . . . 28
4.2.1 Contexto: Visión por computadora (Computer Vision) 28
4.2.2 Historia de la visión por computadora 28
4.2.3 Modelos transformers 30
4.2.4 'Vision Transformers' (ViT) 31
4.3 Aplicación de ViT en la detección de DeepFakes 33

4.1 Conceptos clave

Inteligencia Artificial: Rama de la informática orientada al desarrollo de sistemas capaces de realizar tareas que requieren inteligencia humana, como el razonamiento, el aprendizaje y la percepción. Su objetivo es simular la capacidad humana para resolver problemas en distintos contextos[1].

[1] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3rd ed., 2010

Machine Learning: Subárea de la inteligencia artificial que estudia algoritmos capaces de aprender patrones a partir de datos y mejorar su rendimiento con la experiencia, sin ser explícitamente programados para tareas específicas [2]. Entre estos algoritmos destacan: regresión lineal, regresión logística, árboles de decisión, bosques aleatorios, máquinas de soporte vectorial y algoritmos de vecinos más cercanos. Uno de los enfoques más populares es el uso de redes neuronales.

[2] IBM, "What is machine learning?."

Redes Neuronales: Modelos computacionales inspirados en la estructura del cerebro humano, compuestos por nodos (neuronas artificiales) organizados en capas. Estos transforman entradas en salidas mediante funciones matemáticas, aprendiendo relaciones entre los datos [3].

[3] IBM, "What is a neural network?."

Deep Learning: Rama del aprendizaje automático basada en redes neuronales profundas, es decir, con múltiples capas, que permiten representar y aprender automáticamente patrones complejos a partir de grandes volúmenes de datos [4].

Modelo supervisado: Enfoque de aprendizaje automático en el que el modelo se entrena con datos etiquetados, es decir, con ejemplos que incluyen una respuesta conocida. El objetivo es aprender una función que relacione entradas con salidas [5].

Modelo no supervisado: Tipo de aprendizaje automático en el que los datos no están etiquetados. El objetivo es descubrir estructuras o patrones latentes dentro del conjunto de datos, como agrupaciones o asociaciones [6].

Imagen: Representación visual de una escena, generalmente estructurada como una matriz de valores numéricos que codifican la intensidad de luz o color en cada punto. Las imágenes digitales se componen de unidades mínimas llamadas píxeles [7].

Pixel: Unidad básica de una imagen digital. Representa un único punto dentro de la imagen y almacena información sobre color o intensidad lumínica, dependiendo del tipo de imagen (en escala de grises o en color) [8].

Visión por Computadora: Campo de la inteligencia artificial que busca desarrollar algoritmos capaces de interpretar y comprender el contenido de imágenes y videos, permitiendo que las máquinas "vean" e interactúen con su entorno visual [9].

Canales de color: Dimensiones de una imagen que representan componentes individuales del color. En el modelo RGB, por ejemplo, cada imagen está compuesta por tres canales: rojo, verde y azul, los cuales se combinan para formar el color final de cada píxel [10].

Modelos Transformers: Arquitectura de redes neuronales basada en mecanismos de atención, diseñada para procesar secuencias de datos sin necesidad de recurrencia, lo que permite una mayor eficiencia computacional y paralelización [11].

Recurrencia: Principio presente en ciertas arquitecturas, como las redes neuronales recurrentes (RNN), que permite procesar datos secuenciales utilizando salidas anteriores como entradas futuras. Este enfoque modela dependencias temporales, aunque puede causar pérdida de información en secuencias largas [12].

Mecanismo de Auto-atención: Componente central de los Transformers que permite a cada elemento de la entrada considerar todos los

[4] IBM, "What is deep learning?."

[5] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009

[6] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009

[7] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2002

[8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2002

[9] IBM, "What is computer vision?."

[10] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2002

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," 6 2017

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997

demás, asignando pesos según su relevancia contextual. Esto permite capturar relaciones globales entre los datos [13].

Vision Transformers (ViT): Adaptación de la arquitectura Transformer al ámbito de la visión por computadora. Las imágenes se dividen en parches que se tratan como secuencias, permitiendo aplicar los mismos principios que en el procesamiento de texto [14].

Inteligencia Artificial Generativa: Área de la inteligencia artificial enfocada en la generación de contenido sintético —como texto, imágenes, audio o video— a partir de datos existentes, utilizando modelos como redes generativas adversarias (GANs) o modelos de difusión [15].

Clasificación de Imágenes: Tarea en visión por computadora que consiste en asignar una categoría o etiqueta a una imagen con base en su contenido visual, utilizando modelos previamente entrenados con ejemplos etiquetados [16].

Preprocesamiento: Conjunto de transformaciones aplicadas a los datos antes de ser utilizados en un modelo de aprendizaje automático. En el caso de imágenes, incluye operaciones como redimensionamiento, normalización, eliminación de ruido o técnicas de aumento de datos (*data augmentation*) [17].

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," 6 2017

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," 6 2017

[15] Amazon Web Services, Inc., "¿qué es la ia generativa? - explicación de la ia generativa - aws."

[16] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1097–1105, 2012

[17] A. Famili, W.-M. Shen, R. Weber, and E. Simoudis, "Data preprocessing and intelligent data analysis," *Intelligent Data Analysis*, vol. 1, no. 1, pp. 3–23, 1997

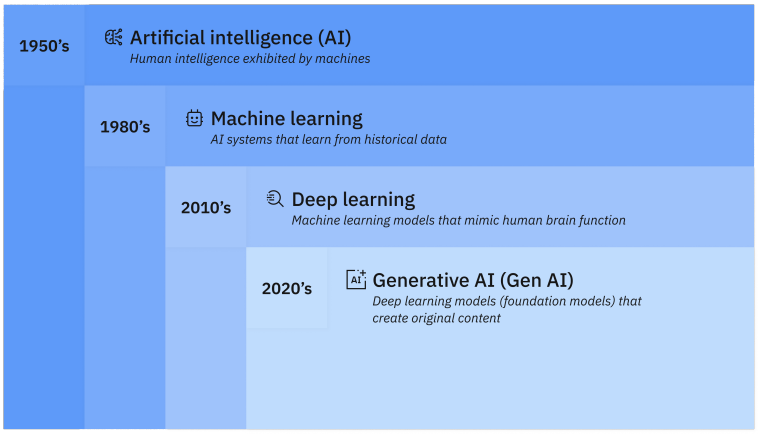

Figura 4.1: Inteligencia artificial y sus subáreas

4.2 Descripción general del modelo utilizado.

El modelo empleado en este trabajo académico se fundamenta, como ya se ha mencionado, en la arquitectura de Vision Transformers (ViT). En contraste con otros modelos que manipulan las imágenes mediante distintas operaciones como las convoluciones, aquí cada imagen se segmenta y se ajusta en vectores.

En esta sección se describirá esta arquitectura, el proceso y la descripción de cómo este modelo posee la capacidad de reflejar o capturar atributos universales que podrían pasar inadvertidos en modelos convencionales.

4.2.1 Contexto: Visión por computadora (Computer Vision)

La visión por computadora es una subdisciplina de la inteligencia artificial que emplea algoritmos de aprendizaje automático, especialmente redes neuronales, para hacer que las computadoras interpreten y extraigan información relevante a partir de imágenes o secuencias de video [18]. Esta tecnología busca replicar de cierta manera la capacidad visual humana, haciendo posible que las máquinas no solo "vean" imágenes, sino que también las analicen y actúen con base en la información extraída.

[18] IBM, "Computer vision," Jan. 2025

Los sistemas de visión por computadora requieren forzosamente ser entrenados con grandes volúmenes de datos que provienen de imágenes. A través de múltiples iteraciones, el modelo es expuesto a numerosos ejemplos del objeto o patrón que se desea reconocer, cada uno acompañado de su etiqueta correspondiente. Con el tiempo, el modelo aprende a identificar diferencias, patrones y regularidades presentes en los datos, refinando así su capacidad de reconocimiento. Es fundamental que el conjunto de entrenamiento sea lo suficientemente amplio y representativo, de modo que el modelo pueda generalizar su conocimiento y reconocer correctamente objetos en imágenes no vistas previamente durante el entrenamiento.

4.2.2 Historia de la visión por computadora

Remontando un poco al origen de la visión por computadora, se rescata el artículo publicado por la empresa 'IBM', donde se menciona que "los esfuerzos por dotar a las máquinas de capacidad visual se remontan a mediados del siglo XX. En 1959, un experimento pionero realizado por neurofisiólogos consistió en exponer a un gato a distintas imágenes para observar respuestas neuronales. Descubrieron que las neuronas

reaccionaban ante estímulos visuales simples, como líneas rectas o bordes, lo cual dio origen a la idea de que el procesamiento visual comienza con la detección de formas básicas. Casi al mismo tiempo, se desarrollaron las primeras tecnologías para escanear imágenes, permitiendo a las computadoras digitalizar y almacenar representaciones visuales".

En el artículo continúa haciendo mención a que un avance importante se logró en 1963, cuando fue posible transformar imágenes bidimensionales en representaciones tridimensionales mediante procesamiento computacional. En esa misma década, la inteligencia artificial emergió como disciplina académica, y con ella surgieron los primeros intentos de resolver el problema de la visión artificial.

A partir del año 2000, el enfoque principal de la visión por computadora se desplazó hacia el reconocimiento de objetos. En 2001 surgieron las primeras aplicaciones de reconocimiento facial en tiempo real, y durante la década se estandarizó el uso de conjuntos de datos anotados para entrenamiento de modelos. En 2010, se publicó el conjunto de datos *ImageNet*, con millones de imágenes clasificadas en mil categorías, lo que impulsó notablemente el desarrollo de modelos de aprendizaje profundo.

En 2012, un equipo de la Universidad de Toronto presentó un modelo CNN llamado *AlexNet*, que participó en una competencia de clasificación de imágenes y redujo drásticamente el margen de error respecto a trabajos anteriores. Este avance marcó un punto de inflexión en la visión por computadora, consolidando el uso de redes neuronales profundas como el enfoque dominante hasta la actualidad. [19].

[19] IBM, "Computer vision," Jan. 2025

Algunos ejemplos del uso y aplicación de la visión artificial mencionados también en la fuente:

1. IBM empleó la visión artificial para crear My Moments para el torneo de golf Masters 2018. IBM Watson vio cientos de horas de material de Masters y pudo identificar las imágenes (y los sonidos) de tomas significativas. Seleccionó estos momentos clave y los entregó a los fanáticos como resúmenes de momentos destacados personalizados.

2. Google Translate permite a los usuarios apuntar con la cámara de un teléfono inteligente a un letrero en otro idioma y obtener casi de inmediato una traducción en su idioma preferido.

3. El desarrollo de vehículos autónomos se basa en la visión artificial para dar sentido a la entrada visual de las cámaras de un automóvil y

otros sensores. Es esencial para identificar otros automóviles, señales de tráfico, marcadores de carril, peatones, bicicletas y toda la demás información visual que se encuentra en la carretera.

4. IBM está aplicando tecnología de visión artificial con asociados, como Verizon, para llevar la IA inteligente al límite y ayudar a los fabricantes automotrices a identificar defectos de calidad antes de que un vehículo salga de fábrica.

4.2.3 *Modelos transformers*

Este modelo, descrito por primera vez en el 2017 por Google, [20] fue creado principalmente con el objetivo de traducir texto de una manera más enriquecida, tomando en consideración el contexto en la traducción y haciéndolo de forma más eficiente mediante mecanismos llamados "Cabezas de atención" que cumplen con la función de asignar importancia a ciertas partes del conjunto de datos, convirtiendo así el texto en vectores.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," 6 2017

Los Transformers son un tipo de arquitectura de una red neuronal. Se han expandido a varias aplicaciones, incluidos los modelos de lenguaje grande (LLM), visión artificial, procesamiento de audios, entre otros.

Las redes neuronales convencionales que manejan secuencias de información generalmente emplean un patrón de arquitectura de codificador/decodificador. El codificador analiza y maneja la secuencia completa de datos de entrada, tal como una frase en inglés, y la convierte en una representación matemática compacta. Esta ilustración es una síntesis que encapsula el núcleo de la entrada. Posteriormente, el decodificador extrae este resumen y, gradualmente, produce la secuencia de salida, que podría ser la misma frase traducida al francés.

Este proceso se lleva a cabo de manera secuencial, lo que implica que debe procesar cada palabra o segmento de la información una después de otra. El procedimiento es pausado y puede desvanecer algunos detalles más precisos en distancias extensas; es decir, que la primera frase que se introdujo en la secuencia pudiera perder total relevancia con respecto a la última oración procesada.

Por su lado, los modelos 'Transformers' alteran este procedimiento; en vez de manejar los datos de manera secuencial, el mecanismo posibilita al modelo examinar diversas secciones de la secuencia simultáneamente y establecer qué secciones son las más relevantes.

El modelo puede enfocar más su atención en los fragmentos de información que él considera más "relevantes" y los fusiona para realizar predicciones más acertadas de las salidas. Este procedimiento potencia la eficiencia de los transformadores, lo que les facilita la formación en grupos de datos más amplios. Además, resulta más efectivo, particularmente en textos extensos donde el contexto remoto

puede afectar el sentido de lo que sigue.

4.2.4 *'Vision Transformers' (ViT)*

Los Vision Transformers (ViT) fueron introducidos por Dosovitskiy et al. (2020) como un enfoque alternativo a las redes neuronales convolucionales (CNN) en tareas de clasificación de imágenes [21]. A diferencia de las CNN, que procesan imágenes a través de operaciones convolucionales locales, ViT adopta un enfoque basado en autoatención para modelar dependencias globales dentro de la imagen.

[21] Y. Huo, K. Jin, J. Cai, H. Xiong, and J. Pang, "Vision transformer (vit)-based applications in image classification," in *2023 IEEE 9th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, pp. 135–140, 2023

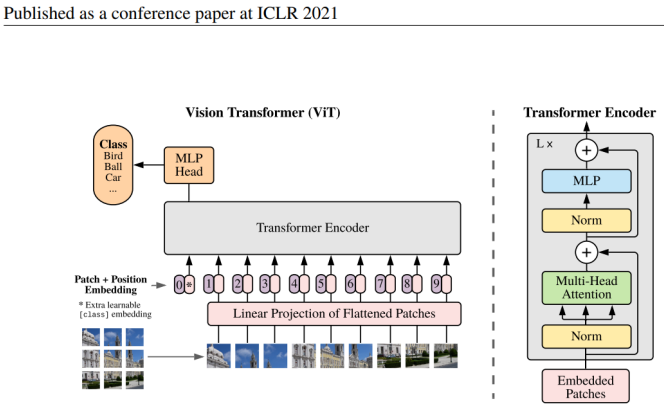

Figura 4.2: 'Vision Transformers' (ViT)

El diseño de ViT se inspiró en los modelos Transformers utilizados en procesamiento del lenguaje natural (NLP), cuya arquitectura demostró una capacidad superior para capturar relaciones de largo alcance en secuencias de texto [22]. En el caso de la visión computacional, este mecanismo permite al modelo analizar toda la imagen a la vez, en lugar de centrarse en regiones pequeñas como lo hacen las CNN con filtros de convolución.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," 6 2017

Los 'Vision Transformers' (ViT) usan la estructura del transformer para labores de categorización de imágenes. En vez de considerar una imagen como un cúmulo de píxeles, perciben los datos como una serie de vectores de tamaño constante, de manera parecida a cómo se manejan las palabras en una frase. Es decir, como si a cada pixel se le asignara un número dependiendo de su color (RGB, Red-Green-Blue) y se concatenara en un vector bidimensional.

Cada parche se despliega, se incrusta de manera secuencial y posteriormente se maneja secuencialmente a través del codificador de transformador normal. Se incorporan vectores de posición para preservar la información espacial. Este método de autoatención global posibilita que el modelo registre las conexiones entre cualquier par de parches, sin importar su ubicación.

Uno de los principales desafíos de las CNN tradicionales es la pérdida de información contextual cuando los primeros datos ingresados quedan en desventaja con respecto a los últimos en redes muy profundas. Los Transformers resuelven este problema mediante el uso de autoatención multicabeza (Multi-Head Self-Attention, MHSA), que evalúa la importancia de cada parte de la imagen en relación con las demás.

A continuación, se describen las características y pasos clave de los (ViT):

1. **División de la imagen en parches (patching)**
   La imagen de entrada se divide en pequeños bloques regulares (por ejemplo, de 16×16 píxeles), los cuales son aplanados y proyectados linealmente a vectores de dimensión fija. Cada parche se considera un *token*.

2. **Codificación posicional**
   Dado que la arquitectura Transformer no posee un mecanismo implícito para preservar la estructura espacial, se añade a cada vector de parche una codificación posicional, que contiene información sobre su ubicación original. Esto permite al modelo aprender relaciones espaciales relativas.

3. **Proyección a representaciones de entrada**
   Cada parche x_i es transformado a un vector de dimensión d mediante una proyección lineal:
   $$z_i = E \cdot x_i + p_i \tag{4.1}$$
   donde E es una matriz de proyección aprendible y p_i representa el vector de codificación posicional correspondiente al parche i.

4. **Auto-atención y cabezas múltiples (Multi-Head Self-Attention)**
   La secuencia de vectores proyectados se alimenta a un codificador Transformer, que aplica mecanismos de auto-atención para capturar relaciones entre todos los parches.

   Dados los vectores de entrada $\mathbf{X} \in \mathbb{R}^{N \times d}$, donde N es el número de parches y d la dimensión del vector, se proyectan en tres espacios

distintos mediante matrices que se actualizarán en cada iteración:

$$\mathbf{Q} = \mathbf{W}_Q\mathbf{X}, \quad \mathbf{K} = \mathbf{W}_K\mathbf{X}, \quad \mathbf{V} = \mathbf{W}_V\mathbf{X} \tag{4.2}$$

Donde $\mathbf{W}_Q$, $\mathbf{W}_K$ y $\mathbf{W}_V$ son matrices de pesos entrenables y X es la representación de entrada de los datos.

La puntuación de atención se calcula mediante el producto escalar entre las consultas y las claves, escalado por la raíz cuadrada de la dimensión de las claves d_k:

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right) \tag{4.3}$$

Finalmente, la salida del mecanismo de atención se obtiene como:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{A} \cdot \mathbf{V} \tag{4.4}$$

Este proceso se ejecuta en paralelo en múltiples *cabezas de atención*, permitiendo al modelo enfocarse simultáneamente en distintos aspectos de la imagen.

5. **Clasificación**
   Un token adicional llamado `[CLS]` se introduce al inicio de la secuencia. Su representación final, luego de atravesar todas las capas del codificador Transformer, se utiliza como entrada para una capa final de clasificación.

4.3 *Aplicación de ViT en la detección de DeepFakes*

Uno de los desafíos en la detección de imágenes generadas artificialmente (deep fakes) es la presencia de anomalías sutiles en la textura, iluminación y detalles faciales. Las CNN han demostrado cierto éxito en la detección de estas anomalías, pero pueden ser limitadas debido a su enfoque en patrones locales.

En contraste, Vision Transformers (ViT) pueden detectar estas inconsistencias con mayor precisión, ya que su mecanismo de autoatención permite evaluar toda la imagen simultáneamente, identificando relaciones atípicas entre los parches. Estos son algunos ejemplos de lo que podría rescatar el modelo:

1. Detectar artefactos en la piel, ojos y cabello que podrían pasar desapercibidos en métodos tradicionales.

2. Analizar irregularidades en la iluminación global de la imagen, ya que los generadores de Deep-fakes suelen tener dificultades para

replicar iluminación coherente en todas las partes del rostro.

3. Capturar dependencias a largo alcance entre regiones de la imagen, lo que permite identificar incoherencias en la estructura facial.

5 Estado del Arte

Contenido

5.1 Introducción general al estado del arte 35
5.2 Detección de DeepFakes: Actualidad 37
5.2.1 Redes Neuronales Convolucionales (CNN) 37
5.2.2 Redes Neuronales Adversariales (GAN) 40
5.2.3 Diferencias principales entre ViT y CNN 41
5.3 Conclusión del capítulo 43

5.1 Introducción general al estado del arte

La presente sección tiene como propósito contextualizar los avances teóricos y prácticos en torno al desarrollo de modelos aplicados a la detección de contenido audiovisual manipulado mediante inteligencia artificial. Este análisis busca proporcionar una base sólida para comprender las técnicas actuales y establecer un punto de comparación con la propuesta metodológica desarrollada en la fase de implementación de este trabajo.

Uno de los fenómenos más relevantes en este ámbito es la generación de contenido sintético conocido como *DeepFake*. El término surge de la combinación de las palabras *deep learning* y *fake*, y hace referencia a videos, imágenes o audios alterados mediante algoritmos avanzados de inteligencia artificial. Este tipo de contenido ha experimentado un crecimiento acelerado en los últimos años, tanto en disponibilidad como en sofisticación, lo que ha incrementado el interés académico y social en su detección y regulación [1].

[1] B. Cruz, "2024 deepfakes guide and statistics," Sept. 2024

Según Cruz (2024) [2], las aplicaciones maliciosas más comunes de los *DeepFakes* incluyen:

[2] B. Cruz, "2024 deepfakes guide and statistics," Sept. 2024

1. Campañas de desinformación.
2. Interferencia en procesos electorales.

3. Ciberacoso o *bullying*.
4. Producción y difusión de contenido pornográfico no consentido.
5. Generación de noticias falsas.
6. Clonación o suplantación de voz.

Este panorama pone en evidencia la necesidad de cuantificar el impacto y el riesgo asociados al uso indebido de tecnologías generativas.

De acuerdo con una encuesta realizada por la empresa McAfee en diciembre de 2024 a 5,000 personas a nivel global, se identificaron los siguientes hallazgos: [3]:

[3] J. Dhaliwal, "State of the scamiverse – how ai is revolutionizing online fraud," Jan. 2025

- Una persona adulta está expuesta, en promedio, a entre 10 y 14 intentos de fraude digital diariamente, incluyendo 2.6 videos *DeepFake*.
- Un tercio de las víctimas reporta pérdidas superiores a 500 USD, y un 10% supera los 5,000 USD en pérdidas económicas.
- Las personas dedican entre 83 y 94 horas anuales revisando contenido sospechoso para evitar fraudes.
- El 59% de los encuestados conoce a alguien que ha sido víctima de una estafa en línea, cifra que aumenta al 77% en el grupo etario de 18 a 24 años.
- El 64% de las estafas digitales resultan en pérdidas financieras o robo de datos personales en menos de una hora.
- Además del impacto económico, el 35% de las víctimas reporta altos niveles de estrés emocional asociado al incidente.

Adicionalmente, se observa una segmentación demográfica en la exposición a *DeepFakes* según la plataforma digital. Mientras que los adultos mayores (65+) tienden a encontrarlos en mayor medida en Facebook (más del 80% así lo reporta), los usuarios jóvenes están más expuestos en Instagram y TikTok. En promedio, los jóvenes entre 18 y 24 años visualizan 3.5 *DeepFakes* diarios, frente a 1.2 en el grupo de mayores de 65 años.

La gravedad del impacto de esta tecnología, desde una perspectiva social, refuerza la necesidad de avanzar en el desarrollo de herramientas capaces de detectar este tipo de contenido de forma eficiente y escalable. En este contexto, los modelos basados en *Transformers*, particularmente su adaptación a tareas visuales mediante *Vision Transformers*, representan una línea de investigación prometedora que será abordada en las siguientes secciones.

5.2 Detección de DeepFakes: Actualidad

Llevar a cabo una investigación acerca de las diversas herramientas y modelos empleados en la identificación de imágenes manipuladas 'DeepFake' resulta de suma importancia para poder efectuar un análisis comparativo y poner en contraste las potenciales diferencias o semejanzas entre el método utilizado en este estudio y aquellos propuestos en la literatura especializada vigente.

Basado en múltiples fuentes de información y 'papers', en la actualidad se emplean tanto modelos de machine learning tradicionales como modelos avanzados de deep learning para llevar a cabo tareas de detección; no obstante, la evidencia científica recopilada sugiere que los modelos de deep learning, en efecto, suelen arrojar resultados superiores y presentan una capacidad de generalización notablemente más sólida en contraste con los modelos convencionales. [4]

[4] M. S. Rana, M. N. Nobi, B. Murali, and A. H. Sung, "Deepfake Detection: A Systematic Literature Review," 2022

La tendencia creciente en la generación de DeepFakes, que suelen generarse mediante redes neuronales profundas, según menciona el estudio realizado por Husrev en el año 2020,[5] lo que ha impulsado aún más las investigaciones en torno a este tipo de modelos y sus aplicaciones. La creciente disponibilidad de bases de datos faciales, junto con los continuos avances en sofisticados modelos de inteligencia artificial, ha propiciado el surgimiento de esta tendencia en los DeepFakes.

[5] H. Taha, S. Luisa, V. Nasir, and M. Editors, "Advances in Computer Vision and Pattern Recognition," tech. rep

Según el estudio publicado en febrero 2022 por el autor MD SHOHEL RANA, [6] en conjunto con sus colegas investigadores, afirman que cerca del 77% de los estudios sobre detección de imágenes DeepFake emplean métodos basados en Deep Learning. El 23% restante son normalmente basados en modelos de machine learning y, aunque sí pueden alcanzar buenos niveles de precisión en conjuntos de datos específicos, cuando se les muestra otro conjunto de datos distinto, la precisión decae significativamente.

[6] M. S. Rana, M. N. Nobi, B. Murali, and A. H. Sung, "Deepfake Detection: A Systematic Literature Review," 2022

Cabe mencionar que hay una gran variedad de modelos existentes, como se muestra en la siguiente figura [7], donde se puede apreciar que el listado es extenso; sin embargo, se mencionarán los que son los más utilizados, también afirmados por el mismo estudio.

[7] M. S. Rana, M. N. Nobi, B. Murali, and A. H. Sung, "Deepfake Detection: A Systematic Literature Review," 2022

5.2.1 Redes Neuronales Convolucionales (CNN)

Las Redes Neuronales Convolucionales son modelos especialmente diseñados para procesar datos con estructura espacial, es decir, donde

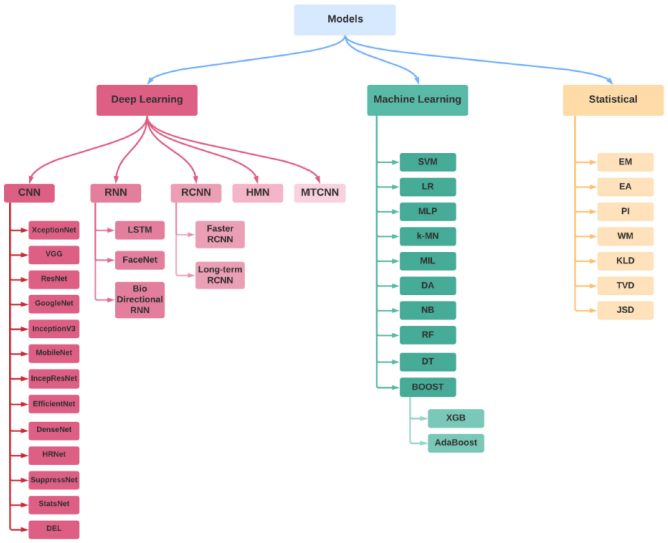

Figura 5.1: Diagrama de todos los modelos para detección de DeepFakes.

Category	Model	#Studies	PCT (%)
Deep Learning	CNN	71	78%
	RNN	12	13%
	RCNN	2	2%
Machine Learning	SVM	11	12%
	k-MN	4	4%
	LR	3	3%
	MLP	3	3%
	BOOST	2	2%
	RF	1	1%
	DT	1	1%
	DA	1	1%
	NB	1	1%
	MIL	1	1%
Statistical	EM	1	1%
	TV, KL, JS	1	1%

Figura 5.2: Distribución de los modelos y los números de estudios utilizados.

el orden de los datos sí es relevante.

Básicamente, su arquitectura consiste en una capa de convolución donde se aplica una especie de filtro pequeño con la finalidad de que se detecten características locales de cada imagen, como borders, texturas y esquinas. Matemáticamente es una suma ponderada del filtro con la imagen.

Estas redes son particularmente la técnica más utilizada en comparación con el resto de modelos basados en deep learning. Su popularidad se basa en varios motivos, pero principalmente se podría mencionar:

1. Capacidad de extracción automática de características. Gracias a la convolución, pueden extraer información como bordes, texturas e incluso rasgos faciales. Esto es sumamente útil para la detección de distorsiones sutiles o inconsistencias en la iluminación en ciertas partes del rostro, lo que ayudaría en la detección.
2. Consumo de recursos computacionales económicos.
3. Consistencia en la precisión.

Arquitecturas como DenseNet169, DenseNet201 o ResNet50 tienden a tener un porcentaje muy alto y consistente en precisión. Modelos especializados como MesoNet o Xception diseñados para vídeo, han alcanzado precisiones de detección cercanas al 99% en el dataset FaceForensics++, confirmando la eficacia de las CNN para aprender directamente las huellas espectrales y espaciales que deja el proceso de generación de DeepFakes.

No obstante, las investigaciones han revelado una serie de desafíos aún sin resolver en la implementación de escenarios reales. Por ejemplo, las redes entrenadas frecuentemente experimentan dificultades frente a DeepFakes generados con modelos distintos a los empleados durante la fase de entrenamiento; la interpretación y modificación de dimensiones en las imágenes en diversas plataformas digitales según el dispositivo, lo que podría ocasionar conflictos en las dimensiones y además perder características sutiles debido a la comprensión; también la aparición de nuevas técnicas de manipulación como la simulación de videos reales como falsos ("fake DeepFakes"), en las que estos ataques adversos generan falsos positivos, lo que incrementa la susceptibilidad de los modelos de aprendizaje profundo a ataques adversarios. Además, los DeepFakes de alta resolución presentan cada vez menos artefactos perceptibles, lo que complica aún más su detección.

A pesar de que el aprendizaje profundo, liderado por las CNN, continúa siendo el método más eficiente para la detección de DeepFakes, resulta esencial proseguir con la investigación para superar estos desafíos y desarrollar métodos más sólidos, generalizables e interpretables en contextos del mundo real.

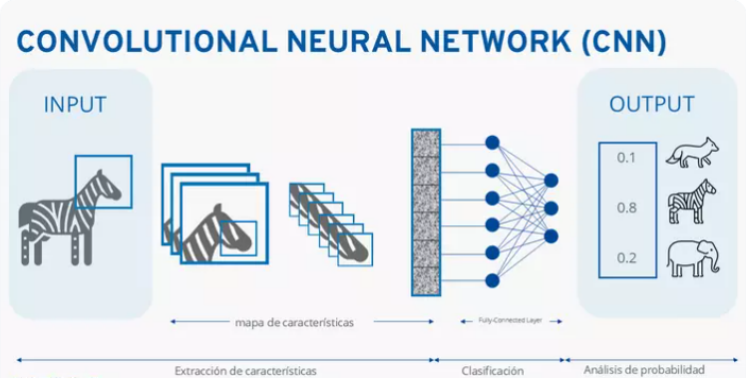

Figura 5.3: Diagrama de CNN para clasificación multiclase.

5.2.2 *Redes Neuronales Adversariales (GAN)*

Según los artículos consultados, un modelo frecuentemente utilizado son las redes neuronales adversariales (GAN), una arquitectura de modelos de aprendizaje profundo introducida por Ian Goodfellow en 2014. Su esencia radica en poner en competencia dos redes neuronales de tal manera que ambas mejoren hasta llegar a ser prácticamente indistinguibles una de la otra.

En términos académicos, se compone de dos elementos fundamentales: un generador, cuyo propósito es generar una imagen nueva que intente replicar la distribución de los datos reales, y un discriminador, que recibe ejemplos de imágenes tanto reales como las generadas por el generador, con el propósito de clasificar de manera adecuada cada muestra.

Esta dinámica resulta sumamente intrigante, dado que, al capacitar al discriminador, se le presentan tanto ejemplos reales como sintéticos, optimizando su función de pérdida para potenciar su habilidad para diferenciar ambos. En contraposición, al capacitar al generador, este busca inducir al discriminador. Finalmente, estos pasos se alternan y, con el transcurso de los épocas, el generador adquiere la habilidad de generar muestras más fidedignas, mientras que el discriminador se vuelve más exigente.

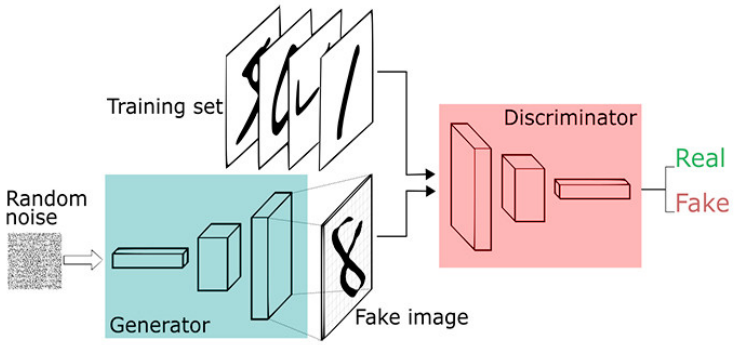

Figura 5.4: Diagrama de GAN para clasificación multiclase.

Las principales aplicaciones de estos modelos son la generación de imágenes hiperrealistas, la mejora sustancial en la resolución de una imagen y también la generación de más muestras en un conjunto de datos pequeño.

5.2.3 *Diferencias principales entre ViT y CNN*

Para resumir la información, se preparó una pequeña tabla comparativa para la distinción de cada modelo:

Característica	**CNN**	**Vision Transformer (ViT)**
Unidad básica de entrada	Píxeles locales y filtros convolucionales	Parches de imagen tratados como tokens
Mecanismo de extracción de características	Convoluciones locales jerárquicas	Auto-atención global sobre toda la imagen
Estructura espacial	Implícita por diseño (estructuras locales)	Requiere codificación posicional explícita
Dependencias de largo alcance	Limitadas por el tamaño del kernel	Capturadas directamente por atención global
Datos necesarios para buen desempeño	Eficiente con datasets medianos	Requiere grandes volúmenes de datos para entrenar eficazmente
Paralelización del cómputo	Limitada por operaciones secuenciales	Altamente paralelizable
Robustez ante pequeñas transformaciones (traslaciones, ruido)	Alta, por diseño local e invarianza espacial	Menor, aunque mejorable con preentrenamiento o data augmentation

Tabla 5.1: Comparación entre CNN y Vision Transformer (ViT)

5.3 Conclusión del capítulo

Con la investigación realizada, se puede deducir que ViT no es tan empleado en comparación con otros modelos como las Redes Neuronales Convolucionales o las Redes Adversariales, principalmente debido al gran volumen de datos requeridos y el esfuerzo computacional que implica para la tarea de clasificación de imágenes; no obstante, existen pruebas de un rendimiento competitivo siempre que haya una gran cantidad de datos o si se utilizan modelos preentrenados.

Cabe destacar, que su capacidad para identificar relaciones globales proporciona un enorme beneficio a este modelo en situaciones donde los patrones relevantes no se encuentran necesariamente agrupados.

En contraste, los grandes retos de estos modelos son su implementación, debido a que son computacionalmente más exigentes al manejar matrices enormes de dimensión $N \times N$, donde N es el número de parches, lo que lógicamente implica más memoria y tiempo de cómputo.

También, se menciona en las fuentes consultadas que este modelo podría ser más vulnerable a imágenes mal recortadas, mal alineadas o con poco contraste.

6 Desarrollo y Resultados

Contenido

6.1 Métricas utilizadas **46**
6.2 Análisis y descripción del conjunto de imágenes **47**
6.3 Proceso de implementación **49**
6.3.1 Generación de muestras Data Augmentation 51
6.3.2 Preprocesamiento de imágenes con modelo de detección de rostros 51
6.3.3 Primer modelo de detección de rostros: Haarcascade 52
6.3.4 Segundo modelo de detección de rostros: YOLO 53
6.3.5 Ventajas y desventajas del uso de YOLOv8 55
6.3.6 Comparación entre ambos modelos de detección de rostros. Haarcascade y YOLOv8 57
6.3.7 Carga de rostros detectados y normalización de imágenes 58
6.3.8 División de imágenes en parches . . . 59
6.3.9 Proyección lineal de cada parche . . . 59
6.3.10 Codificación posicional de los vectores 60
6.3.11 Bloque Transformer: Mecanismo de autoatención 60
6.3.12 Feed Forward Networks (FNN) y conexiones residuales 62
6.3.13 Resumen de un bloque Transformer . 63
6.4 Arquitectura e hiper parámetros **64**
6.5 Resultados **64**

En este capítulo se describe el desarrollo y construcción del modelo, así como su implementación. Adicionalmente, se hace un análisis descriptivo del conjunto de datos utilizado; se propone una manera de preprocesamiento del conjunto de datos con la ayuda de un modelo preentrenado para la detección de rostros y, finalmente, se describe

la implementación general del modelo para la detección de imágenes 'DeepFake', así como el detalle del entrenamiento y la evaluación, junto con la experimentación con diferentes hiperparámetros.

6.1 *Métricas utilizadas*

Para analizar adecuadamente el rendimiento de los modelos, se utilizarán las siguientes métricas estándar en clasificación binaria: Accuracy, Recall, Precision, False Positive Rate (FPR) y F1 Score.

Cada una de estas métricas ofrece una perspectiva distinta sobre el desempeño del modelo, dependiendo del tipo de error que se desea minimizar o del fenómeno que se busca interpretar. En la literatura, es habitual analizar el comportamiento del clasificador en función de los aciertos y errores al detectar clases positivas o negativas.

En este proyecto, se considera como clase positiva (1) a las imágenes generadas mediante DeepFake, y como clase negativa (0) a las imágenes reales. Bajo esta convención, se definen las métricas de la siguiente forma:

1. **TP (True Positives):** Imágenes DeepFake clasificadas correctamente como falsas.
2. **TN (True Negatives):** Imágenes reales clasificadas correctamente como reales.
3. **FP (False Positives):** Imágenes DeepFake clasificadas incorrectamente como reales.
4. **FN (False Negatives):** Imágenes reales clasificadas incorrectamente como falsas.

Dicho lo anterior, se presentan las definiciones matemáticas de las métricas mencionadas, conforme a lo descrito en la documentación técnica de Google para clasificación binaria [1]:

[1] Google, "Classification: Accuracy, recall, precision, and related metrics," 3 2025

1. **Exactitud (Accuracy)**
   Proporción de clasificaciones correctas (tanto positivas como negativas) sobre el total de ejemplos.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

2. **Recall (True Positive Rate, TPR)**
   Proporción de ejemplos positivos reales (DeepFakes) que fueron

correctamente clasificados como positivos.

$$Recall = \frac{TP}{TP + FN} \tag{6.2}$$

3. **False Positive Rate (FPR)**
   Proporción de ejemplos negativos reales (imágenes reales) que fueron incorrectamente clasificados como positivos (DeepFakes).

$$FPR = \frac{FP}{FP + TN} \tag{6.3}$$

4. **Precisión (Precision)**
   Proporción de ejemplos clasificados como positivos (DeepFakes) que efectivamente eran positivos.

$$Precision = \frac{TP}{TP + FP} \tag{6.4}$$

5. **F1 Score**
   Media armónica entre precisión y recall, que busca balancear ambos indicadores en un solo valor.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \tag{6.5}$$

6.2 *Análisis y descripción del conjunto de imágenes*

El conjunto de datos utilizado en este proyecto está compuesto por aproximadamente 147 000 imágenes de uso público descargadas desde la plataforma Kaggle (https://www.kaggle.com/datasets/manjilkarki/deepfake-and-real-images), que a su vez hace referencia a la fuente original del dataset (https://zenodo.org/records/5528418#.YpdlS2hBzDd). Las imágenes representan a personas de diferentes edades, etnias, géneros y poseen distintas expresiones faciales, gestos, poses y la presencia o ausencia de accesorios como gafas, maquillaje, sombreros, etc.

La idea detrás de la inclusión de esta diversidad en las muestras es la de enriquecer al modelo en su aprendizaje de características humanas y reducir posibles sesgos en su clasificación entre rostros reales y rostros con modificaciones o generaciones artificiales.

Continuando con la descripción del conjunto de imágenes, podemos observar que contiene más imágenes reales que falsas en una relación de 1.8:1 aproximadamente. Esto es significativo y no ideal, debido a que se requiere que el modelo tenga la misma oportunidad de aprender de ambas clases sin brindar preferencia por alguna porque contiene

Imagen real 1

Imagen real 2

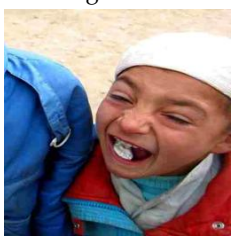

Imagen real 3

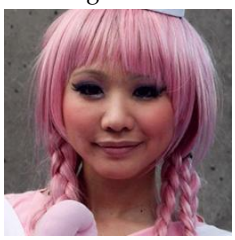

Imagen real 4

Figura 6.1: Ejemplos de imágenes de personas reales dentro del conjunto de datos.

más muestras. En este caso, como hay más imágenes reales que falsas, sí podría inducir a que el modelo favorezca esa clase.

```
Distribución de clases - Train: {'Fake': 27087, 'Real': 70001}
Distribución de clases - Validation: {'Fake': 19641, 'Real': 19787}
Distribución de clases - Test: {'Fake': 5492, 'Real': 5413}
```

Figura 6.2: Balanceo de clases del conjunto de datos.

La razón por la cual sucede esto es porque el modelo intentará buscar minimizar la función de pérdida global; entonces, por ejemplo, si en su labor de clasificación, acierta con más frecuencia la clasificación "real", el modelo podría aprender que puede clasificar real la mayoría de las imágenes y aún así obtener buena exactitud sacrificando la predicción de las falsas.

Ahora bien, para mitigar esto, existen varias técnicas utilizadas comúnmente para intentar balancear las clases; algunas de ellas podrían ser asignar mayor peso a la clase minoritaria, repetir o descartar ejemplos de las clases o técnicas como el "data augmentation", que fue la elegida para este proyecto y que consiste en aplicar transformaciones aleatorias (rotaciones, flips, recortes) sobre la clase minoritaria para enriquecerla. Se hablará más de la implementación de esta técnica más adelante.

Continuando con la parte del análisis exploratorio del conjunto de datos, se generó un histograma de intensidad de píxeles con el objetivo de identificar posibles diferencias significativas en la distribución de valores de color entre las imágenes reales y las imágenes DeepFake.

Esta comparación resulta relevante, dado que, como se mencionó ini-

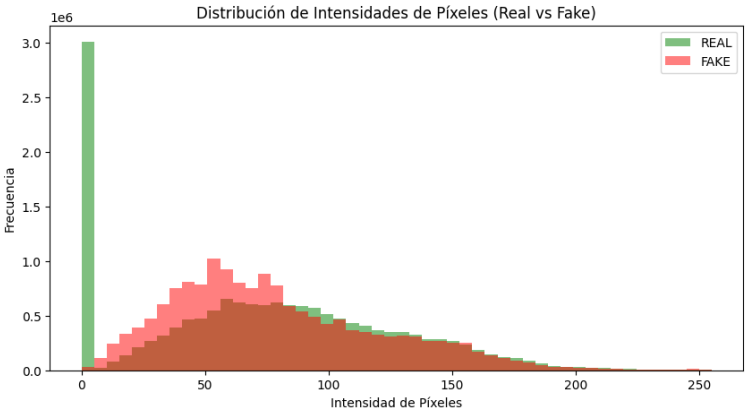

Figura 6.3: Histograma de intensidad de color por píxel

cialmente, los modelos Vision Transformer (ViT) pueden ser sensibles a variaciones en la iluminación o contrastes atípicos. Por ello, un análisis previo de la intensidad de los píxeles permite evaluar si existen sesgos en el preprocesamiento que podrían afectar el rendimiento del modelo.

Podemos observar que, aunque hay solapamiento en un rango amplio de intensidades, se aprecia que la distribución de píxeles en "Fake" está más extendida en intensidades medias (entre 30 y 80, aproximadamente). Aunque no sea una diferencia grande, podría sugerir que las falsas presentan una distribución más uniforme en valores bajos y medios lo que podría deberse a diferencias en iluminación, fondo, o a la forma en que fueron obtenidas o procesadas.

Cabe mencionar que basarse solo en la intensidad de píxeles podría no ser suficiente para diferenciar notoriamente cada clase en el sentido estricto, pero sí sugiere que hay un patrón distinto en la forma de la distribución, lo que podría darnos algunas pistas sobre algún procesamiento dentro del conjunto de imágenes.

6.3 *Proceso de implementación*

Para la etapa de la implementación, se comenzó con la carga y preprocesamiento de las imágenes. Estas ya están divididas en tres carpetas distintas: Train, Test, Validation; y dentro de cada una hay un subconjunto de carpetas que contienen Reales y Fake.

Se realizó el siguiente diagrama de flujo para visualizar los pasos realizados.

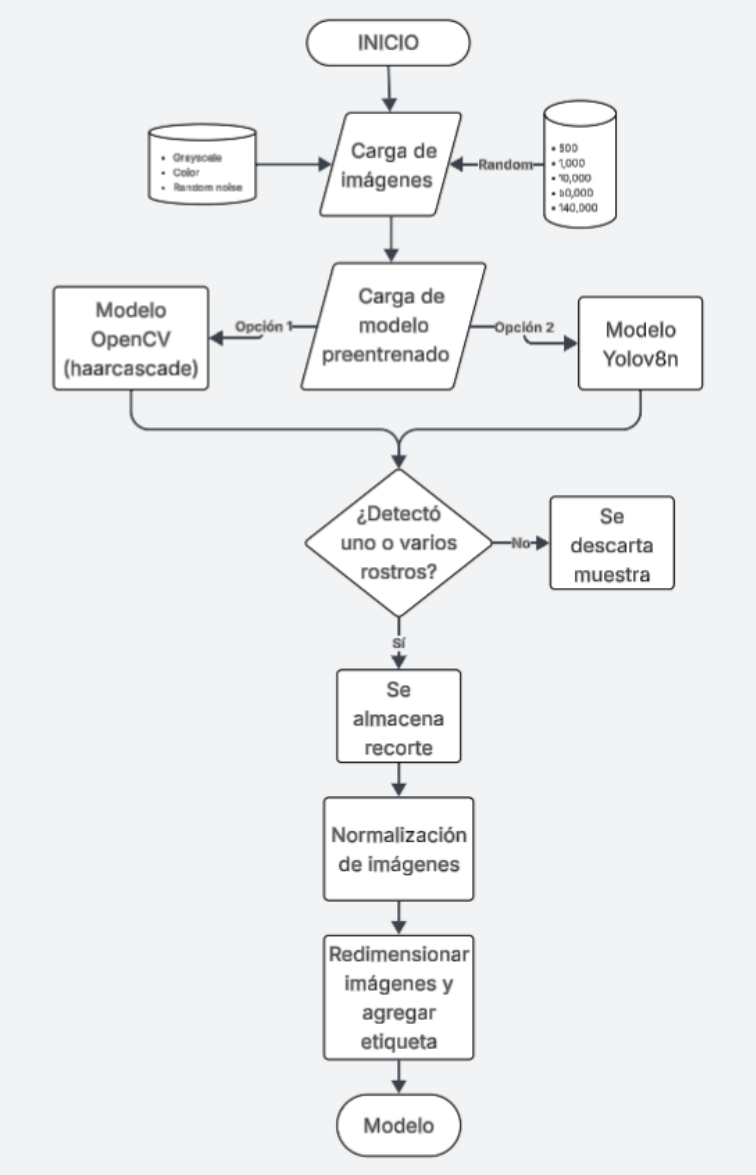

Figura 6.4: Diagrama de flujo para carga y preprocesamiento de imágenes

6.3.1 *Generación de muestras Data Augmentation*

Como se mencionó en la sección anterior, "Data augmentation" es una técnica que genera variaciones de las imágenes del dataset, como rotaciones hacia los lados o giros verticales, para ampliar el número de ejemplos. Esto permite que el modelo aprenda a reconocer rostros en distintas posiciones, lo cual es útil considerando que en contextos reales (o en DeepFakes) las caras no siempre están centradas o rectas.

Esta técnica también ayuda a equilibrar el dataset, especialmente cuando una clase tiene menos muestras. En este caso, se aplicó para aumentar la clase fake, ya que los modelos ViT requieren grandes volúmenes de datos para entrenar eficazmente, y se buscó aprovechar al máximo las imágenes disponibles.

A continuación se muestran algunos ejemplos de cómo se ve implementada esta técnica:

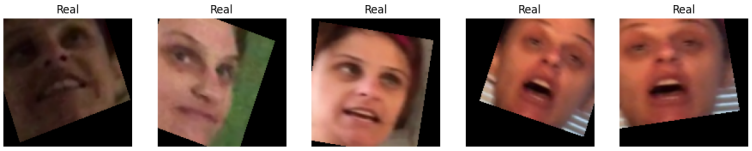

Figura 6.5: Ejemplo de técnica de data augmentation

Con esto, el dataset logró equilibrarse para ambas clases.

6.3.2 *Preprocesamiento de imágenes con modelo de detección de rostros*

Una vez se terminó con la carga y balanceo de clases, se hizo la carga de las imágenes con la ayuda de la librería de openCV en python. Cabe mencionar que las imágenes del conjunto de datos son de 256 x 256 píxeles de tamaño, y, aunque no son imágenes de alta calidad y algunos detalles podrían perderse debido a la compresión, se considera que podría ayudar a que el modelo pueda procesar más rápido las imágenes.

Es importante destacar que las imágenes lógicamente presentan diferentes escenarios y fondos detrás de las personas, que podrían introducir ruido o sesgos indeseables en el aprendizaje del modelo; por ello, se optó por emplear un modelo preentrenado de detección de rostros que busca extraer únicamente la región facial, descartando la mayor parte del fondo de cada persona. De esta manera, se intenta que el modelo pueda mejorar su capacidad de aprendizaje.

6.3.3 *Primer modelo de detección de rostros: Haarcascade*

El primer modelo preentrenado que se utilizó para la detección de los rostros se llama 'haarcascade'; es un modelo sencillo de implementar y está disponible mediante la librería de OpenCV.

El objetivo también es optimizar el rendimiento y ahorrar poder computacional, enfocando el poder del modelo en las regiones de interés y por eso el recorte en la región del rostro. Esta técnica está basada en el algoritmo propuesto por Viola y Jones (2001) en su artículo "Rapid Object Detection using a Boosted Cascade of Simple Features". [2].

[2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 511–518, IEEE, 2001

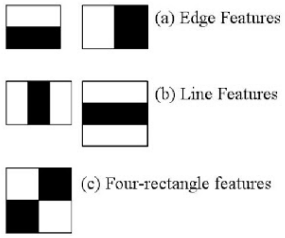

Figura 6.6: Ejemplo de detección de rostro con modelo Haar cascade de la librería OpenCV.

El proceso detrás de este algoritmo, según se menciona en la documentación, se inicia con la capacitación de un clasificador a partir de un volumen considerable de imágenes positivas (con representaciones faciales) y negativas (sin representaciones faciales).

Se derivan características, que se manifiestan en patrones simples de intensidad, análogos a los filtros convolucionales. Cada propiedad se determina al sustraer la suma de píxeles en una región blanca de la suma de píxeles en una región negra. No obstante, dado que se producen miles de características potenciales (más de 160,000 en ventanas de 24x24 píxeles), resulta imperativo elegir exclusivamente aquellas que demuestren mayor discriminación. El procedimiento de selección se realiza a través del algoritmo *Adaboost*, que distingue los clasificadores débiles (características individuales) que mejor distinguen las imágenes de rostros y no rostros. Subsecuentemente, integra diversos clasificadores débiles en un clasificador robusto, ponderando sus aportaciones en función de su exactitud.

El modelo de OpenCV, desgraciadamente, presentaba ciertas limitantes; por ejemplo, todos los rostros donde la persona estuviese de perfil o con muchos accesorios presentaban una detección inconsistente.

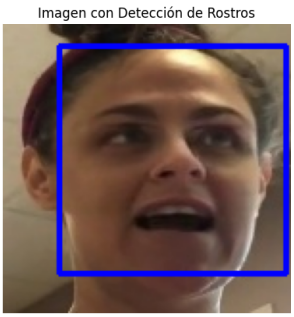

Figura 6.7: Ejemplo de detección de rostro con modelo Haar cascade de la librería OpenCV.

En ocasiones, el modelo no realizaba la detección y se descartaba la muestra. Definitivamente, no eran muchos casos particularmente con estas características, sin embargo, sí es importante mencionarlo porque, los descartes, poco a poco provocaban un desbalanceo en el conjunto de datos.

Figura 6.8: Ejemplo rostro de perfil.

También, por la forma en que funciona el modelo, ocasionalmente no hacía detecciones debido a la iluminación de la imagen o, en ocasiones, hacía falsas detecciones donde no había rostro, también debido a la luz.

Figura 6.9: Ejemplo rostro a contraluz.

6.3.4 *Segundo modelo de detección de rostros: YOLO*

Para intentar mejorar la detección y evitar los descartes del primer modelo, se buscó implementar otro modelo. YOLO, modelo mucho más robusto y moderno para detección de objetos en general, es muy

popular principalmente por su balance entre precisión y velocidad, sobre todo en extracción de características y detección de objetos en tiempo real.

Yolo funciona con una red convolucional entrenada para localizar rostros mediante cajas alrededor de las regiones faciales de la imagen. Yolo aprende representaciones complejas de rostros a partir de los datos.

Teóricamente, YOLO alcanza niveles superiores en comparación con el modelo anterior porque es particularmente mejor en situaciones de iluminación variable, presencia de varios rostros o posturas no frontales de las personas.

Sin embargo, el modelo demanda una cantidad muy superior de datos para su entrenamiento y su implementación es más compleja. Sin un entrenamiento apropiado para el tipo de objeto que se requiere detectar, el modelo tiende a sobreajustarse o a fallar.

En el caso de este trabajo, los principales problemas que se tuvieron fueron la detección de falsos rostros en múltiples objetos.

El siguiente ejemplo muestra que YOLO tuvo la capacidad de detectar un rostro en la parte de atrás de este DeepFake, que es algo correcto, y como hubo detección, la imagen se recorta y se añade como muestra; sin embargo, se cree que añadir ejemplos incompletos de rostros, podría interferir con el aprendizaje de las características.

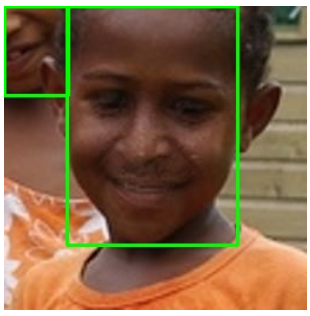

Figura 6.10: Ejemplo 1 de detección con modelo YOLO

Aquí se muestra otro ejemplo de falsas detecciones de "rostros" en objetos. En este caso, se enviaron 3 muestras al modelo: el señor, el traje y la corbata.

Se cree que este tipo de falsas detecciones pudieron haberse provocado o porque el modelo de Yolo se configuró con una sensibilidad muy alta o porque el conjunto de datos utilizado, al ser imágenes de baja resolución, interfiere en la correcta detección.

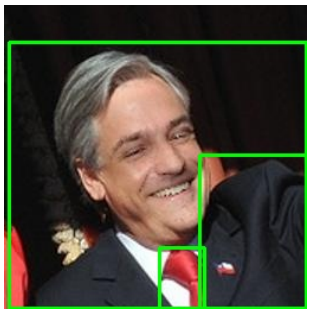

Figura 6.11: Ejemplo 2 de detección con modelo YOLO

Figura 6.12: Ejemplo 3 de detección con modelo YOLO

6.3.5 *Ventajas y desventajas del uso de YOLOv8*

- **Alta precisión:** YOLOv8 logra resultados superiores en precisión en comparación con métodos tradicionales, especialmente en condiciones de iluminación variable, múltiples rostros en la imagen o poses no frontales.
- **Velocidad de inferencia:** Gracias a su arquitectura optimizada, permite realizar detecciones en tiempo real, incluso en dispositivos con recursos computacionales limitados.
- **Robustez ante variaciones:** Es más robusto frente a cambios de escala, rotaciones parciales y oclusiones parciales del rostro.
- **Detección de múltiples rostros simultáneamente:** Puede identificar varios rostros dentro de una misma imagen de manera eficiente y precisa.
- **Requiere mayor poder de cómputo:** Comparado con métodos clásicos como Haarcascade, el entrenamiento y despliegue de YOLOv8 demanda más capacidad de procesamiento, especialmente en dispositivos de baja potencia.
- **Dependencia de gran cantidad de datos etiquetados:** El modelo requiere grandes volúmenes de imágenes anotadas para alcanzar su máximo rendimiento.

- **Mayor complejidad de implementación:** Integrar y ajustar YOLOv8 en un sistema puede ser más complejo que utilizar clasificadores preentrenados más ligeros.
- **Sobreajuste en datasets pequeños:** Si no se realiza un entrenamiento adecuado o no se utilizan técnicas de regularización, el modelo puede sobreajustarse fácilmente a los datos de entrenamiento.

6.3.6 *Comparación entre ambos modelos de detección de rostros. Haarcascade y YOLOv8*

Característica	Haarcascade	YOLOv8
Tipo de modelo	Basado en clasificadores simples (boosted cascade)	Red neuronal convolucional profunda
Velocidad de inferencia	Muy rápida en CPU de bajo rendimiento	Rápida, pero optimizada para GPU y hardware potente
Precisión en condiciones reales	Limitada, baja tolerancia a variaciones de pose, iluminación y escala	Alta precisión, robusto ante variaciones de iluminación, escala, oclusiones y poses
Capacidad de detección múltiple	Detección limitada de múltiples rostros, menos precisa	Detección eficiente y precisa de múltiples rostros simultáneamente
Requerimiento de datos para entrenamiento	Moderado (entrenamientos clásicos)	Alto (grandes volúmenes de datos etiquetados)
Facilidad de implementación	Muy sencilla, ideal para prototipos rápidos o sistemas embebidos	Requiere integración de frameworks modernos (Ultralytics, PyTorch, etc.)
Robustez ante ruido o fondos complejos	Sensible a ruido y fondos no controlados	Alta tolerancia y adaptabilidad a entornos complejos
Consumo de recursos	Muy bajo (ideal para CPUs)	Moderado a alto (mejor en GPUs modernas)

Tabla 6.1: Comparativa entre Haarcascade y YOLOv8 para detección de rostros

6.3.7 *Carga de rostros detectados y normalización de imágenes*

Una vez hecha la carga de las imágenes recortadas únicamente con las detecciones de los rostros, estas se convierten internamente en arreglos numéricos multidimensionales utilizando la estructura de NumPy (librería de Python que ayuda a trabajar con matrices numéricas). Cada imagen es tratada como una matriz tridimensional, donde la estructura general es: Alto, Ancho, Canal

Dentro del marco de las imágenes a color, cada píxel se representa mediante un vector de tres valores enteros que corresponden a los canales de color. OpenCV adopta por defecto el formato BGR (Blue, Green, Red). En otras palabras, cada píxel se representa como un vector de la siguiente categoría: [pixel] = [B, G, R]

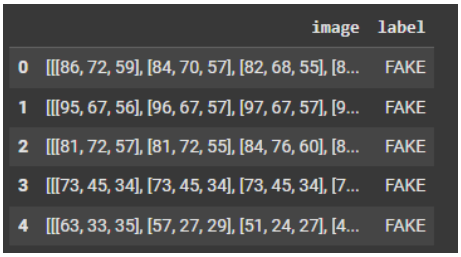

	image	label
0	[[[86, 72, 59], [84, 70, 57], [82, 68, 55], [8...	FAKE
1	[[[95, 67, 56], [96, 67, 57], [97, 67, 57], [9...	FAKE
2	[[[81, 72, 57], [81, 72, 55], [84, 76, 60], [8...	FAKE
3	[[[73, 45, 34], [73, 45, 34], [73, 45, 34], [7...	FAKE
4	[[[63, 33, 35], [57, 27, 29], [51, 24, 27], [4...	FAKE

Figura 6.13: Dataset y su etiqueta.

En este caso, la figura muestra un fragmento de la estructura de estas matrices una vez cargadas. Se observa que cada elemento en la columna image es una lista de listas anidadas que representan filas de píxeles. A su vez, cada píxel está representado por un vector numérico que contiene los valores de intensidad de los canales BGR. La columna label indica la clase asignada a la imagen; en este caso, todas corresponden a la categoría “FAKE”.

Esta modalidad de representación es esencial para sustentar modelos de aprendizaje automático, dado que el modelo interpreta los valores numéricos como rasgos o características. Sin embargo, es imperativo que estos valores sean adecuadamente normalizados o redimensionados antes de ser incorporados en el modelo ViT.

Para ello, se necesita escalar los valores de intensidad de los píxeles a un intervalo homogéneo, normalmente de [0 , 1] ya que la intensidad de los píxeles va de 0 a 255. Esta transformación no cambia la información esencial, pero sí mejora la forma en que el modelo la procesa. Ya que, si los valores son muy grandes, las operaciones internas como algunas multiplicaciones matriciales podrían convertirse en números más grandes, provocando desbordamientos y haciendo que los gradientes

disminuyan demasiado o haciendo que los valores pequeños pierdan importancia.

6.3.8 *División de imágenes en parches*

Una vez que las imágenes de los rostros han sido normalizadas, se procedió a dividirlas en pequeñas regiones cuadradas conocidas como *patches*. El tamaño de cada 'patch' corresponde a uno de los hiperparámetros de este modelo y se puede modificar.

Para este trabajo, se eligió que se utilizaran patches de tamaño 16×16 píxeles, lo que implica que cada imagen de entrada de256×256 se transforma en una secuencia de 256 vectores.

$$N = \frac{H \cdot W}{P^2} = \frac{256 \cdot 256}{16^2} = 256 \text{ patches} \tag{6.6}$$

Entonces, un parche de P x P pixeles y 3 canales de color (BGR) se convierte en un vector de tamaño (P x P x 3); en este caso, que el tamaño del parche elegido es de 16, significa que cada vector es de tamaño 16 x 16 x 3 = 768 elementos.

6.3.9 *Proyección lineal de cada parche*

Cada parche, entonces, se transforma en un vector de una dimensión. A ese proceso se le conoce como 'flattening' y después se debe proyectar en un espacio de dimensión fija D mediante una capa de proyección lineal.

Ese vector de 768 dimensiones se multiplica por una matriz de pesos para llevarlo a una dimensión fija llamada 'D'. Típicamente, según la literatura, D tiene un valor de 768, pero puede haber otros modelos donde D = 512 o 1024, dependiendo del tamaño del parche.

Figura 6.14: Ejemplo ilustrativo de división de una imagen en parches.

Este proceso entonces convierte una matriz de entrada de dimensión N (P x P x 3) en una matriz de (N, D) donde N es el número total de

parches y D es la dimensión de los embeddings.

El proceso es conceptualmente similar a cómo los modelos Transformer de procesamiento de lenguaje natural (NLP) interpretan las secuencias de palabras, ya que cada parche es tratado como un token en la secuencia de entrada.

6.3.10 *Codificación posicional de los vectores*

Ahora bien, como en las imágenes sí toma bastante importancia el orden en el que se encuentran los pixeles, se debe introducir en cada vector información sobre la ubicación de cada parche dentro de la imagen. Para ello, se utiliza una matriz de embedding posicional aprendible, donde cada posición i tiene asociado un vector de dimensión D, igual al de la proyección de parches. Matemáticamente, la representación del parche i-ésimo es x_i y su codificación posicional es p_i, la entrada final al modelo se define como:

$$z_i = x_i + p_i \tag{6.7}$$

Donde:

- $x_i \in \mathbb{R}^D$ es el embedding del parche.
- $p_i \in \mathbb{R}^D$ es el embedding posicional correspondiente.
- $z_i \in \mathbb{R}^D$ es el vector combinado que se introduce al modelo Transformer.

Esta estrategia asegura que el modelo pueda adquirir relaciones espaciales esenciales para la correcta clasificación.

6.3.11 *Bloque Transformer: Mecanismo de autoatención*

Posteriormente, cada vector es introducido al bloque principal de procesamiento del modelo, el encoder, que posee el mecanismo de autoatención; esto permite a cada token interactuar dinámicamente con los demás. Un modelo está formado por varios bloques de codificación y cada uno de ellos posee múltiples cabezas de atención, una red 'feed-forward' y la conexión entre ellos. Cada cabeza opera de forma paralela, aprendiendo diferentes patrones de dependencia, y sus resultados son posteriormente concatenados y combinados.

El objetivo de la autoatención es calcular una representación enriquecida de cada token, considerando no solo su propia información, sino también las relaciones de dependencia con otros tokens en la

imagen. De esta manera, el modelo puede capturar patrones globales y contextuales sin limitarse a un conjunto local.

El procedimiento de autoatención se puede resumir en los siguientes pasos:

1. Para cada vector de entrada z_i, se generan tres vectores distintos a través de transformaciones lineales: **Query** (Q), **Key** (K) y **Value** (V).

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V \tag{6.8}$$

   donde W_Q, W_K y W_V son matrices de pesos de tamaño $\mathbb{R}^{D \times d_k}$, y d_k es la dimensión del subespacio de atención utilizado por cada cabeza.

   - Q (Query): codifica lo que el token está "buscando" en los demás.
   - K (Key): representa lo que cada token "ofrece" a los otros.
   - V (Value): contiene la información que se transmite si un token recibe atención.

2. Se calcula una puntuación de atención entre el token i y todos los demás tokens, mediante el producto escalar de Q y K, escalado por la raíz cuadrada de la dimensión de las claves d_k:

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6.9}$$

   donde dk es un factor para escalar ese producto punto para que no crezca demasiado cuando dk es grande.

$$d_k = \left(\frac{D = 768}{h = cabezas}\right) \tag{6.10}$$

3. La función *softmax* asegura que los pesos de atención asignados a cada token sumen uno, permitiendo así un promedio ponderado de los valores V.

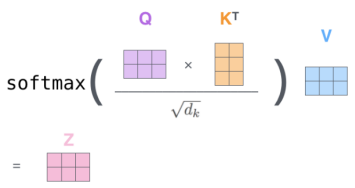

Figura 6.15: Ejemplo ilustrativo de las operaciones matriciales

Cabe resaltar que también se añade un vector adicional, **Token CLS** que actúa como un resumen global de la imagen y su representación final se utiliza para la clasificación.

Entonces, para este proyecto:

- Se procesan 257 tokens (256 parches + 1 token CLS).
- Cada parche originalmente es de dimensión 768 ($D = 768$).
- Se emplean 3 cabezas de atención ($h = 3$).
- Cada cabeza opera sobre un subespacio de dimensión:

$$head\_dim = \frac{D}{h} = \frac{768}{4} = 192 \tag{6.11}$$

La atención calcula la similitud entre Q y K para decidir qué proporción de información de V debe incluirse en la salida final de cada token. Es decir, la salida final para cada token es una combinación ponderada de los vectores V_j, utilizando los coeficientes α_{ij} como pesos:

$$\alpha_{ij} = \text{softmax}\left(\frac{Q_i \cdot K_j^T}{\sqrt{d_k}}\right) \tag{6.12}$$

$$\text{Attention}(Q, K, V)_i = \sum_{j=1}^{N} \alpha_{ij} V_j \tag{6.13}$$

Durante el entrenamiento del modelo, las matrices de pesos W_Q, W_K y W_V son optimizadas por retropropagación.

6.3.12 *Feed Forward Networks (FNN) y conexiones residuales*

Dentro del modelo, después de aplicar los mecanismos de autoatención, se incorpora una red neuronal densa que tiene como objetivo aplicar operaciones no lineales para que cada token pueda enriquecer su aprendizaje antes de pasar al siguiente bloque.

La estructura típica de un FFN en Vision Transformers consiste en dos capas densas con una función de activación intermedia. Matemáticamente, la operación se puede representar como:

$$\text{FFN}(x) = W_2 \cdot \sigma(W_1 \cdot x + b_1) + b_2 \tag{6.14}$$

donde:

- $x \in \mathbb{R}^D$ es el vector de entrada (token),
- $W_1 \in \mathbb{R}^{D \times D_{ff}}$ y $W_2 \in \mathbb{R}^{D_{ff} \times D}$ son matrices de pesos aprendibles,
- b_1, b_2 son los vectores de sesgo,
- $\sigma(\cdot)$ es una función de activación, comúnmente ReLU o GELU,
- D_{ff} es la dimensión oculta, usualmente 2 a 4 veces mayor que D.

Para facilitar el entrenamiento de redes profundas, cada subcomponente dentro del bloque Transformer (tanto la autoatención como el FFN) está rodeado por una estructura que combina:

- **Normalización por capas** (*Layer Normalization*),
- **Conexiones residuales** (*residual connections*).

La operación general de un sub-bloque sigue el patrón:

$$\text{Output} = x + \text{SubBloque}(\text{LayerNorm}(x)) \tag{6.15}$$

Esto significa que se normaliza la entrada, se pasa por el subbloque funcional (atención o FFN), y se suma la salida al valor original de entrada. Este esquema ayuda a:

- Mantener el flujo de gradientes en redes profundas.
- Evitar la degradación de información relevante.
- Acelerar la convergencia durante el entrenamiento.

6.3.13 *Resumen de un bloque Transformer*

Cada bloque encoder en ViT incluye dos subcomponentes principales, aplicados secuencialmente:

1. Mecanismo de autoatención con:
   - LayerNorm
   - Multi-head attention
   - Suma residual
2. Feed Forward Network con:
   - LayerNorm
   - Dos capas densas con activación
   - Suma residual

Este diseño modular y repetitivo puede apilarse múltiples veces lo que permite que Vision Transformers escalen a modelos de mayor tamaño y profundidad.

Al finalizar los bloques Transformer, se extrae la salida correspondiente al token CLS y se utiliza como entrada para una o más capas densas que realizan la predicción final. Esta capa de salida suele tener una función de activación `softmax` (para clasificación multiclase) o `sigmoid` (para clasificación binaria). En este trabajo, se utilizó una función `softmax` para distinguir entre dos clases: imágenes reales e imágenes DeepFake.

La operación se representa matemáticamente como:

$$\hat{y} = \text{softmax}(W \cdot z_{CLS} + b) \tag{6.16}$$

donde:

- $z_{CLS} \in \mathbb{R}^D$: vector final del token CLS después del último bloque Transformer.
- $W \in \mathbb{R}^{D \times C}$: matriz de pesos de la capa de salida.
- $b \in \mathbb{R}^C$: vector de sesgo
- $\hat{y} \in \mathbb{R}^C$: vector de probabilidades para cada clase.

Para mitigar el riesgo de sobreajuste, se incluye una capa de *Dropout* con probabilidad de 0.2 antes de la salida.

6.4 Arquitectura e hiper parámetros

Función de pérdida

Se utilizó la función de pérdida **Categorical Crossentropy**, implementada en *TensorFlow/Keras*. Esta función mide la discrepancia entre la distribución de probabilidades predicha por el modelo ($\hat{y}$) y las etiquetas verdaderas (y).

Optimizador y tasa de aprendizaje

Se utilizó el optimizador **Adam** (Adaptive Moment Estimation), ampliamente empleado en redes neuronales profundas debido a su capacidad de ajustar dinámicamente las tasas de aprendizaje para cada parámetro.

Regularización y prevención de sobreajuste

Para mitigar el riesgo de sobreajuste debido al tamaño relativamente reducido del conjunto de entrenamiento, se aplicaron las siguientes estrategias:

- **Dropout**: se introdujo una probabilidad del 20% de desactivación de neuronas después del *pooling global*.
- **Early stopping**: se monitoreó la métrica de validación para detener el entrenamiento si no había mejora tras varias épocas consecutivas.
- **Aumento de datos (data augmentation)**: aplicado en la fase de preprocesamiento (rotación, cambio de brillo, recorte aleatorio) para incrementar la diversidad del conjunto de entrenamiento.

6.5 Resultados

Los resultados de cada modelo con los diferentes hiperparámetros fueron los siguientes:

Modelo	Imágenes	LR	Cabezas	Capas	Batch	Épocas	Optimizador	Regularización	Accuracy	Recall	F1	Time
1	500	0.01	4	5	32	100	AdamW	dropout_rate(0.1)	99%	99%	99%	5 min
2	1,000	0.001	4	5	32	100	AdamW	dropout_rate(0.1)	99%	99%	99%	7 min
3	1,500	Dinámico -1.56E-05	8	7	32	100	AdamW	dropout_rate(0.2)	65%	65%	65%	30 min
4	140,000	Dinámico	8	7	32	100	AdamW	dropout_rate(0.2)	50%	51%	50%	15 h
5	140,000	Dinámico	8	7	32	100	AdamW	dropout_rate(0.2)	50%	51%	51%	> 20 h

Figura 6.16: Modelos y sus hiperparámetros.

Como se puede observar, los primeros 2 modelos obtuvieron métricas demasiado altas, lo que sugiere que cada modelo se sobreajustó al memorizar prácticamente todos los ejemplos disponibles, perdiendo la capacidad de generalizar.

Gradualmente, se incrementó la cantidad de datos y el rendimiento comenzó con caídas significativas. Esto podría explicarse por una arquitectura no óptima en el modelo y que los datos no fueron suficientes.

Para los modelos 5 y 6 se utilizaron 140,000 imágenes y se ajustaron la tasa de aprendizaje y el tamaño de batch. Sin embargo, los resultados no mejoraron: la precisión y el recall se mantuvieron alrededor del 50%, lo que indica que el modelo no aprendió a distinguir entre clases. El tiempo de entrenamiento aumentó significativamente (más de 15 y 20 horas), lo que sugiere que, aunque hubo mayor esfuerzo computacional, no se logró extraer conocimiento útil.

Ahora bien, centrándonos en el modelo 3, que tuvo métricas un poco más realistas o esperadas, quise realizar más pruebas utilizando este modelo.

Obteniendo un reporte en sus clasificaciones, se obtuvo lo siguiente:

```
Classification Report:
              precision    recall  f1-score   support

        Real       0.61      0.72      0.66      1289
        Fake       0.69      0.58      0.63      1411

    accuracy                           0.65      2700
   macro avg       0.65      0.65      0.65      2700
weighted avg       0.65      0.65      0.64      2700
```

Figura 6.17: Modelo 3 y sus métricas

La matriz de confusión para este modelo, se ve de la siguiente manera:

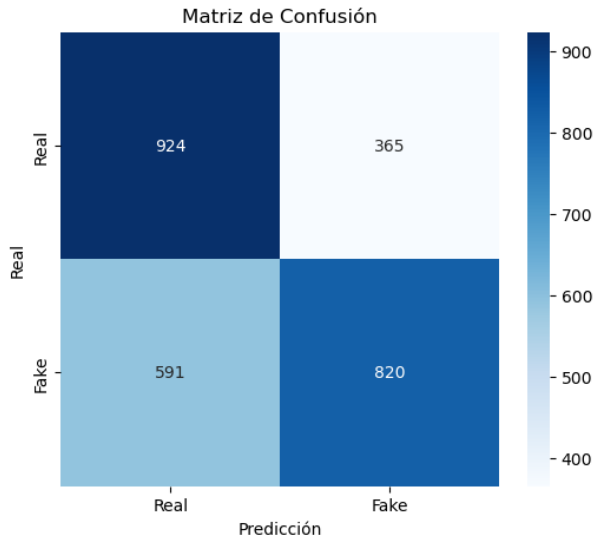

Figura 6.18: Modelo 3 y su matriz de confusión

La evolución de pérdida se ve de la siguiente manera:

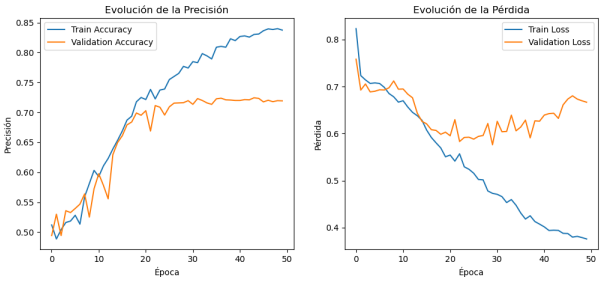

Figura 6.19: Modelo 3 y sus aprendizajes

Podemos observar que en la fase de entrenamiento comenzó a aprender correctamente; sin embargo, en la validación comenzó a flaquear el modelo.

Al momento de realizar sus predicciones, le fue mejor con las imágenes reales (que hace sentido basado en sus métricas, que con las falsas).

Figura 6.20: Modelo 3 y sus predicciones. Las imágenes sí eran "fake", "real", "fake" en ese orden.

Figura 6.21: Modelo 3 y sus predicciones. La imagen era "fake"

Para concluir, a pesar de que el modelo ViT exhibió un rendimiento óptimo en conjuntos de datos reducidos (por sobreajuste), no logró generalizar en contextos realistas con un volumen considerable de datos. Esto indica que la arquitectura implementada requiere una reevaluación y experimentar más.

Recomendaciones para futuros experimentos

- Aumentar la profundidad del modelo (más bloques Transformer).
- Probar otras dimensiones de embedding y mayor número de cabezas de atención.
- Aplicar técnicas de *fine-tuning* sobre modelos preentrenados.
- Evaluar la discriminación visual de los datos antes del entrenamiento.
- **Tamaño de patch:** Un tamaño de patch pudo haber sido demasiado grande, perdiendo detalles finos relevantes para detectar manipulación.

7 Análisis de resultados y conclusiones

Contenido

7.1 Trabajo a futuro 70

La creación y capacitación del modelo fundamentado en Vision Transformers (ViT) para la categorización de imágenes reales y generadas a través de técnicas de *DeepFake* posibilitó la exploración del potencial de esta arquitectura en un problema de creciente importancia. No obstante, los hallazgos derivados de la fase de pruebas sugieren que el modelo no logró generalizar de manera apropiada, exhibiendo un desempeño inferior al anticipado en métricas como precisión, recall y F1-Score.

En general, se cumplió el objetivo de implementar un modelo basado en Vision Transformers para la clasificación de imágenes. Además, se implementó también un modelo preentrenado para la detección y recorte de rostro como parte del procesamiento y se analizaron varios intentos con sus métricas. Desafortunadamente, el aprendizaje y desempeño en los últimos modelos no fue el esperado.

A pesar de las variaciones intentadas en el número de imágenes y los híperparámetros, no logró generalizar adecuadamente. En los primeros modelos, las métricas podrían explicarse debido a la pequeña cantidad de muestras; además, da indicios de un sobreajuste debido a que son tan pocas muestras que el modelo simplemente memorizó las imágenes en lugar de aprender patrones generales.

Con el aumento de muestras, se esperaba que el modelo pudiera captar patrones más generales del conjunto de datos; sin embargo, la disminución constante de las métricas nos dice que el modelo no está siendo capaz de adaptarse adecuadamente a la complejidad de los datos. Incluso en el último modelo, se tuvo que detener el proceso debido a la cantidad de horas que llevaba el proceso. Definitivamente, el poder computacional actual no fue suficiente.

También es justo mencionar que la resolución del conjunto de datos, al ser imágenes tan pequeñas, pudo de alguna manera impactar en el aprendizaje del modelo. Si estas imágenes recibieron algún proceso de compresión, pudieron haberse perdido características sutiles, haciendo muy difícil diferenciar una categoría de otra. Además, es cierto que el conjunto de datos tenía variabilidad alta en cuestiones como ruido, iluminación, ángulos de los rostros e incluso hay algunas imágenes a las que se les añadió ruido artificial como el método sal y pimienta o algún filtro.

Por otro lado, si la resolución de las imágenes no fue el principal problema, entonces es posible que la arquitectura elegida no haya sido la mejor para el conjunto de datos. Quizá el modelo no haya tenido la suficiente profundidad o número de cabezas de atención que capturen patrones relevantes suficientes.

Finalmente, entre las acciones que podrían haberse implementado para mejorar los resultados destacan:

- **Arquitecturas alternativas:** Modelos más adaptados a visión por computadora como Swin Transformer, o híbridos CNN-Transformer, podrían haber ofrecido mejor desempeño.
- **Preentrenamiento y fine-tuning:** Utilizar un ViT preentrenado en un dataset grande (como ImageNet) y luego aplicar fine-tuning específico en el conjunto de DeepFakes habría facilitado un mejor punto de partida.
- **Exploración más exhaustiva de hiperparámetros:** Aplicar técnicas como grid search o Bayesian optimization para encontrar combinaciones óptimas de LR, batch size, weight decay, dropout, entre otros.

Para concluir, a pesar de que el rendimiento final del modelo no logró los aprendizajes como se esperaba, la implementación de estos fascinantes modelos ofreció conocimientos valiosos acerca de las condiciones requeridas para la capacitación de Vision Transformers en tareas de clasificación de contenido manipulado, facilitando la posibilidad de mejoras metodológicas futuras.

7.1 *Trabajo a futuro*

Sin duda, ViT es una técnica con un potencial enorme y ese poder demanda una cantidad de datos y poder computacional de la misma magnitud. Algunos de los modelos preentrenados ya existentes de ViT

fueron entrenados con millones de imágenes; sin embargo, los más utilizados y poderosos están únicamente entrenados para identificar objetos o animales. No se encontraron modelos preentrenados, abiertos al público en general, con rostros de personas para poder hacer una comparativa clara.

Definitivamente, queda como trabajo a futuro el poder contar con bases de datos públicas y reguladas de imágenes de rostros de personas que están dispuestas a compartir su información. A la vez, intentar añadir más profundidad a la arquitectura del modelo, añadiendo más cabezas de atención y aplicar técnicas más robustas de regularización para que el modelo sea capaz de generalizar adecuadamente. Adicionalmente, tener muy presente la parte ética detrás del uso de esta información biométrica de cada persona involucrada.

Por último, utilizar más poder computacional, quizá con la ayuda de servicios externos privados que tengan la infraestructura necesaria para entrenar y ejecutar este tipo de modelos.

8 Bibliografía y Referencias

[1] A. Diana, "Deepfake: la violencia machista parece no tener límites - CRUCE," Marzo 2025.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," 6 2017.

[3] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3rd ed., 2010.

[4] IBM, "What is machine learning?."

[5] IBM, "What is a neural network?."

[6] IBM, "What is deep learning?."

[7] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2002.

[9] IBM, "What is computer vision?."

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] Amazon Web Services, Inc., "¿qué es la ia generativa? - explicación de la ia generativa - aws."

[12] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1097–1105, 2012.

[13] A. Famili, W.-M. Shen, R. Weber, and E. Simoudis, "Data preprocessing and intelligent data analysis," *Intelligent Data Analysis*, vol. 1, no. 1, pp. 3–23, 1997.

[14] IBM, "Computer vision," Jan. 2025.

[15] Y. Huo, K. Jin, J. Cai, H. Xiong, and J. Pang, "Vision transformer (vit)-based applications in image classification," in *2023 IEEE 9th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, pp. 135–140, 2023.

[16] B. Cruz, "2024 deepfakes guide and statistics," Sept. 2024.

[17] J. Dhaliwal, "State of the scamiverse – how ai is revolutionizing online fraud," Jan. 2025.

[18] M. S. Rana, M. N. Nobi, B. Murali, and A. H. Sung, "Deepfake Detection: A Systematic Literature Review," 2022.

[19] H. Taha, S. Luisa, V. Nasir, and M. Editors, "Advances in Computer Vision and Pattern Recognition," tech. rep.

[20] Google, "Classification: Accuracy, recall, precision, and related metrics," 3 2025.

[21] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 511–518, IEEE, 2001.

[22] Siwei Lyu, *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. Institute of Electrical and Electronics Engineers (IEEE), 2020.

[23] H. Taha, S. Luisa, V. Nasir, and M. Editors, "Advances in Computer Vision and Pattern Recognition," tech. rep.

[24] H. Taha, S. Luisa, V. Nasir, and M. Editors, "Advances in Computer Vision and Pattern Recognition," tech. rep., 2022.

[25] S. R. Ahmed, E. Sonuc, M. R. Ahmed, and A. D. Duru, "Analysis Survey on Deepfake detection and Recognition with Convolutional Neural Networks," in *HORA 2022 - 4th International Congress on Human-Computer Interaction, Optimization and Robotic Applications, Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2022.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," 6 2017.

[27] J. John and B. V. Sherif, "Comparative Analysis on Different DeepFake Detection Methods and Semi Supervised GAN

Architecture for DeepFake Detection," in *6th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2022 - Proceedings*, pp. 516–521, Institute of Electrical and Electronics Engineers Inc., 2022.

[28] R. P. G. Designed Research; M, "Deepfake detection by human crowds, machines, and machine-informed crowds," 2021.

[29] Siwei Lyu, *DEEPFAKE DETECTION: CURRENT CHALLENGES AND NEXT STEPS*. IEEE, 2020.

[30] A. Koçak and M. Alkan, "Deepfake Generation, Detection and Datasets: A Rapid-review," in *15th International Conference on Information Security and Cryptography, ISCTURKEY 2022 - Proceedings*, pp. 86–91, Institute of Electrical and Electronics Engineers Inc., 2022.

[31] B. Ghita, I. Kuzminykh, A. Usama, T. Bakhshi, and J. Marchang, "Deepfake Image Detection using Vision Transformer Models," tech. rep.

[32] Sashank Sridhar, Siddartha Mootha, and Sudha Subramanian, "Detection of Market Manipulation using Ensemble Neural Networks," tech. rep., 2020.

[33] H. Mark Nguyen, R. Derakhshani, s. Hoang Nguyen, and n. Reza Derakhshani, "Eyebrow Recognition for Identifying Deepfake Videos," tech. rep., 2021.

[34] A. Matheven and B. V. D. Kumar, "Fake News Detection Using Deep Learning and Natural Language Processing," in *2022 9th International Conference on Soft Computing and Machine Intelligence, ISCMI 2022*, pp. 11–14, Institute of Electrical and Electronics Engineers Inc., 2022.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," tech. rep.

[36] Artem A. Maksutov, Viacheslav O. Morozov, Aleksander A. Lavrenov, and Alexander S. Smirnov, *Methods of Deepfake Detection based on Machine Learning*. IEEE, 2020.

[37] S. Shaposhnikov, *Proceedings of the 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus) : January 27-30, 2020, St. Petersburg and Moscow, Russia*. St. Petersburg Electrotechnical University "LETI", 2020.

[38] J. M. Ashraf, S. A. Hadi, A. Rezk, N. A. Madjid, W. Alnaqbi, A. Alhammadi, and A. Nayfeh, "Using Otsu's Method for Image Segmentation to Determine the Particle Density, Surface Coverage and Cluster Size Distribution of 3 nm Si Nanoparticles," *IEEE Transactions on Nanotechnology*, vol. 20, pp. 765–774, 2021.

[39] K. Lu, Y. Xu, and Y. Yang, "Comparison of the potential between transformer and cnn in image classification," in *ICMLCA 2021; 2nd International Conference on Machine Learning and Computer Application*, pp. 1–6, 2021.

[40] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[41] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Deep learning for deepfakes creation and detection: A survey," *arXiv preprint arXiv:1909.11573*, 2019.

[42] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[43] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2672–2680, 2014.

[44] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," *arXiv preprint arXiv:1901.08971*, 2019.

[45] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[46] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2nd ed., 2018.

[47] A. Bunn, "Artificial imposters—cybercriminals turn to ai voice cloning for a new breed of scam," Sept. 2024.

[48] M. Westerlund, "The emergence of deepfake technology: A review," *Technology Innovation Management Review*, vol. 9, pp. 40–53, 11/2019 2019.