Método de detección de Deepfake mediante técnicas de Machine Learning

Estudiantes:

Bayron Daymiro Campaz Hurtado

Juan David Diaz Monsalve

Santiago Gutierrez Bolaños

Tutor:

Christian Camilo Urcuquí López, Msc

Universidad ICESI
Facultad de Ingeniería
Ingeniería de Sistemas
Cali
2020

ABSTRACT

In recent years artificial intelligence has advanced at an accelerated rate, these advances have been used for different purposes such as the manipulation of images and vídeos. This type of manipulation performed for spoofing purposes is what we know as Deepfake. This practice can be used for blackmail, intimidation, and even generate repercussions at a political and social level.

There are currently several detection methods that work well for first-generation datasets which feature poor-quality Deepfakes, however they do not offer very good results for some superior next-generation datasets as Deepfake. In this project, a Deepfake detection method based on a convolutional Xception network that uses Transfer Learning, through ImageNet, is proposed. This network is trained with faces drawn from various first and second generation Deepfake datasets. With this configuration, an accuracy of 92.12% and an AUC of 92.15% were obtained for the data sets.

RESUMEN

En los últimos años la inteligencia artificial ha avanzado a un ritmo acelerado, estos avances han sido usados para diferentes propósitos como, la manipulación de imágenes y vídeos. Este tipo de manipulación realizada con objetivos de suplantación es lo que conocemos como Deepfake. Esta práctica puede ser usada para el chantaje, la intimidación, e incluso generar repercusiones a nivel político y social.

Actualmente existen diversos métodos de detección que funcionan bien para conjuntos de datos de primera generación, los cuales presentan Deepfakes de mala calidad, sin embargo, no ofrecen muy buenos resultados para algunos conjuntos de datos de última generación superiores en calidad de Deepfake. En este proyecto se propone un método de detección de Deepfake basado en una red convolucional Xception que hace uso de Transfer Learning, mediante ImageNet. Esta red se entrena con los rostros extraídos de varios conjuntos de datos de primera y segunda generación de Deepfake. Con esta configuración se obtuvo un accuracy de 92,12% y un AUC del 92,15% para los conjuntos de datos.

CONTENIDO

LISTA DE ACRÓNIMOS

IA	Inteligencia Artificial
GAN	Red generativa antagónica. (Generative Adversarial Network)
CNN	Red neuronal convolucional. (Convolutional Neural Network)
RNN	Red neuronal recurrente. (Recurrent Neural Network)
SVM	Máquina de Soporte de Vectores. (Support Vector Machine)
AUC	Área bajo la curva del ROC. (Area Under the ROC Curve)
ELA	Análisis de Nivel de Error (Error Level Analysis)

GLOSARIO DE TÉRMINOS

Método: Es un conjunto de estrategias, herramientas y procedimientos que usados de manera sistemática y ordenada permiten llegar a un fin o resultado determinado.

Modelo: En machine learning es una estructura que puede estar representada en un archivo que se ha entrenado para reconocer determinados tipos de patrones.

Sesgo: Es la desviación de un valor estimado respecto al valor real.

Instancia: Representa un objeto único de datos implícitamente estructurados en un conjunto de datos.

Vector: Término empleado para referirse a un arreglo de números que describen una específica combinación de propiedades.

Topología: En redes neuronales consiste en la organización y disposición de las neuronas en una red formando capas o agrupaciones de neuronas.

Accuracy: Es la fracción de predicciones que se realizaron correctamente en un modelo de clasificación.

Código abierto: Término empleado para referirse al software que permite el acceso a su código de programación, lo que facilita modificaciones por parte de otros programadores ajenos a los creadores originales del software en cuestión.

Estocástico: Término empleado para referirse a un sistema cuyo comportamiento no es determinista, es decir que su comportamiento está definido por el azar.

Interpolación: Término empleado para referirse a obtención de nuevos puntos partiendo del conocimiento de un conjunto de puntos

Artefacto visual: Anomalía en una representación gráfica o visual ya sea en un gráfico digital o una imagen.

Compresión: Codificación de información con menos volumen que la información original, con el objetivo de disminuir la cantidad de espacio ocupado.

Mesoscópico: Escala o contexto en el que se puede observar de forma razonable las propiedades de un objeto de estudio.

Transfer Learning: Transferencia del conocimiento adquirido resolviendo un problema para usarlo en la solución de un problema relacionado.

Sintetización: Resultado de reunir distintos elementos que estaban dispersos o separados, organizándolos y relacionándolos.

Minería de datos: Conjunto de técnicas que permiten explorar una cantidad de datos, con el objetivo de encontrar patrones, correlaciones, tendencias o reglas, que permitan entenderlos o hacer predicción con base en ellos.

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. MOTIVACIÓN Y ANTECEDENTES

1.1 Contexto

En las últimas décadas la inteligencia artificial **(IA)** ha avanzado a un ritmo acelerado. Esto se debe principalmente al mejoramiento en la capacidad de procesamiento y a las aplicaciones de esta tecnología en diversas áreas del conocimiento.

Machine Learning es una rama de la **IA** que posee un área denominada **Deep Learning** está última ha permitido grandes avances en el procesamiento de contenido visual. Uno de los más recientes usos que se les ha dado a los algoritmos de Deep Learning ha sido manipular imágenes y vídeos; por ejemplo, insertando la imagen de un rostro de una persona en la imagen del rostro de otra. La manipulación en este tipo de contenido que usa estos algoritmos se denomina **Deepfake.**

Estas tecnologías son cada vez más accesibles, y "la creciente facilidad de crear contenido de audio y vídeo falso crea amplias oportunidades para el chantaje, la intimidación y el sabotaje" (Chesney & Citron, 2019), además de las posibles repercusiones a nivel político y social.

1.2 Antecedentes del problema

La herramienta que ha permitido al Deepfake producir resultados realistas con la capacidad de engañar a las personas son las **Redes Neuronales** que se pueden usar para procesar imágenes; lo que incluye el intercambio de rostros y la generación de nuevos rostros instantáneamente (luego de un entrenamiento previo). Esta tecnología es uno de los avances más prolíficos en IA, dado que permite simular el cerebro humano de manera que una máquina pueda lograr un nivel de aprendizaje con un alto grado de detalle.

La idea del Deepfake comienza en 2012 durante una competencia de ImageNet en donde el ganador usa por primera vez un algoritmo de red neuronal profunda. En 2014 Ian Goodfellow crea las redes generativas antagónicas (GAN's), que consiste en un sistema de inteligencia artificial que puede crear imágenes completamente nuevas. Pero es hasta noviembre de 2017 donde un usuario de Reddit llamado "Deepfake" pública vídeos pornográficos que hacían uso de rostros de celebridades. A lo largo de 2018 el contenido aparece en cientos de artículos de prensa como: The New York Times, Washington Post, The Guardian, The Economist, The Times y The BBC.

En el mismo 2018 BuzzFeed publica un vídeo falso de Barack Obama con la colaboración del actor Jordan Peele. "En abril, el director de cine Jordan Peele y BuzzFeed lanzan un vídeo en el cual se observa a Barack Obama insultando a Donald Trump, esto con el fin de crear conciencia sobre cómo los contenidos sintéticos generados por la IA podrían usarse para distorsionar y manipular la realidad" (Schwartz O., 2018).

En síntesis, estas situaciones en las cuales se han involucrado elementos tales como vídeos que toman a personas que poseen influencia en el mundo actual no debe pasarse por alto, dado que esta actividad continúa y se amplifica hasta hacerse más cercana y de mayor acceso, aumentando la participación alrededor del mundo en el uso de Deepfake.

1.3 Justificación

El Deepfake puede representar un gran problema a nivel social dado que tiene un potencial de provocación elevado, imagine un vídeo del primer mandatario de un país expresando un discurso con contenido bélico o realizando un discurso de odio contra un grupo social, "*en un mundo ya preparado para la violencia, tales grabaciones tendrán un gran potencial de incitación*" (Chesney & Citron, 2019).

Deepfake podría ser usado en la contienda política para modificar las intenciones de voto o para afectar la imagen de una figura pública, "Estos avances amenazan con desdibujar aún más la línea entre la verdad y la ficción en política" (Weems, 2017). El uso de la tecnología para alterar contenido hace que se vuelva cada vez más difícil diferenciar entre lo que es falso y lo que no.

"Lo que distingue a Deepfake de otras técnicas de manipulación de vídeo es, en primer lugar, su potencial para obtener resultados fotorrealistas" (Koopman, Rodriguez, & Geradts, 2018); estos impactantes resultados, aunado al hecho de la accesibilidad y facilidad de uso de esta técnica, permiten que casi cualquier persona pueda emplearla. Esto hace necesario el desarrollo de métodos que nos ayuden a distinguir el contenido auténtico del alterado con Deepfake.

Existen diferentes modelos que emplean distintas tecnologías para hacer Deepfake, es por esto por lo que desarrollar métodos que permitan detectar el contenido alterado para estos distintos tipos de modelos, brindará la posibilidad de dictaminar la veracidad de estos, y disminuirá la posibilidad de engaño y la manipulación de la opinión.

2 DESCRIPCIÓN DEL PROBLEMA

Vivimos en un mundo interconectado en donde tenemos acceso a una gran cantidad de información; desafortunadamente esta información es propensa a ser alterada o falsificada. Una de las técnicas más recientes y con la que se han obtenido resultados impactantes es el Deepfake que permite falsificar contenido de audio, imágenes y vídeo.

Actualmente existe una gran dificultad para discernir la suplantación mediante el uso de Deepfake en conjuntos de datos de vídeos e imágenes de última generación, esto se debe principalmente a los grandes avances de la inteligencia artificial, la facilidad de acceso y uso de herramientas que implementan esta técnica y al sesgo que tienen los métodos de detección actuales a imágenes de baja calidad de Deepfake. Como consecuencia este problema puede generar la afectación negativa en la imagen de las víctimas, la manipulación de la información, un cambio en las percepciones y decisiones de las personas e incluso la incitación a la violencia.

3 OBJETIVOS

3.1 Objetivo General

Evaluar un método de detección de Deepfake que emplee técnicas de Machine Learning que permita distinguir entre contenido visual alterado y contenido visual auténtico.

3.2 Objetivos Específicos

- Analizar el funcionamiento y las características de métodos generadores y detectores de Deepfake.

- Recopilar un conjunto de datos reales y alterados.

- Implementar un método de detección de Deepfake.

- Evaluar el método propuesto y compararlo con otros métodos existentes.

4 MARCO TEÓRICO

Para exponer el marco teórico del proyecto, se abordará la definición de: Inteligencia Artificial, Machine Learning, Aprendizaje no supervisado, Aprendizaje supervisado, Protocolos de evaluación, Métricas de evaluación, Aprendizaje semi-supervisado, Aprendizaje por refuerzo, Redes Neuronales Artificiales, Deep Learning, Red Neuronal Xception, Modelo generativo, Generative Adversarial Network (GAN), Ciberseguridad, Métodos generadores de Deepfake.

4.1 Inteligencia Artificial

La inteligencia artificial es el área de las ciencias de la computación que se encarga de la simulación del comportamiento inteligente humano en las máquinas. Con inteligencia humana nos referimos a procesos de percepción sensorial (visión, audición etc.) y a sus consiguientes procesos de reconocimiento de patrones (Benítez, Escudero, Kanaan, & Rodó, 2014).

Para lograr los procesos anteriormente mencionados se hace necesario la construcción de sistemas inteligentes. Un sistema se considera inteligente si presenta las características de la conducta inteligente las cuales son: el aprendizaje, el razonamiento y la adaptación. Además, se debe tener en cuenta dos elementos adicionales: la ontología general del sistema y el medio ambiente en el que operará. (Romero, J. J., Dafonte, C., Gòmez, Á., & Penousal, F. J., 2007).

“La ontología general se refiere (...) a todo lo que el sistema es capaz de reconocer e interpretar del medioambiente y que le es útil para tomar una decisión.” (Romero, J. J., Dafonte, C., Gòmez, Á., & Penousal, F. J., 2007). Mientras que el medio ambiente es el sistema en donde se encuentra inmerso el sistema inteligente es decir el sistema inteligente es un subsistema del medio ambiente.

Cuando se habla de aprendizaje se hace referencia a los cambios en la forma de reaccionar de un ente frente a una situación experimentada anteriormente. El razonamiento es la operación en la que partiendo de uno o varios juicios (premisas), se infiere un nuevo juicio que se desprende lógicamente de las premisas. Por otro lado, la adaptación es la capacidad de responder a un determinado objeto o evento a través de una conexión estímulo-reacción.

4.2 Machine Learning

Machine Learning es un campo de estudio de la Inteligencia Artificial que busca darle la habilidad de aprendizaje a las computadoras sin ser explícitamente programadas. Usando datos, ejemplos o instrucciones, se buscan patrones con el objetivo de tomar mejores decisiones con base en los ejemplos provistos.

El machine learning se divide en cuatro tipos diferentes de aprendizaje; el aprendizaje no supervisado, el aprendizaje supervisado, el aprendizaje semi-supervisado y el aprendizaje por refuerzo. (Burkov, 2019)

4.3 Aprendizaje no supervisado

Los algoritmos basados en aprendizaje no supervisado manejan un conjunto de datos que contiene muchas variables de una instancia, las cuales usa para aprender propiedades útiles de la estructura de ese conjunto de datos. (Goodfellow, Bengio, Courville, 2014)

4.4 Aprendizaje supervisado

Los algoritmos de aprendizaje supervisados manejan un conjunto de datos que contiene diversas variables, donde cada instancia está asociada con una etiqueta u objetivo, que permite que el algoritmo use estos datos para predecir el valor de una nueva instancia.

Formalmente el conjunto de datos es una colección de ejemplos etiquetados $\{(x_i, y_i)\}_{i=1}^{N}$. Cada elemento x_i entre N se llama un vector de características el cual contiene un valor que describe el ejemplo de alguna manera. Ese valor se llama característica y se denota como $x^{(j)}$. Para todos los ejemplos en el conjunto de datos, la característica en la posición j en el vector de características siempre contiene el mismo tipo de información. La etiqueta y_i puede ser un elemento que pertenece a un conjunto finito de clases $\{1,2,\dots,C\}$, o un número real, o a una estructura más compleja, como un vector, una matriz, un árbol o un gráfico. El objetivo de un algoritmo de aprendizaje supervisado es utilizar el conjunto de datos para producir un modelo que tome un vector de características x como información de entrada y salida que permita deducir la etiqueta para este vector de características. (Burkov, 2019)

Finalmente, el algoritmo produce un modelo que toma un vector de características x como información de entrada y genera como salida una posible etiqueta para este vector de características.

4.4.1 Protocolos de evaluación

Existen protocolos para la evaluación de la capacidad de generalización de los modelos y que están destinados a evitar el sesgo generado por usar el mismo conjunto de datos, entre los más usados se encuentran el Hold-out y el cross-validation.

El Hold-out consiste en realizar una partición del conjunto de datos en dos partes, una es el conjunto de entrenamiento que corresponden a los datos usados por el modelo

para el aprendizaje y la otra el conjunto de evaluación o test que se usa para evaluar que tan bien predice un modelo un conjunto de datos nuevos.

K-fold cross-validation realiza una partición del conjunto de datos en K conjuntos disyuntos del mismo tamaño. K − 1 partes se utilizan para el entrenamiento, y 1 parte para la validación o test. Este proceso se repite K veces y finalmente se agregan las métricas de evaluación. Se estima que los mejores resultados se obtienen con un valor de K entre 5 y 10.

Cabe aclarar que en ambos casos es necesario que la selección de los datos sea de forma aleatoria, de esta manera se asegura que la selección represente fielmente al conjunto de datos del cual proviene.

4.4.2 Métricas de evaluación

Una vez construido el modelo y evaluados los datos de validación es necesario hacer uso de métricas de evaluación que nos permitan cuantificar que tan bueno es el modelo. Actualmente existen diversas métricas formales y herramientas para este cometido. Entre estas tenemos la matriz de confusión, el kappa, el accuracy y el AUC.

La matriz de confusión es una tabla que resume cuán exitoso es el modelo de clasificación para predecir ejemplos que pertenecen a varias clases. Un eje de la matriz de confusión es la etiqueta que predijo el modelo, y el otro eje es la etiqueta real.

		Clase predicha		
		Positivo(1)	Negativo(0)	Total
Clase verdadera	Positivo(1)	VP	FN	$VP + FN$
	Negativo(0)	FP	VN	$FP + VN$
	Total	$VP + FP$	$FN + VN$	N

Figura 1. Matriz de confusión

En el gráfico de la figura X VP (verdaderos positivos) y VN (verdaderos negativos) representan los datos clasificados correctamente mientras que por otro lado FP (falsos positivos) y FN (falsos negativos) representan los datos clasificados erróneamente. De esta matriz se pueden calcular otras métricas útiles para evaluar el modelo, una de ella es el accuracy el cual refleja que porcentaje de los datos fueron bien clasificados. En términos de la matriz de confusión se define como $(VP + VN) / (VP + VN + FP + FN)$. (Burkov, 2019)

Por otra parte, el kappa es la medida estadística que ajusta los valores de azar. Esta medida se usa en elementos categóricos o cualitativos. “Sustrae el efecto de concordancia por suerte (AC) del valor del accuracy (concordancia observada OA)” (Diaz, 2019).

Otra métrica muy usada es el AUC (Area Under ROC Curve), la curva ROC es una combinación de la tasa de verdaderos positivos $(VP/(VP + FN))$ y la tasa de falsos positivos $(FP/(FP + VN))$ esta curva indica el rendimiento de un modelo de clasificación de acuerdo a como varía el umbral de clasificación (valor a partir del cual decidimos que un caso es un positivo). El AUC que es el área bajo la curva ROC se puede interpretar como la probabilidad de que el modelo clasifique un ejemplo positivo aleatorio más alto que un ejemplo negativo aleatorio. Así que un modelo es mejor mientras más alto sea su valor de AUC. (Google, 2020)

4.5 Aprendizaje semi-supervisado

Los algoritmos de aprendizaje semi-supervisados manejan un conjunto de datos que contienen tanto datos etiquetados como datos no etiquetados, este tipo de aprendizaje busca crear un modelo que permita predecir el valor de una nueva instancia al igual que el aprendizaje supervisado.

4.6 Aprendizaje por refuerzo

El aprendizaje por refuerzo se da a través de la interacción de un sistema con su entorno, en donde este sistema es capaz de percibir el estado de su entorno como un vector de propiedades. Este sistema puede ejecutar acciones en este estado. Diferentes acciones traen consigo diferentes recompensas, y esto genera que el sistema pase a otro estado dentro de su entorno. La meta del aprendizaje por refuerzo es aprender una política, esta última es una función f que toma el vector de características de un estado como entrada y genera una acción óptima para ejecutar en ese estado. La acción es óptima si maximiza la recompensa promedio esperada. (Burkov, 2019)

4.7 Redes Neuronales Artificiales

Como su nombre lo indica, las redes neuronales artificiales son en términos generales, redes computacionales las cuales intentan simular el proceso de decisión que hacen las neuronas del sistema nervioso central biológico. (Daniel Graupe, 2017).

Se tiene entonces que, así como en el sistema nervioso central biológico existe una unidad fundamental como lo es la neurona, en una red neuronal artificial se tiene un análogo llamado perceptrón.

Un perceptrón tiene varias entradas y las combina, normalmente con una suma básica. La suma de las entradas es modificada por una función de transferencia y el valor de la salida de esta función de transferencia se pasa directamente a la salida del perceptrón. (Xabier Basogain, 2008). Esto se puede apreciar en la Figura 2.

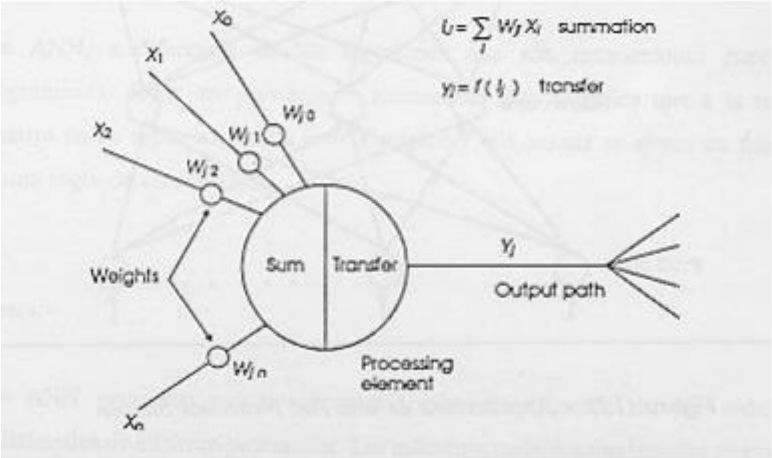

Figura 2. Diagrama de neurona artificial

La salida del perceptrón se puede conectar a las entradas de otras neuronas artificiales o perceptrones mediante conexiones ponderadas. Por ende, una red neuronal consiste en un conjunto de perceptrones conectados de una forma concreta.

Generalmente los perceptrones están organizados en grupos llamados niveles o capas. Una red normalmente consiste en una secuencia de capas con conexiones entre capas adyacentes consecutivas. Existen tres tipos de capas en una red neuronal; una capa de entrada donde se le suministran los datos a la red, una capa de salida donde mantiene los resultados de la red, y entre estas capas un conjunto de capas denominadas capas ocultas (Xabier Basogain, 2008). La figura 3 muestra el aspecto de una red neuronal artificial.

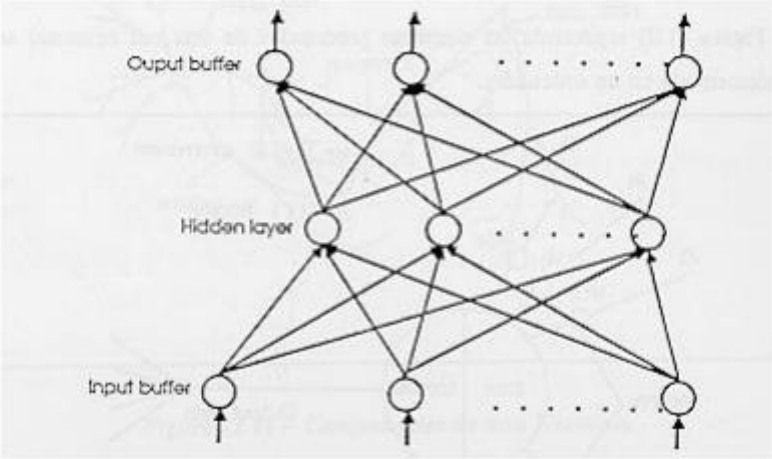

Figura 3. Diagrama de una Red Neuronal Artificial

4.8 Deep Learning

El deep learning es un tipo particular de machine learning que logra un gran poder y flexibilidad al aprender a representar el mundo como una jerarquía anidada de conceptos, con cada concepto definido en relación con conceptos más simples y

representaciones más abstractas calculadas en términos de conceptos menos abstractos (Goodfellow, Bengio, Courville, 2014).

El deep learning se basa en redes neuronales con una mayor cantidad de capas ocultas que las redes neuronales convencionales. Este tipo de topología le permite obtener patrones o características simples a partir de entradas complejas. Adicionalmente permite que cada capa se enfoque en resolver un problema específico.

El deep learning abarca diversas arquitecturas de redes neuronales, como lo son: las redes neuronales profundas, las redes neuronales convolucionales y las redes neuronales recurrentes.

Las redes neuronales profundas son simplemente una red neuronal tradicional con un mayor número de capas ocultas.

Las redes convolucionales, también conocidas como redes neuronales convolucionales o CNN, son un tipo especializado de red neuronal que procesa datos que tienen una topología similar a una cuadrícula. Los ejemplos incluyen datos de series temporales, que pueden considerarse como una cuadrícula 1D que toma muestras a intervalos de tiempo regulares, y datos de imágenes, que pueden considerarse como una cuadrícula de píxeles 2D. El nombre "red neuronal convolucional" indica que la red emplea una operación matemática llamada convolución. La convolución es un tipo especializado de operación lineal. Las redes convolucionales son simplemente redes neuronales que utilizan la convolución en lugar de la multiplicación general de la matriz en al menos una de sus capas. (Goodfellow, 2014)

Las redes neuronales recurrentes o RNN son un tipo de red neuronal que procesa datos secuenciales. Al igual que una red convolucional es una red neuronal especializada para procesar una cuadrícula de valores, como una imagen. Así como las redes convolucionales pueden escalar fácilmente a imágenes con gran ancho y alto, y algunas redes convolucionales pueden procesar imágenes de tamaño variable, las redes recurrentes pueden escalar a secuencias mucho más largas de lo que sería práctico para redes sin especialización basada en secuencias. La mayoría de las redes recurrentes también pueden procesar secuencias de longitud variable.
(Godfellow, 2014)

4.9 Red Neuronal Xception

Es una red neuronal convolucional compuesta enteramente por capas separables de convolución profunda, basada en la arquitectura de la red neuronal convolucional Inception. La arquitectura Xception tiene 36 capas convolucionales estructuradas en 14 módulos, todos los cuales se conectan con las otras por medio de conexiones residuales, excepto el primer y el último módulo. (Chollet, 2017).

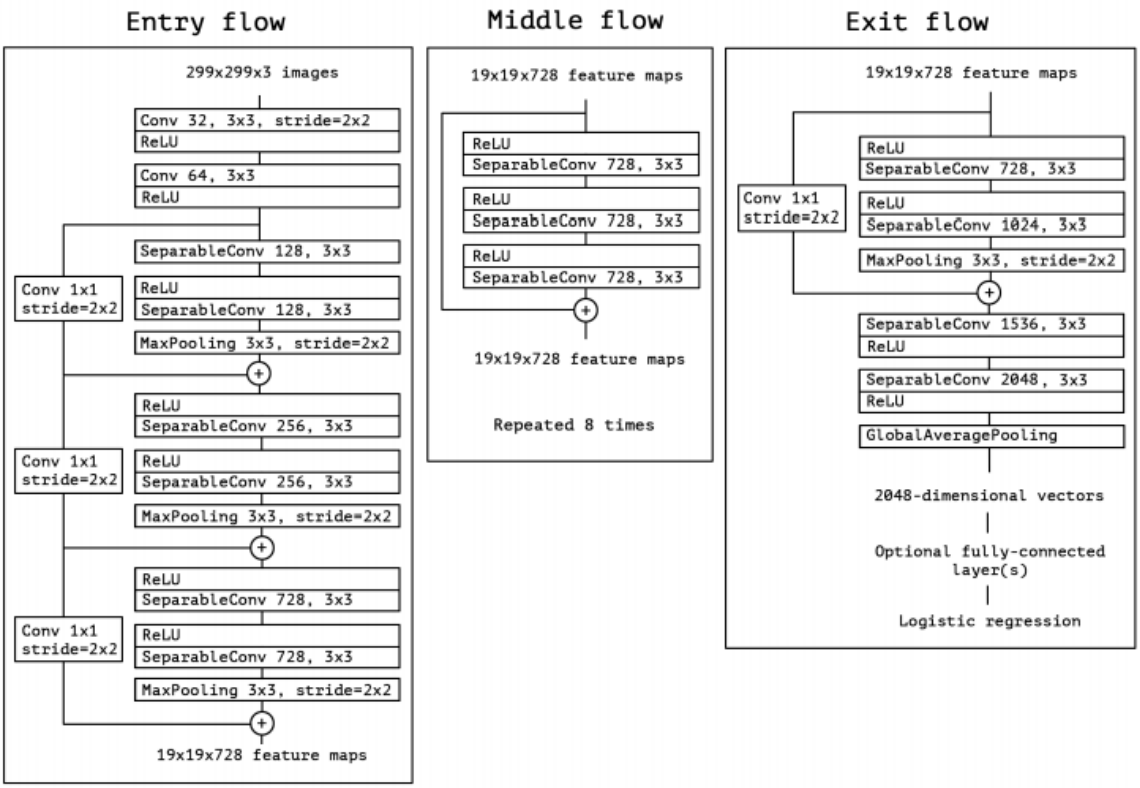

Figura 4. Esquema de la arquitectura Xception.

Lo anterior hace que esta arquitectura sea fácil de implementar y modificar, es de código abierto y está disponible en el módulo de aplicaciones de Keras.

4.10 Modelo generativo

Un modelo generativo describe como un conjunto de datos es generado en términos de un modelo probabilístico, de forma que por muestreo de este modelo es posible generar nuevos datos. (Foster, 2019).

Un modelo generativo requiere un conjunto de datos con muchos ejemplos de la entidad que se quiere generar, este conjunto son los datos de entrenamiento, a cada dato se le denomina observación y a su vez cada observación tiene características que van a permitir crear o generar nuevos datos.

El objetivo de un modelo generativo es imitar una distribución de datos de la forma más cercana posible, de manera que esos datos generados parezcan haber sido incluidos en el conjunto de entrenamiento original.

4.11 Generative Adversarial Network GAN

Es un tipo de sistema de machine learning basado en dos modelos que utilizan redes neuronales. Uno de estos es un modelo generativo que a partir de una distribución de datos genera datos similares y el otro es un modelo discriminatorio que aprende a

determinar si una muestra proviene de la distribución del modelo o de la distribución de datos. Es decir, el modelo generativo intenta crear contenido falso mientras que el modelo discriminatorio trata de detectar el contenido falso del modelo generativo. La competencia en este sistema impulsa a ambos modelos a mejorar sus resultados hasta que las falsificaciones sean indistinguibles de los artículos genuinos. (Goodfellow, 2014).

4.12 Ciberseguridad

La ciberseguridad es el conjunto de tecnologías, procesos, prácticas y medidas de respuesta y mitigación diseñadas para proteger redes, computadoras, programas y datos de ataques, daños y accesos no autorizados de manera que se garantice la confidencialidad, integridad y disponibilidad. (Seguridad pública de Canadá, 2014). Debido al desarrollo de cada vez más sofisticadas herramientas de edición que hacen uso de inteligencia artificial, es necesario desarrollar herramientas que permitan clarificar cuando un contenido está alterado. La ciberseguridad lucha contra el uso criminal o no autorizado de datos electrónicos, y desarrolla las medidas para lograrlo. (Oxford University Press, 2014).

> El software convencional de seguridad para la identificación de amenazas cibernéticas requiere de un esfuerzo humano que implica tiempo y recursos, la aplicación de la ciencia de datos es actualmente uno de los caminos más prometedores, porque a través de sus técnicas permite conseguir soluciones aproximadas a problemas complejos. (Urcuquí et al., 2018, p23)

4.13 Métodos generadores de Deepfake

Actualmente, existen métodos y herramientas para la generación de imágenes y vídeos no reales basados en inteligencia artificial, entre estos podemos encontrar desde herramientas de código abierto tales como Face2Face que permiten hacer Deepfake en vivo hasta aplicaciones de escritorios como Faceswap que permiten hacer un intercambio de caras entre dos vídeos. A continuación, una descripción de cada uno de los métodos y/o herramientas más conocidas y un cuadro comparativo entre ellas.

4.13.1 Faceswap

Es un programa para escritorio que usa una red neuronal (NN) para intercambiar un rostro original por un rostro de intercambio. Para hacer esto cuenta con:

- **Codificador compartido:** Que se entrena con dos conjuntos de rostros; conjunto A y B. Conjunto A (las caras originales que queremos reemplazar) y el conjunto B (las caras de intercambio que deseamos colocar en el original). Compartiendo el codificador se logra que este genere un algoritmo único para ambos conjuntos de rostros.

- **Decodificadores diferentes:** El decodificador A se encarga de reconstruir los rostros de conjunto A y de igual forma trabaja el decodificador B. Cuando se quiere hacer el intercambio de rostro (luego de entrenar al modelo con suficientes imágenes de cada conjunto), el decodificador A reconstruirá los rostros del conjunto B y el decodificador B reconstruirá los rostros del conjunto A. Dando como resultado una cara intercambiada. Lo anterior está esquematizado en la Figura 5.

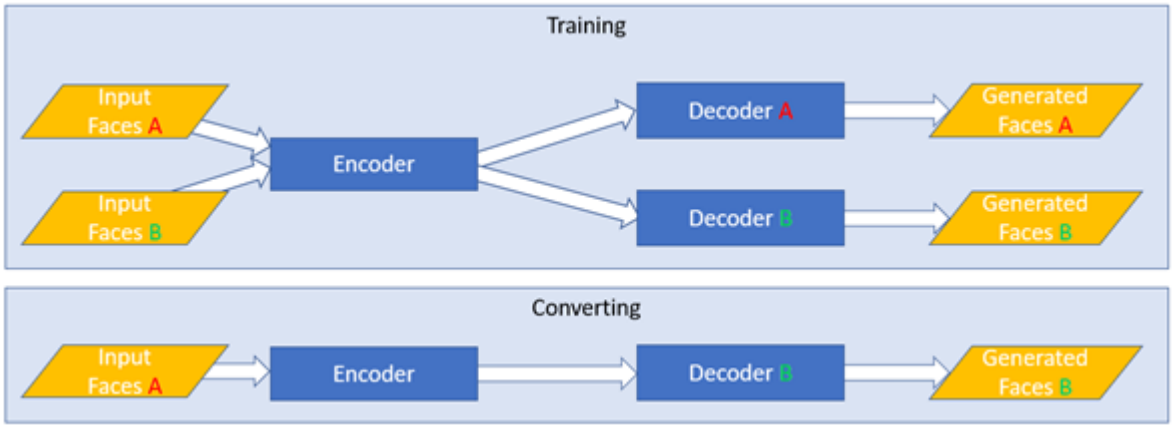

Figura 5. Esquema de funcionamiento de Faceswap.

Resulta importante mencionar que permite el uso de diferentes configuraciones de parámetros y modelo. El anteriormente descrito es el modelo original cuya entrada y salida son imágenes de rostros de 64 x 64 píxeles, sin embargo, existen modelos más sofisticados, que permiten entradas y producen salidas en diversas resoluciones.

4.13.2 Face2Face

Face2Face es una herramienta de código abierto que opera a través de dos entradas, en primer lugar, el sistema procesa una transmisión de vídeo en vivo a través de una cámara web del sujeto a reemplazar, en segundo lugar, un vídeo en el que este presente el rostro del objetivo (vídeo del sujeto que se quiere suplantar) utilizado. Las imágenes de los rostros se sintetizan usando un modelo multi-linear de rostros y un modelo de transformación rígida.

El anterior método permite encontrar parámetros de los rostros, que posteriormente se someterán a la aplicación de un método de minimización de energía variacional que permitirá optimizar los parámetros. Para minimizar esta energía se usa un solucionador de mínimos cuadrados iterativos reponderados iterativamente (Iteratively Reweighted Least Squares IRLS).

Lo anterior permite obtener la identidad facial entre otros datos relevantes de los actores fuente y objetivo. En ejecución las animaciones se reconstruyen usando un seguimiento a cuadro con una formulación energética similar a la anterior. Para la

recreación se usa una deformación rápida que opera en el estadístico usado anteriormente.

4.13.3 FakeAPP

Es una aplicación de escritorio que permite crear vídeos Deepfake. Se caracteriza por usar un modelo de red neuronal profunda (DNN). Las representaciones comprimidas o vectores latentes que los autoencoders convolucionales actuales aprenden, son la piedra angular detrás de las capacidades de intercambio de caras de esta aplicación (Yuezun Li, 2019).

4.13.4 FaceApp

Es una aplicación para dispositivos móviles. Usa una foto del rostro de una persona y posteriormente se le realiza la edición que se seleccione. La versión gratuita ofrece diferentes ediciones como, aplicar una sonrisa básica, añadir una barba básica, realizar una mezcla de fotos de rostro, aplicar un aumento de edad, cambiar el color del cabello, agregar accesorios, entre otras. La versión Pro ofrece herramientas de edición más potentes y al ser una aplicación comercial no es de código abierto.

4.13.5 Faceswap-GAN

Este método está basado en el uso de redes generativas antagónicas (GAN), específicamente en una variación denominada Red Adversaria Generativa de Auto-Atención SAGAN). Este tipo de red introduce un mecanismo de auto atención en las GAN convolucionales. El módulo de auto-atención es complementario a las convoluciones y ayuda a modelar dependencias de múltiples niveles a largo plazo en las regiones de la imagen. Armado con atención propia, el generador puede dibujar imágenes en las que los detalles finos en cada ubicación se coordinan cuidadosamente con los detalles finos en partes distantes de la imagen. (Han Zhang, 2019)

Visto de manera general la arquitectura del modelo de Faceswap-GAN se compone principalmente de 4 partes: un codificador, un decodificador, una red generativa y una red discriminadora (Shaoanlu, 2019).

Por otro lado, dentro de las características importantes de este método tenemos:

- La pérdida perceptual de cara VGG que mejora la dirección de los globos oculares permitiendo al Deepfake ser más realista y consistente con la cara de entrada.
- Una resolución de salida configurable. El modelo admite resoluciones de salida de 64x64, 128x128 y 256x256.

- Alineación de rostros usando MTCNN y filtro de Kalman en conversión de vídeo. Se introduce MTCNN para detecciones más estables y una alineación facial confiable.

4.13.6 Style-Based Generator Architecture GAN.

Este método está basado en una red generativa antagónica con una arquitectura basada en transferencia de estilos. La transferencia de estilos se basa en representar el contenido de una imagen en el estilo de otra. Esta arquitectura conduce a una separación automática y sin supervisión de los atributos de alto nivel (por ejemplo pose e identidad cuando se entrena en rostros humanos) y a la variación estocástica en las imágenes generadas (por ejemplo pecas, cabello) (Tero Karras, 2019).

El generador parte de una entrada constante aprendida y ajusta el "estilo" de la imagen en cada capa de convolución en función del código latente, por lo tanto, controla directamente la fuerza de las características de la imagen a diferentes escalas. Combinado con el ruido inyectado directamente en la red, este cambio arquitectónico es el que conduce a la separación automática y sin supervisión de los atributos de alto nivel de la variación estocástica en las imágenes generadas, permitiendo una mezcla específica a escala intuitiva, así como operaciones de interpolación.

Cabe resaltar que esta herramienta no solo sirve para generar Deepfakes sino que también permite “transferencia de estilo” entre diversos tipos de imágenes ya sea que estas contengan objetos, animales o cualquier tipo de representación de forma visual. El modelo bajo el cual se basa este método permite que haya un entrenamiento con cualquier tipo de resolución. Sin embargo, este método solo usa como ejemplo un modelo pre entrenado para imágenes de 1024 x 1024 píxeles.

Tabla 1. Comparación entre métodos generadores.

Método/ Característica	**Faceswap**	**Face2Face**	**FakeAPP**	**Faceswap-GAN**	**Style-Based Generator Architecture GAN.**	**FaceAPP**
Tipo de dato de entrada	Conjunto de imágenes en formato PNG y/o JPG	Transmisión de vídeo en vivo a través de una cámara web y vídeo monocular	Vídeo fuente y objetivo en cualquier formato de vídeo	Vídeo fuente y vídeo objetivo en formato MP4	Conjunto de imágenes en formato PNG	Imagen en formato JPG o PNG
Resolución de dato de entrada	Cualquier resolución	1280 x 720 (Vídeo monocular) 640 x 480 (Vídeo en vivo)	Cualquier resolución	Cualquier resolución	1024 x 1024 píxeles	Cualquier resolución

Tipo de dato de salida	Imagen en formato PNG o vídeo en MP4 que refleja el intercambio de un rostro	Vídeo en tiempo real del objetivo con los gestos de la fuente	Vídeo y conjunto de imágenes con el rostro del vídeo objetivo en el vídeo fuente	Vídeo en formato MP4 que refleja el intercambio de rostro	Imagen en formato PNG que refleja la transferencia de estilo	Imagen en formato PNG o JPG con una modificación facial escogida
Resolución de dato salida	La misma que la resolución del vídeo o imagen ingresada	1280 x 720 píxeles	La misma que el vídeo objetivo	64 x 64 128 x 128 256 x 256 pixeles	1024 x 1024 píxeles	La misma que la resolución de la imagen ingresada
Tipo de modelo	Red Neuronal	Consta de dos modelos: un modelo multi-linear de rostros y un modelo de transformación rígida	No disponible	Red Generativa Antagónica (GAN) + Auto encoder	Red Generativa Antagónica (GAN) + normalización de instancia adaptativa (AdaIN)	No disponible
Uso	Se encuentra integrado en una aplicación de escritorio	No aplica	Es en si una aplicación de escritorio	Se encuentra el código fuente organizado en Notebooks que permite realizar Deepfakes a través del navegador con el uso de la plataforma Colab de Google	Se encuentra el código fuente del modelo con instancias pre-entrenadas del modelo. Se puede usar para transferencia de estilos entre 2 imágenes	No aplica
Acceso a código fuente (Open Source)	Si	No	No	Si	Si	No

Se puede vislumbrar que existen diferentes métodos de generación de Deepfake que emplean diferentes técnicas y tecnologías, además reciben y producen diferentes tamaños y tipos de contenido visual (algunos vídeos otros imágenes). Resulta relevante destacar que las redes neuronales son ampliamente usadas en los generadores de Deepfake ya sean redes neuronales clásicas, profundas o GAN.

4.14 Error Level Analysis

El análisis de nivel de error (ELA por sus siglas en inglés) es un método forense de análisis de artefactos comprimidos; aprovecha los esquemas de compresión con pérdida de imágenes manipuladas para identificar su falsificación. El nivel de calidad original de una imagen es una característica única en sí misma, por lo tanto, cualquier

proceso de alteración deja sus huellas también en él. Brevemente, ELA funciona mediante el uso de una imagen comprimida por un esquema con pérdidas y recomprimiéndola con una tasa de error conocida, luego, calcula la diferencia absoluta entre la imagen analizada y la recomprimida.

Esta diferencia entre las imágenes son los niveles de error asociados con los píxeles originales, estos niveles de error, vistos como una cantidad de cambio, están directamente asociados con la pérdida de compresión. Si la cantidad de cambio es pequeña, el píxel ha alcanzado sus mínimos locales de error a la tasa de error determinada. Sin embargo, si hay una gran cantidad de cambio, entonces los píxeles no están en sus mínimos locales y es probable que se hayan insertado artificialmente a la imagen (Krawetz, 2007).

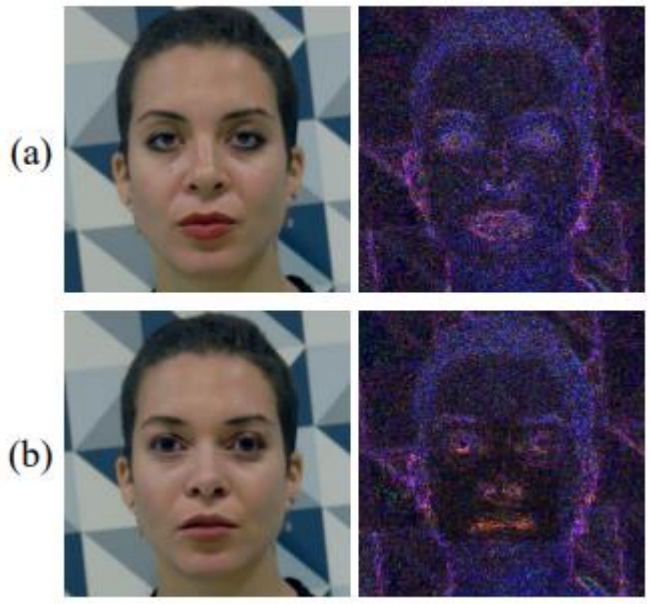

Figura 6. Método forense ELA.
(a) Imagen real, (b) imagen alterada. Columna derecha ELA.

5 ESTADO DEL ARTE

Con el fin de exponer el estado del arte del proyecto, se mostrarán 5 trabajos previos que constan de propuestas similares enfocadas en la detección de Deepfake.

5.1 Two-Stream Neural Networks for Tampered Face Detection

En este proyecto se propone una red de dos flujos para la detección de alteraciones faciales. Entrenan una red mediante el servicio de Google GoogLeNet para detectar artefactos de manipulación en una secuencia de clasificación de caras, se entrena esta red con el fin de aprovechar las características que capturan los residuos de ruido local y las características de la cámara como una segunda secuencia. Además, utilizan dos aplicaciones diferentes de intercambio de caras para crear un nuevo conjunto de datos que consta de imágenes manipuladas. Este método no puede detectar caras manipuladas muy pequeñas, esto se debe a que la secuencia de clasificación de caras necesita cambiar el tamaño de la cara de entrada a 299 × 299, y en el muestreo de caras pequeñas se pierde información visual crucial para la detección de alteraciones.

5.2 FWA: Exposing Deepfake Vídeos By Detecting Face Warping Artifacts.

En este trabajo, se describe un método basado en Deep Learning que puede distinguir los vídeos falsos generados por IA de los vídeos reales. Este método se basa en que los algoritmos de Deepfake deben realizar unas transformaciones que dejan elementos distintivos en los vídeos resultantes, y muestran que estos pueden ser capturados por redes neuronales convolucionales (CNN). Tiene la ventaja que en comparación con otros métodos no necesita imágenes generadas por Deepfake para el entrenamiento.

5.3 Mesonet: a compact facial vídeo forgery detection network.

Este método busca detectar de manera automática y eficiente la manipulación de la cara en vídeos. Sigue un enfoque de aprendizaje profundo y presenta dos redes, ambas con un bajo número de capas, para enfocarse en las propiedades mesoscópicas de las imágenes. Se evalúan esas redes tanto en un conjunto de datos existente como en un conjunto de datos que ellos han constituido a partir de vídeos en línea. Las pruebas demuestran una tasa de detección muy exitosa con más del 98% para Deepfake y 95% para Face2Face para tamaños de imágenes 256 x 256 píxeles.

5.4 HeadPose: Exposing deep fakes using inconsistent head poses.

Este método busca detectar imágenes o vídeos de caras falsas generadas por Deepfake. Se basa en las observaciones de que el contenido alterado se genera al

empalmar la región de la cara sintetizada en la imagen original y, al hacerlo, se introducen errores que pueden revelarse cuando se estiman las posturas de la cabeza en 3D a partir de las imágenes de la cara. Se realizan experimentos para demostrar este fenómeno y usando características basadas en este indicio, se evalúa un clasificador SVM (Máquinas de Vectores de Soporte) usando un conjunto de imágenes de caras reales y falsificadas.

5.5 Multi-task Learning For Detecting and Segmenting Manipulated Facial Images and Vídeos

En este proyecto diseñan una red neuronal convolucional que utiliza el enfoque de aprendizaje de tareas múltiples para detectar simultáneamente imágenes y vídeos manipulados y ubicar las regiones manipuladas. La información obtenida al realizar una tarea se comparte con la otra tarea y, por lo tanto, mejora el rendimiento de ambas tareas. Se utiliza un enfoque de aprendizaje semi-supervisado. La red incluye un codificador y un decodificador en forma de Y. La activación de las características codificadas se utiliza para la clasificación binaria. La salida de una rama del decodificador se usa para segmentar las regiones manipuladas, mientras que la de la otra rama se usa para reconstruir la entrada, lo que ayuda a mejorar el rendimiento general.

En la Tabla 2, se realiza un resumen de lo que se encuentra en el estado del arte. Las columnas representan los trabajos discutidos y en la última (izquierda a derecha) se incluye el trabajo realizado en este proyecto, por último, las filas representan los criterios de comparación entre cada uno de los trabajos.

Tabla 2. Tabla comparativa de métodos de detección.

Método/ características	**Two-Stream**	**FWA**	**Mesonet-4**	**HeadPose**	**Multitask**	**Proyecto de Grado**
Tecnología usada	Red neuronal profunda GoogLeNet	Red neuronal convolucional (ResNet50)	Red neuronal convolucional	Modelos matemáticos para estimar la postura de la cabeza en 3D a partir de imágenes de la cara	Red neuronal convolucional	Red Neuronal Convolucional Xception + Imagenet
Número de imágenes usadas en el entrenamiento	705 alterada + 1.400 reales	15.185 alteradas + 15.185 reales	5.111 alteradas + 7.250 reales	10.847 alteradas + 10.847 reales	218.179 alteradas + 218.179 reales	4.073 alteradas + 2.597 reales

Conjunto de datos de entrenamiento	Datos generados con SwapMe	Datos recopilados por los autores en distintas plataformas de vídeo	Datos recopilados por los autores en distintas plataformas de vídeo	Conjunto de datos UADFV	Conjunto de datos FF-DF	Conjunto de datos FF-DF, DFD, DFDC, Celeb-DF
Conjunto de datos público	No	Si	No	Si	Si	Si
AUC mayor a 80% sobre el conjunto de datos de PRIMERA generación.	No	Si	No	No	No	Si
AUC mayor a 80% sobre el conjunto de datos de SEGUNDA generación.	No	No	No	No	No	Si
Acceso a código fuente (Open Source)	No	Si	Si	Si	Si	Si

6 METODOLOGÍA

Para la metodología del proyecto, se siguen los lineamientos de CRISP-DM. Dado que esta metodología está probada y es ampliamente utilizada en proyectos que requieren análisis y procesamiento de datos.

CRISP-DM de las siglas Cross-Industry Standard Process for Data Mining es una metodología de análisis o minería de datos especializada en el manejo de grandes cantidades de información o big-data. Esta metodología permite describir las fases del proyecto, las tareas a realizar en cada fase y las relaciones entre las tareas.

La metodología CRISP-DM (Figura 9) consiste en un proceso iterativo de seis fases: Comprensión del negocio o problema, Compresión de los Datos, Preparación de los Datos, Modelado, Evaluación e Implantación, esta última fase no aplica para este proyecto dado que el propósito de este es evaluar un modelo viable para la detección de Deepfake.

6.1 Descripción de fases

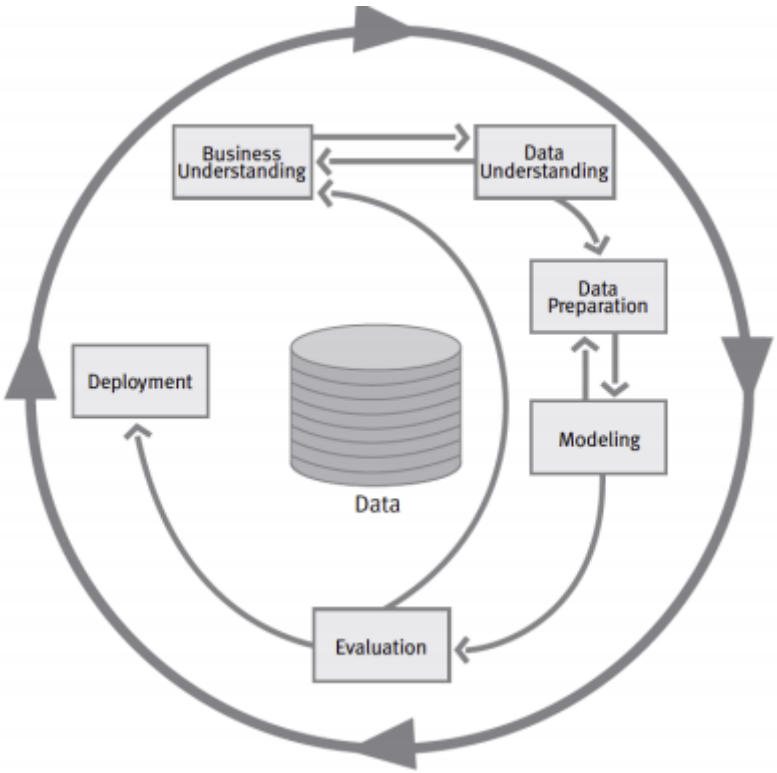

Figura 7. Diagrama de ciclo de vida CRISP-DM.

6.1.1 Comprensión del negocio

En esta primera fase se hizo una comprensión del problema, esto mediante el estudio de diferentes métodos generadores y detectores de Deepfake que existen

actualmente (visto en las secciones 4.13 y 5). Con el objetivo de entender cómo se puede generar Deepfake se usó la herramienta Faceswap.

Se generaron 3 pares de vídeos con rostros intercambiados. Para generar el primer par de Deepfakes, se grabaron dos vídeos que contenían los rostros de los dos sujetos (sujeto A y sujeto B) a los que se querían intercambiar los rostros. El vídeo del sujeto A duraba 3 minutos 31 segundos y el de sujeto B duraba 5 minutos 41 segundos. Usando la funcionalidad “Extract” de Faceswap se obtuvieron todas las imágenes en las que el algoritmo detectaba un rostro y usando la funcionalidad “Sort” se ordenaron las imágenes con base en su contenido, esto con el propósito de remover las imágenes que no tuvieran un rostro (por fallas en la detección del algoritmo) o que tuvieran rostros borrosos. Una vez se tuvieron los rostros a usar para el intercambio se procedió a usar la funcionalidad “Train” que permite entrenar el modelo que aprenderá a intercambiar los rostros, se seleccionó el modelo `Dfl-H128` y un `Batch_size=64.` El modelo realizó 232162 iteraciones obteniendo una pérdida en el rostro A de 0.019 y en el rostro B de 0.022. Finalmente usando la herramienta “Convert” se generan las imágenes que componen el vídeo del sujeto B con el rostro del sujeto A usando la configuración predeterminada y usando la funcionalidad “Effmpg” se unen las imágenes del paso anterior para producir el vídeo resultado. Lo mismo para el vídeo del sujeto A con el rostro del sujeto B, seleccionando la opción “Swap Model” en la funcionalidad “Convert”.

Para los Deepfakes del segundo par de vídeos se usaron grabaciones con duraciones de 6 minutos 35 segundos y 4 minutos 7 segundos, el procedimiento fue el mismo que para el primer par de vídeos y el modelo realizó 78379 iteraciones obteniendo una pérdida en el rostro A de 0.025 y en el rostro B de 0.024.

Para los Deepfakes del tercer par de vídeos se usaron grabaciones con duraciones de 56 segundos y 1 minuto 14 segundos, el procedimiento fue el mismo que para el primer par de vídeos sin embargo el modelo escogido fue el `Dfaker`. El modelo realizó 95601 iteraciones obteniendo una pérdida en el rostro A de 0.025 y en el rostro B de 0.025.

Figura 8. Tercer par de vídeos usados para generación de Deepfake. Columna izquierda rostros originales. Columna derecha intercambiados.

6.1.2 Fase de comprensión de los datos

En esta fase se hizo una caracterización de cinco conjuntos de datos más relevantes desde el punto de vista académico en el estudio de los Deepfakes, se hizo una descripción de cada uno de estos conjuntos teniendo en cuenta elementos como la cantidad, la resolución y la variabilidad de los vídeos que los componen, así como otros atributos. A continuación, se presenta cada uno de los conjuntos de datos con su respectiva descripción:

6.1.2.1 DF-TIMIT

El Deepfake-TIMIT dataset incluye 640 vídeos Deepfake generados con Faceswap-GAN basado en el conjunto de datos Vid-TIMIT. Los vídeos se dividen en dos subconjuntos de igual tamaño: DF-TIMIT-LQ y DF-TIMIT-HQ, con caras sintetizadas de 64 × 64 y 128 × 128 píxeles, respectivamente.

Seleccionaron manualmente 16 pares de personas de aspecto similar de la base de datos DF-TIMIT. Para cada 32 sujetos, entrenaron dos modelos diferentes, en el documento, conocidos como modelos de baja calidad (LQ), con 64 × 64 de entrada / salida, y de alta calidad (HQ), con tamaño de 128 × 128. Dado que hay 10 vídeos por persona en la base de datos DF-TIMIT, generaron 320 vídeos correspondientes a cada versión, lo que resulta en un total de 640 vídeos con caras intercambiadas. No se realizó ninguna manipulación en el canal de audio. Este conjunto de datos fue lanzado en diciembre de 2018. (Korshunov & Marcel, 2018)

6.1.2.2 FF-DF

FaceForensics ++ es un conjunto de datos que consta de 1000 secuencias de vídeo originales que han sido manipuladas con cuatro métodos automatizados de manipulación facial: Deepfakes, Face2Face, Faceswap y NeuralTextures. Los datos provienen de 977 vídeos de youtube y todos los vídeos contienen una cara frontal rastreable, lo que permite que los métodos automáticos de manipulación generen falsificaciones realistas. (Rossler et al., 2019)

La distribución de los datos respecto al género, la resolución y el cubrimiento de pixeles en las caras es como sigue:

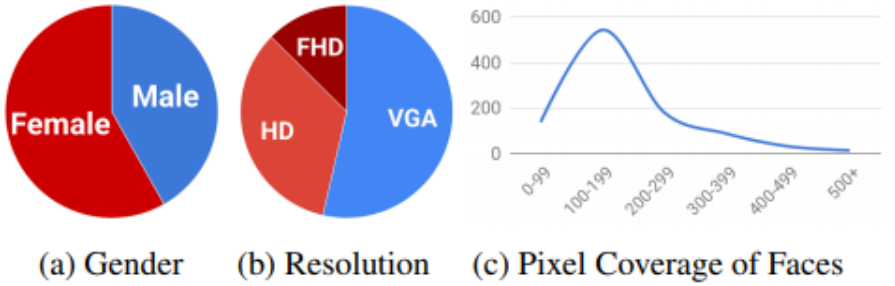

Figura 9. Composición del conjunto de datos FF-DF. En el gráfico (b) VGA denota 480p, HD denota 720p, y FHD denota resolución 1080p. El gráfico (c) muestra el número de secuencias (eje y) con una altura de píxel del cuadro delimitador dado (eje x).

Este conjunto de datos fue lanzado en enero de 2019.

6.1.2.3 DFD

El Google/Jigsaw Deepfake Detection Dataset tiene 3068 vídeos Deepfake basados en 363 vídeos originales de 28 individuos de varias edades, géneros y grupos étnicos. No se revela mayor detalle del algoritmo de generación usado, pero si establecen que es una mejora a un algoritmo de generación de Deepfake básico. Este conjunto de datos fue lanzado en septiembre de 2019. (Li, et al., 2019)

6.1.2.4 DFDC

Los vídeos de este conjunto de datos incluyen condiciones de iluminación y ángulos del rostro variados. Los creadores de este conjunto de datos manifiestan que los participantes pudieron grabar sus vídeos con el fondo que desearan, lo que produjo fondos visualmente diversos. La aproximación de la distribución general de género y raza en este conjunto de datos es 74% femenino y 26% masculino; y 68% caucásicos, 20% afroamericanos, 9% del este asiático y 3% del sur asiático. Este conjunto de datos consta de 66 individuos. Este conjunto de datos fue creado usando dos diferentes algoritmos de generación, sin embargo, no se especifican cuáles son. DFDC fue lanzado en octubre de 2019. Finalmente, este conjunto de datos tiene un total de 4,464 clips de entrenamiento y 780 clips de prueba. (Dolhansky, et al., 2019)

6.1.2.5 CELEB-DF

El conjunto de datos Celeb-DF consta de 590 vídeos reales y 5,639 vídeos Deepfake (correspondientes a más de dos millones de cuadros de vídeo). La duración promedio de todos los vídeos es de aproximadamente 13 segundos con una velocidad de cuadro estándar de 30 cuadros por segundo. Los vídeos reales se eligen de vídeos de YouTube disponibles al público, correspondientes a entrevistas de 59 celebridades con una distribución diversa en sus géneros, edades y grupos étnicos. El 56.8% de los sujetos en los vídeos reales son hombres y el 43.2% son mujeres. El 8,5% tiene 60 años o más, el 30,5% tiene entre 50 y 60 años, el 26,6% tiene 40 años, el 28,0% tiene 30 años y el 6,4% tiene menos de 30 años. El 5,1% son asiáticos, el 6,8% son afroamericanos y el 88,1% son caucásicos. Además, los vídeos reales exhiben una gran variedad de cambios en aspectos tales como los tamaños de cara de los sujetos (en píxeles), orientaciones, condiciones de iluminación y fondos. Los vídeos de Deepfake se generan intercambiando caras para cada par de los 59 sujetos. Los vídeos finales están en formato MPEG4.0 y no cuentan con sonido. Celeb-DF fue lanzado en noviembre de 2019. (Li, et al., 2019)

Al terminar esta fase de comprensión de los datos se decidió hacer énfasis en la detección de Deepfakes en el conjunto de vídeos de Celeb-DF, dado que es el conjunto de datos que más dificulta la detección de contenido en los métodos detectores actuales. (Li, et al., 2019)

6.1.3 Fase de preparación de los datos:

Para los diferentes experimentos que se llevaron a cabo en la fase de modelamiento y evaluación, se realizó previamente una selección de los conjuntos de imágenes-vídeos y un preprocesamiento de los datos, las actividades fueron:

1. Se tomaron **1052 vídeos falsos** (modificados con Deepfake) y **890 vídeos reales** de **Celeb-DF**.

2. Para los **experimentos 1 y 4** se extrajeron de los vídeos mencionados en el **numeral 1** un total de 7704 **imágenes completas** (frames) falsas y 7380 **imágenes completas** reales.

3. Para los **experimentos 2 y 5** se extrajeron de los vídeos mencionados en el **numeral 1** un total de 7704 **rostros** falsos y 7380 **rostros** reales haciendo uso de la librería OpenCV.

4. Para los **experimentos 3 y 6** se aplicó el método **ELA** con un **ratio de error del 90%** a los rostros extraídos en el **numeral 3**.

5. Se tomaron **320 vídeos reales** de los conjuntos de datos **Celeb-DF, DFDC**, **FF-DF/raw**, **FF-DF/C23**, **FF-DF/C40**, **DFC/raw**, **DFC/C23** y **DFC/C40** donde C23 y C40 corresponden a los grados de compresión de los vídeos. También se tomaron 320 **vídeos modificados** con Deepfake de los conjuntos de datos anteriormente mencionados junto con otros **320 vídeos de DF-TIMIT-LQ** y **320 de DF-TIMIT-HQ**. Se extrajeron imágenes (frames) de estos vídeos y se obtuvieron **4073 imágenes falsas** y **2597 imágenes reales.**

6. Para el **experimento 7** se extrajo el rostro a cada una de las imágenes extraídas en el numeral 5 haciendo uso de la librería **OpenCV**.

7. Para el **experimento 8** se extrajo el rostro a cada una de las imágenes extraídas en el numeral 5 haciendo uso de la librería **OpenCV** y se le aplicó el método **ELA** con un **ratio de error del 90%**.

6.1.4 Fase de modelado

En lo referente a esta fase se entrenó una red convolucional Xception pre entrenada con Imagenet a la cual se le agregó una capa de clasificación compuesta por un GlobalAveragePooling2D, una Relu y un Softmax (Otenim, 2019). Esta arquitectura de red acepta imágenes de 299 x 299 pixeles. Se efectuaron 8 experimentos en donde se hicieron variaciones de los datos de entrenamiento y evaluación de la siguiente manera:

1. En los experimentos del 1 al 6 se tomaron 7024 imágenes falsas y 7024 imágenes reales para entrenamiento y se hizo la evaluación con 680 imágenes falsas y 356 imágenes reales. Es importante aclarar que las imágenes usadas para evaluación fueron extraídas del conjunto de vídeos definidos por Celeb para evaluación. Lo anterior con el fin de poder comparar los resultados obtenidos con otras propuestas de detección.

2. En los experimentos 7 y 8 se tomaron 2624 imágenes falsas y 1686 imágenes reales para entrenamiento y se hizo la evaluación con 1449 imágenes falsas y 911 imágenes reales.

6.1.5 Fase de evaluación

Por último, se escogió el mejor modelo teniendo en cuenta las métricas de accuracy (por su fácil interpretación) y de AUC (que permite hacer una comparación con otros métodos de detección existentes) resultante de evaluar con los vídeos de evaluación del conjunto de datos Celeb-DF.

También se tendrá en cuenta el accuracy y AUC resultantes de evaluar los vídeos que se generaron en la fase de comprensión del negocio (3 pares de vídeos).

En lo referente a la metodología en general solo se hizo necesario una iteración sobre el proceso, dado que en el momento en que se necesitaban cambios se regresó a etapas previas buscando completar tareas o corregir errores.

7 RESULTADOS Y EXPERIMENTOS

7.1 Tecnologías empleadas

Durante el proyecto se utilizó como lenguaje de programación Python 3 junto con Jupyter Notebook y las librerías de scikit-learn, numpy, pandas, dlib, OpenCV, os, pylab, PIL y Keras.

7.2 Experimento de clasificación de datos

Como parte del proceso de clasificación de datos en la fase de comprensión de los mismos, se planteó la hipótesis de que las métricas de evaluación de calidad para un vídeo original respecto a sus compresiones podían servir para comparar clips de vídeo diferentes usando un vídeo como referencia. Las métricas evaluadas fueron la VMAF desarrollada por Netflix que busca emular la percepción humana para determinar la calidad y PSNR que mide el nivel de distorsión midiendo el error cuadrático medio entre la señal original y la distorsionada.

Para evaluar esta hipótesis se usó el conjunto de datos “The Facebook Deepfake Detection Challenge (DFDC)” (Dolhansky, Brian, et al., 2019). Inicialmente tomamos seis vídeos del conjunto de datos (1920 x 1080); 2 de calidad baja, 2 de calidad media y 2 de calidad alta esto basados en una métrica subjetiva según la percepción de los tres autores de este proyecto, usamos el software libre “ffmpeg-quality-metrics” (Werner Robitza, 2020) para obtener las métricas VMAF y PSNR; los resultados de ambas métricas resultaron coherentes a lo clasificado sin embargo eran muy pocos datos. Se realizó la misma tarea con 50 vídeos (usando siempre un vídeo como referencia) se verificaron con percepción humana, obteniendo un 42% de accuracy.

Lo anterior permitió concluir que las métricas de evaluación de calidad para un vídeo original respecto a sus compresiones no funcionan para comparar la calidad de clips de vídeos diferentes y por lo tanto no es útil para clasificar el contenido visual según la calidad. Al no encontrar un método que permitiera clasificar contenido visual evaluando la calidad de forma algorítmica, se decidió clasificar el contenido como conjunto de datos de primera generación y de segunda generación. Donde los de primera generación “incluyen caras sintetizadas de baja calidad, límites de empalme visibles, falta de coincidencia de colores, partes visibles de la cara original, y orientaciones faciales sintetizadas inconsistentes, y los de segunda generación mejoran la cantidad y los problemas de calidad que presentan los conjuntos de datos de primera generación de Deepfakes” (Li, Yang, Sun, Qi, & Lyu, 2019). Los conjuntos de datos de primera generación empleados en este proyecto son: **DF-TIMIT, FF-DF** y los de segunda generación fueron **DFD, DFDC, Celeb-DF**.

7.3 Experimento Xception # 1

En la fase de modelado, se entrenó el modelo con los datos especificados en la fase de preparación de datos. Se configuró una red **Xception** pre entrenada con

ImageNet, con `learning_rate=1e-4`, `epoch=50` y `batch_size=16`. Además, se fijaron los pesos de toda la red exceptuando la última capa (clasificador).

En la fase de evaluación se utilizó el conjunto de datos de evaluación provisto por Celeb-DF lo que dio como resultado un **accuracy** de **40,34%** y un **AUC** del **52,75%** y se obtuvo la matriz de confusión que se presenta en la Tabla 3.

Tabla 3. Matriz de confusión del experimento 1.

Experimento 1		**Reales**	
		Real	**Fake**
Predichos	**Real**	329	591
	Fake	27	89

7.4 Experimento Xception # 2

En la fase de modelado, se entrenó el modelo con los datos especificados en la fase de preparación de datos. Se configuró una red **Xception** pre entrenada con ImageNet, con `learning_rate=1e-4`, `epoch=50` y `batch_size=16`. Además, se fijaron los pesos de toda la red exceptuando la última capa (clasificador).

En la fase de evaluación se utilizó el conjunto de datos de evaluación provisto por Celeb-DF lo que dio como resultado un **accuracy** de **55,86%** y un **AUC** del **62,26%** y se obtuvo la matriz de confusión que se presenta en la Tabla 4.

Tabla 4: Matriz de confusión experimento 2.

Experimento 2		**Reales**	
		Real	**Fake**
Predichos	**Real**	126	161
	Fake	230	519

7.5 Experimento Xception # 3

En la fase de modelado, se entrenó el modelo con los datos especificados en la fase de preparación de datos. Se configuró una red **Xception** pre entrenada con ImageNet, con `learning_rate=1e-4`, `epoch=50` y `batch_size=16`. Además, se fijaron los pesos de toda la red exceptuando la última capa (clasificador).

En la fase de evaluación se utilizó el conjunto de datos de evaluación provisto por Celeb-DF lo que dio como resultado un **accuracy** de **49,03%** y un **AUC** del **53,68%** y se obtuvo la matriz de confusión que se presenta en la Tabla 5.

Tabla 5. Matriz de confusión experimento 3

Experimento 3		Reales	
		Real	Fake
Predichos	Real	244	416
	Fake	112	264

7.6 Experimento Xception # 4

En la fase de modelado, se entrenó el modelo con los datos especificados en la fase de preparación de datos. Se configuró una red **Xception** pre entrenada con ImageNet, con `learning_rate=1e-4`, `epoch=50` y `batch_size=16.`

En la fase de evaluación se utilizó el conjunto de datos de evaluación provisto por Celeb-DF lo que dio como resultado un **accuracy** de **54,15%** y un **AUC** del **63,53%** y se obtuvo la matriz de confusión que se presenta en la Tabla 6.

Tabla 6. Matriz de confusión experimento 4

Experimento 4		Reales	
		Real	Fake
Predichos	Real	333	452
	Fake	23	228

7.7 Experimento Xception # 5

En la fase de modelado, se entrenó el modelo con los datos especificados en la fase de preparación de datos. Se configuró una red **Xception** pre entrenada con ImageNet, con `learning_rate=1e-4`, `epoch=50` y `batch_size=16`.

En la fase de evaluación se utilizó el conjunto de datos de evaluación provisto por Celeb-DF lo que dio como resultado un **accuracy** de **81,27%** y un **AUC** del **83,99%** y se obtuvo la matriz de confusión que se presenta en la Tabla 7.

Tabla 7. Matriz de confusión experimento 5

Experimento 5		Reales	
		Real	Fake
Predichos	Real	330	168
	Fake	26	512

7.8 Experimento Xception # 6

En la fase de modelado, se entrenó el modelo con los datos especificados en la fase de preparación de datos. Se configuró una red **Xception** pre entrenada con ImageNet, con `learning_rate=1e-4`, `epoch=50` y `batch_size=16`.

En la fase de evaluación se utilizó el conjunto de datos de evaluación provisto por Celeb-DF lo que dio como resultado un **accuracy** de **60,91%** y un **AUC** del **65,74%** y se obtuvo la matriz de confusión que se presenta en la Tabla 8.

Tabla 8. *Matriz de confusión del experimento 6*

Experimento 6		Reales	
		Real	Fake
Predichos	Real	289	338
	Fake	67	342

7.9 Experimento Xception # 7

En la fase de modelado, se entrenó el modelo con los datos especificados en la fase de preparación de datos. Se configuró una red Xception pre entrenada con ImageNet, con `learning_rate=1e-4,` `epoch=50` y `batch_size=16.`

En la fase de evaluación se utilizó el conjunto de datos de evaluación que incluye los 5 conjuntos de datos seleccionados para este proyecto lo que dio como resultado un **accuracy** de **92,12%** y un **AUC** del **92,15%** y se obtuvo la matriz de confusión que se presenta en la Tabla 9.

Tabla 9. Matriz de confusión del experimento 7.

Experimento 7		Reales	
		Real	Fake
Predichos	Real	841	70
	Fake	116	1333

7.10 Experimento Xception # 8

En la fase de modelado, se entrenó el modelo con los datos especificados en la fase de preparación de datos. Se configuró una red **Xception** pre entrenada con ImageNet, con `learning_rate=1e-4`, `epoch=50` y `batch_size=16`.

En la fase de evaluación se utilizó el conjunto de datos de evaluación que incluye los 5 conjuntos de datos seleccionados para este proyecto lo que dio como resultado un **accuracy** de **60,90%** y un **AUC** del **75,95%** y se obtuvo la matriz de confusión que se presenta en la Tabla 10.

Tabla 10. Matriz de confusión del experimento 8.

Experimento 8		Reales	
		Real	**Fake**
Predichos	**Real**	684	336
	Fake	227	1113

A continuación, se muestra una matriz que resume los resultados y las características de los métodos evaluados en cada experimento.

Tabla 11. Comparación de experimentos

Experimento / Características	Descripción	Cantidad imágenes entrenamiento	Cantidad imágenes evaluación	Kappa	Accuracy	AUC
Experimento #1	Imágenes completas, Xception entrenamiento del Clasificador	7024 falsas, 7024 reales de Celeb-DF	680 falsas, 356 reales de Celeb-DF	0.04	40,34%	52,75%
Experimento #2	Rostros, Xception entrenamiento del Clasificador	7024 falsas, 7024 reales de Celeb-DF	680 falsas, 356 reales de Celeb-DF	0.12	55,86%	62,26%
Experimento #3	Rostros + ELA, Xception entrenamiento del Clasificador	7024 falsas, 7024 reales de Celeb-DF	680 falsas, 356 reales de Celeb-DF	0.06	49,03%	53,68%
Experimento #4	Imágenes completas, Xception entrenamiento completo	7024 falsas, 7024 reales de Celeb-DF	680 falsas, 356 reales de Celeb-DF	0.21	54,15%	63,53%
Experimento #5	Rostros, Xception entrenamiento completo	7024 falsas, 7024 reales de Celeb-DF	680 falsas, 356 reales de Celeb-DF	**0.62**	**81,27%**	**83,99%**
Experimento #6	Rostros + ELA, Xception entrenamiento completo	7024 falsas, 7024 reales de Celeb-DF	680 falsas, 356 reales de Celeb-DF	0.26	60,91%	65,74%
Experimento #7	Rostros, Xception entrenamiento completo	2624 falsas, 1686 reales de Celeb-DF, DFDC, FF-DF, DFC, DF-TIMIT	1449 falsas, 911 reales de Celeb-DF, DFDC, FF-DF, DFC, DF-TIMIT	**0.84**	**92,12%**	**92,15%**
Experimento #8	Rostros + ELA, Xception entrenamiento completo	2624 falsas, 1686 reales de Celeb-DF, DFDC, FF-DF, DFC, DF-TIMIT	1449 falsas, 911 reales de Celeb-DF, DFDC, FF-DF, DFC, DF-TIMIT	0.51	60,90%	75,95%

Figura 10. Gráfico comparativo AUC. Valores porcentuales.

7.11 Evaluación del modelo resultante del experimento # 7 con el conjunto de datos propio.

Al evaluar el modelo resultante del experimento 7, con 914 imágenes reales y 914 alteradas de los rostros de los videos usados para la generación de Deepfake en la fase de comprensión del negocio se obtuvo un **accuracy** de **94,87%** y un **AUC** del **94,87%.** La matriz de confusión que se presenta en la Tabla 12.

Tabla 12. Matriz de confusión. Evaluación modelo exp. 7 conjunto de datos propio

Evaluación modelo exp. 7 conjunto de datos propio.		**Reales**	
		Real	**Fake**
Predichos	**Real**	953	79
	Fake	21	895

8 ANÁLISIS DE RESULTADOS

De acuerdo con los experimentos realizados se encontró lo siguiente:

- Comparando los experimentos **1 y 2** en donde se entrena y se evalúa con la misma cantidad imágenes, y se usa el mismo modelo (Xception con entrenamiento solo en el clasificador) se obtiene un accuracy del **55,86%** en el experimento que es entrenado con rostros (experimento **2**) lo cual es significativamente mayor al **40,34%** de accuracy del experimento que se entrena con imágenes completas (experimento **1**). Lo mismo sucede con los experimentos **4 y 5** en donde el experimento **5** tiene un accuracy considerablemente alto de **81,27%** frente el experimento **4** de **54,15%.**

- Comparando los experimentos **2 y 3** en donde se entrena y se evalúa con las mismas imágenes, y se usa el mismo modelo (Xception con entrenamiento solo en el clasificador) se obtiene un accuracy del **55,86%** en el experimento que no hace uso de **ELA** (experimento **2**) lo cual es ligeramente mayor al **49,03%** de accuracy del experimento que hace uso de **ELA** (experimento **3**). Lo mismo sucede con los experimentos **5 y 6** en donde el experimento **5** tiene un accuracy considerablemente alto de **81,27%** frente el experimento **6** de **60,91%.** Esto también se presenta en los experimentos **7 y 8** en donde el experimento **7** obtiene un accuracy de **92,12%** lo cual es considerablemente mayor al accuracy de **60,90%** que da el experimento **8**.

- Contrastando los experimentos **1 y 4** en donde se entrena y se evalúa con las mismas imágenes, se obtiene un accuracy del **54,15%** en el experimento en el cual no se fijan los pesos de la red (experimento **4**) lo cual es considerablemente mayor al **40,34%** de accuracy del experimento que fija los pesos de la red entrenando solo el clasificador (experimento **1**). Lo mismo sucede con los experimentos **2 y 5** en donde el experimento **5** tiene un accuracy considerablemente alto de **81,27%** frente el experimento **2** de **55,86%.** Esto también se presenta en los experimentos **3 y 6** en donde el experimento **6** obtiene un accuracy de **60,91%** lo cual es considerablemente mayor al accuracy de **49,03%** que da el experimento **3**.

- El experimento **7** que se entrenó y se evaluó con todos los conjuntos de datos seleccionados para este proyecto, obtuvo un accuracy del **92,12%**, el mayor de todos los experimentos realizados. Al calcular este valor para cada conjunto de datos individualmente tenemos que para **DF-TIMIT** se obtuvo un accuracy de **100%**, para **FF-DF** de **96,62%**, para **Celeb-DF** de **95,25%**, para **DFDC** de **85,9%** y finalmente para **DeepFakeDetection** del **81%**.

- Al evaluar el modelo resultante del experimento 7 con el conjunto de datos generado en la fase de compresión del negocio se evidencia una sobresaliente capacidad del modelo para clasificar el contenido visual real y alterado generado con la herramienta Faceswap. Esto nos sirve para validar el nivel de deteccion del modelo.

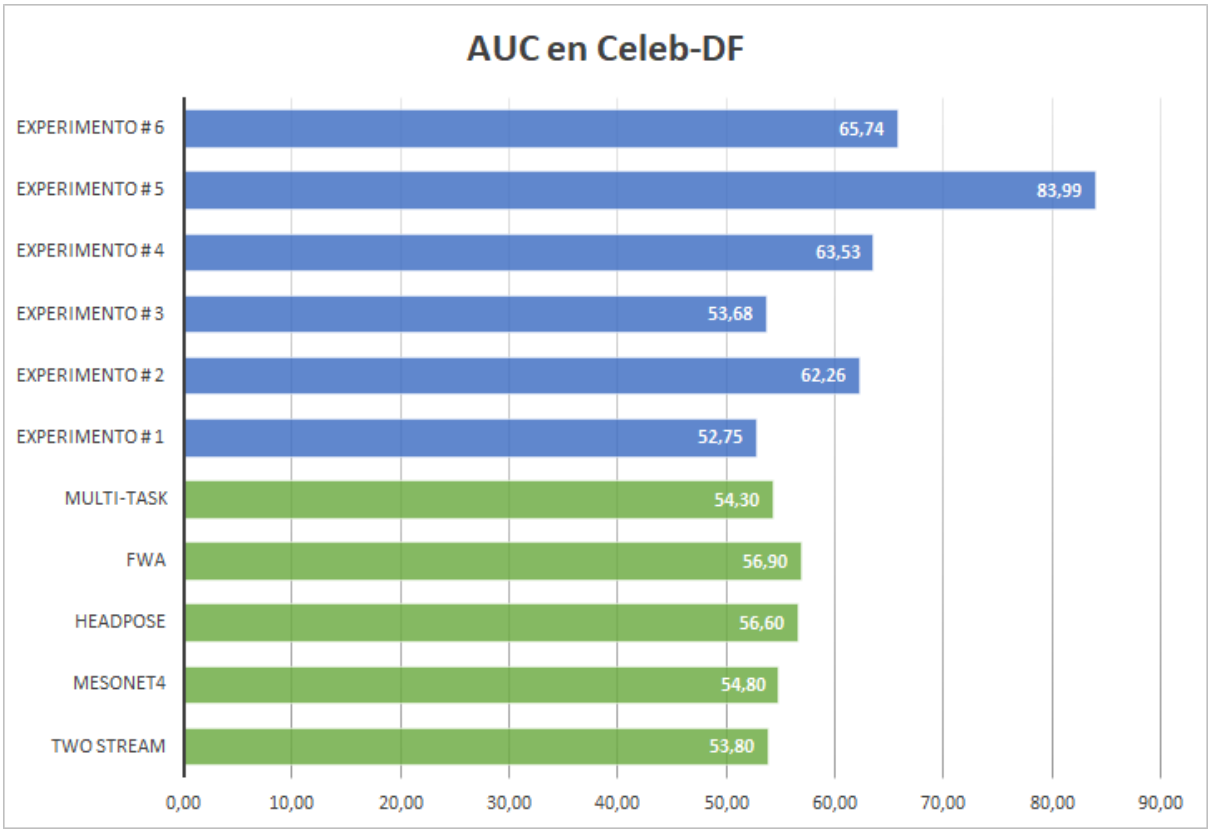

Figura 11. Grafica comparativa de AUC entre estado del arte y experimentos ejecutados. Valores porcentuales.

- De los 6 experimentos entrenados y evaluados solo con el conjunto de datos **Celeb-DF** 4 obtuvieron un **AUC mayor** que los que obtuvieron los métodos de detección expuestos en el estado del arte. El experimento **5** fue el que obtuvo un **mayor AUC**.

Tabla 13. Comparación de AUC entre métodos de detección del estado del arte y el experimento **7**

Método/Datos	DF-TIMIT	FF-DF	DFD	DFDC	Celeb-DF
Two Stream	73,5	70,1	52,8	61,4	53,8
Mesonet4	68,4	84,7	76	75,3	54,8
Headpose	53,2	47,3	56,1	55,9	56,6
FWA	93,2	80,1	74,3	72,7	56,9
Multi-task	55,3	76,3	54,1	53,6	54,3
Experimento # 7	**100**	**96,62**	**81**	**85,9**	**95,25**

- De acuerdo con la Tabla 12 el experimento **7** que fue entrenado con rostros de los 5 conjuntos de datos seleccionados se evidencia que obtiene un mayor valor de **AUC** frente a los métodos de detección expuestos en el estado del arte.

9 CONTRIBUCIONES Y ENTREGABLES

9.1 Contribuciones

9.1.1 Aportes relacionados con el objeto del proyecto

Para la humanidad, una herramienta que permite realizar una diferenciación entre el contenido generado con Deepfake y el contenido real.

Para la comunidad de desarrolladores e investigadores un método de detección mejorado frente a otros métodos en la clasificación de un conjunto de datos de segunda generación.

9.1.2 Aportes relacionados con el desarrollo de capacidades del investigador

Este proyecto de grado nos permitió como investigadores desarrollar diferentes capacidades tales como la comunicación escrita y oral efectiva, la capacidad de síntesis de la información, la gerencia de un proyecto basado en analítica de datos, así como el conocimiento sobre el manejo de contenido visual como un conjunto de datos, y finalmente la aplicación de la inteligencia artificial para la resolución de problemas.

9.2 Entregables

A continuación, se presentan los entregables correspondientes a los objetivos planteados anteriormente:

Objetivo 1. Analizar el funcionamiento y las características de métodos generadores y detectores de Deepfake basado en GAN.

- Documento que detalla las características claves para la generación y detección de los contenidos visuales alterados con Deepfake.

Objetivo 2. Recopilar un conjunto de datos reales y alterados.

- Documento con el conjunto de datos recopilados y su respectiva descripción preliminar.

- Jupyter Notebook que ejecuta el preprocesamiento de los datos recopilados de acuerdo con las características relevantes para la evaluación de los modelos.

- Conjunto de datos propio realizado con Faceswap.

Objetivo 3. Implementar un método de detección de Deepfake.

- Codigo en Python de la red Xception usada.
- Modelos de analítica que clasifica el contenido visual como real o como Deepfake.

Objetivo 4. Evaluar el método propuesto y compararlo con otros métodos existentes.

- Jupyter Notebook que determina la matriz de confusión, el accuracy, el AUC entre otras métricas de un conjunto de evaluación.
- Documento de proyecto final que incluye la evaluación a los métodos desarrollados.

10 CONCLUSIONES Y TRABAJO A FUTURO

En el presente proyecto se propuso un método de detección que permite clasificar contenido visual como real o falso (real, fake) a partir de una red convolucional **Xception**, un **preprocesamiento de imágenes** (extracción de rostro) y una **optimización del umbral de decisión de la red**. Como parte de las conclusiones se tiene que:

- Las métricas de evaluación de calidad para un vídeo original respecto a sus compresiones **(VMFA, PSNR)** no funcionan para comparar la calidad de imágenes diferentes y por lo tanto no es útil para clasificar el contenido visual según la calidad.

- Para el problema de detección de **Deepfake** el modelo **Xception** obtiene mejores resultados cuando se entrena y se evalúa con imágenes que contienen únicamente rostros. Esto se debe a que las imágenes completas contienen información que no es relevante para determinar si una imagen es real o está alterada, lo que hace que el modelo no se enfoque en las características importantes para hacer la distinción.

- La aplicación del método forense **ELA** (Error Level Analysis) sobre los datos de entrenamiento y evaluación no mejora el nivel de detección del método, por el contrario, lo desmejora, haciendo más difícil para la red el reconocimiento de patrones diferenciadores entre imágenes reales y alteradas.

- La red **Xception** mejora su nivel de detección cuando no se fijan los pesos del entrenamiento con **Imagenet**, es decir la red mejora su predicción cuando es entrenada completamente y no solamente el clasificador de la red.

- El método presenta una leve mejoría en la predicción de la clase (real o fake) al calcular un **umbral de decisión que maximiza el AUC**.

- Se evaluó un método capaz de detectar contenido visual real y contenido visual falso para un conjunto de datos compuesto por datos de **Celeb-DF**, **DFDC**, **DFD**, **FF-DF**, **DF-TIMIT** con un accuracy de **92,12%** y un **AUC** del **92,15%** (Experimento **7**). Este método da como resultado al menos un **81%** de accuracy para cada conjunto de datos de manera individual. Adicionalmente obtiene el **mayor valor de AUC** para cada conjunto de datos de manera individual al compararlo con los métodos detectores expuestos en el estado del arte.

Finalmente, como trabajo futuro se propone lo siguiente:

- Experimentar con otro tipo de métodos forenses como el **análisis de ruido** (noise analysis) aplicados a los datos con el objetivo de que los algoritmos de Deep Learning puedan identificar con mayor facilidad artefactos distintivos entre los datos reales y alterados.

- Realizar un entrenamiento del método aquí propuesto con una **mayor cantidad de imágenes** y evaluar si esto mejora la predicción.

- Llevar a producción el método del presente proyecto mediante **un aplicativo web o móvil** que permita a un usuario identificar si un vídeo o imagen es real o está alterado con Deepfake.

11 BIBLIOGRAFÍA

Chesney, R., & Citron, D. (2019). Deepfakes and the New Disinformation War: The Coming Age of Post-Truth Geopolitics. Foreign Affairs, 98(1), 147–155. Recuperado de: http://search.ebscohost.com/login.aspx?direct=true&db=asn&AN=133503997&lang=es&site=ehost-live

Koopman, M., Rodriguez, A. M., & Geradts, Z. (2018). Detection of Deepfake Vídeo Manipulation. *Proceedings of the 20th Irish Machine Vision and Image Processing Conference*, (August), 133–136. https://www.researchgate.net/profile/Zeno_Geradts/publication/329814168_Detection_of_Deepfake_Vídeo_Manipulation/links/5c1bdf7da6fdccfc705da03e/Detection-of-Deepfake-Vídeo-Manipulation.pdf

Weems, M. E. (2017). “Fake America Great Again?” *Qualitative Inquiry*, Vol. 23, pp. 168–170. https://doi.org/10.1177/1077800416674752

Dang, L. M., Hassan, S. I., Im, S., Lee, J., Lee, S., & Moon, H. (2018). Deep learning based computer generated face identification using convolutional neural network. Applied Sciences (Switzerland), 8(12). https://doi.org/10.3390/app8122610

Wagner, T. L., & Blewer, A. (2019). “The Word Real Is No Longer Real”: Deepfakes, Gender, and the Challenges of AI-Altered Vídeo. *Open Information Science*, *3*(1), 32–46. https://doi.org/10.1515/opis-2019-0003

Schwart O. (2018, 02 Nov), You thought fake news was bad? Deep fakes are where truth goes to die*, The Guardian* recuperado de: https://www.theguardian.com/technology/2018/nov/12/deep-fakes-fake-news-truth

Harán J.M. (2019, 11 Sep), Utilizan AI para imitar la voz del CEO de una compañía y robar 220 mil euros*, welivesecurity* recuperado de: https://www.welivesecurity.com/la-es/2019/09/11/estafadores-utilizan-inteligencia-artificial-imitar-voz/

Daniel, G. (2013). Principles of artificial neural networks (Vol. 7). World Scientific.

Basogain, X. (2008). Redes neuronales artificiales y sus aplicaciones. Escuela Superior de Ingeniería de Bilbao, EHU. Departamento de ingeniería de sistemas y automática

Basogain, X. (2008). Redes neuronales artificiales y sus aplicaciones. Escuela Superior de Ingeniería de Bilbao, EHU. Departamento de ingeniería de

sistemas y automática.[Figura 2]. Recuperado de: https://ocw.ehu.eus/pluginfile.php/9047/mod_resource/content/1/redes_neuro/contenidos/pdf/libro-del-curso.pdf

Basogain, X. (2008). Redes neuronales artificiales y sus aplicaciones. Escuela Superior de Ingeniería de Bilbao, EHU. Departamento de ingeniería de sistemas y automática.[Figura 3]. Recuperado de: https://ocw.ehu.eus/pluginfile.php/9047/mod_resource/content/1/redes_neuro/contenidos/pdf/libro-del-curso.pdf

Afchar, D., Nozick, V., Yamagishi, J., Echizen, I., Afchar, D., Nozick, V., … Nozick, V. (2018). *MesoNet: a Compact Facial Vídeo Forgery Detection Network*. (hal-01867298).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. Recuperado de: http://www.deeplearningbook.org/front_matter.pdf

Huang, X., Li, Y., Poursaeed, O., Hopcroft, J., & Belongie, S. (2017). Stacked generative adversarial networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-January*, 1866–1875. https://doi.org/10.1109/CVPR.2017.202

Li, Y., Yang, X., Sun, P., Qi, H., & Lyu, S. (2019). *Celeb-DF: A New Dataset for Deepfake Forensics*. *1*, 1–6. Recuperado de: http://arxiv.org/abs/1909.12962

Nguyen, T. T., Nguyen, C. M., Nguyen, D. T., Nguyen, D. T., & Nahavandi, S. (2019). *Deep Learning for Deepfakes Creation and Detection*. 1–16. Recuperado de: http://arxiv.org/abs/1909.11573

Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). *Deep Face Recognition*. (Section 3), 41.1-41.12. https://doi.org/10.5244/c.29.41

Romero, J. J., Dafonte, C., Gòmez, Á., & Penousal, F. J. (2007). Inteligencia Artificial Y Computación Avanzada. In *Inteligencia Artificial* …. Recuperado de: http://fmachado.dei.uc.pt/wp-content/papercite-data/pdf/ms07.pdf#page=9

Xabier Basogain. (2008). Redes Neuronales Artificiales y sus Aplicaciones. In *Escuela Superior de Ingeniería de Bilbao, UPV-EHU* …. Recuperado de: https://ocw.ehu.eus/file.php/102/redes_neuro/contenidos/pdf/libro-del-curso.pdf

Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Burkov, A. (2019). *The hundred-page machine learning book.* Quebec CIty.

Yang, X., Li, Y., & Lyu, S. (2019). Exposing Deep Fakes Using Inconsistent Head Poses. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *2019-May*, 8261–8265. https://doi.org/10.1109/ICASSP.2019.8683164

Krawetz, N., & Solutions, H. F. (2007). A picture's worth. *Hacker Factor Solutions, 6(2), 2. https://www.blackhat.com/presentations/bh-usa-07/Krawetz/Whitepaper/bh-usa-07-krawetz-WP.pdf*

Li, L., Bao, J., Zhang, T., Yang, H., Chen, D., Wen, F., & Guo, B. (2019). Face X-ray for More General Face Forgery Detection.[Figura 6]. Recuperado de: https://arxiv.org/pdf/1912.13458.pdf

torzdf. (2019) NN de Faceswap [Figura 5]. Recuperado de https://forum.Faceswap.dev/viewtopic.php?t=146

Korshunov, P., & Marcel, S. (2018). Deepfakes: a new threat to face recognition? assessment and detection. arXiv preprint arXiv:1812.08685. Recuperado de https://arxiv.org/pdf/1812.08685.pdf

Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019). Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1-11). Recuperado de https://arxiv.org/pdf/1901.08971.pdf

Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019). Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1-11).[Figura 9]. Recuperado de https://arxiv.org/pdf/1901.08971.pdf

Dolhansky, B., Howes, R., Pflaum, B., Baram, N., & Ferrer, C. C. (2019). The Deepfake Detection Challenge (DFDC) Preview Dataset. *arXiv preprint arXiv:1910.08854*. Recuperado de https://arxiv.org/pdf/1910.08854.pdf

Li, Y., Yang, X., Sun, P., Qi, H., & Lyu, S. (2019). Celeb-df: A new dataset for Deepfake forensics. Recuperado de https://arxiv.org/pdf/1909.12962.pdf

Li, Y., Yang, X., Sun, P., Qi, H., & Lyu, S. (2019). Celeb-df: A new dataset for Deepfake forensics.[Tabla 12]. Recuperado de https://arxiv.org/pdf/1909.12962.pdf

Google. (10 de Febrero de 2020). Google. Recuperado de Google: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=es-419

Li, L., Bao, J., Zhang, T., Yang, H., Chen, D., Wen, F., & Guo, B. (2019). Face X-ray for More General Face Forgery Detection. Recuperado de: https://arxiv.org/abs/1912.13458

Foster, D. (2019). Generative deep learning: teaching machines to paint, write, compose, and play. O'Reilly Media.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258). Recuperado de: http://openaccess.thecvf.com/content_cvpr_2017/papers/Chollet_Xception_Deep_Learning_CVPR_2017_paper.pdf

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).[Figura 4]. Recuperado de: http://openaccess.thecvf.com/content_cvpr_2017/papers/Chollet_Xception_Deep_Learning_CVPR_2017_paper.pdf

Urcuqui, C. C., Peña, M. G., Quintero, J. L. O., & Cadavid, A. N. (2018). Ciberseguridad: Un enfoque desde la ciencia de datos. Universidad ICESI.

Diaz, J., (2019). Aprendizaje Automático 04-01-Supervisado-[ML+Metricas+KNN]. Universidad ICESI.

Diaz, J., (2019). Aprendizaje Automático 04-01-Supervisado-[ML+Metricas+KNN]. Universidad ICESI [Figura 1].Recuperado de:https://github.com/i2tResearch/Ciberseguridad/blob/master/Android/SL%20para%20detección%20de%20Android%20Malware%20(PDG)/Documentación/Doc%20Final%20SL%20Android%20Jhoan_Delgado%20(51901).pdf.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0: Step-by-step data mining guide. *SPSS inc*, *9*, 13. [Figura 7]. Recuperado de: https://d3i71xaburhd42.cloudfront.net/0e30efc80bac996cadb1af04c7253e7b11446187/6-Figure2-1.png

12 ANEXOS

12.1 Enlace del repositorio de los entregables

Los entregables se encuentran disponibles en el repositorio del grupo i2t de la Universidad ICESI, en la carpeta dentro de la carpeta Deepfake, donde se encuentra otra carpeta con el título del proyecto. El enlace (privado) es el siguiente:

https://github.com/i2tResearch/Ciberseguridad.git