Red generativa adversarial GAN para generar rostros humanos identificando características faciales

Jorge Sebastian Mora, Abel Felipe Zambrano, Danny Aldemar Jimenez
Departamento de Ingeniería Eléctrica y Electrónica
Análisis inteligente de señales y sistemas
Universidad de los Andes

***Resumen*— Cada vez son más los sistemas que buscar determinar con eficacia la identidad de una persona por medio de identificación y comparación de diferentes patrones capaces de diagnosticar su autenticidad. El reconocimiento facial, resulta ser uno de los métodos más utilizados en ciberseguridad, que hoy en día permiten localizar personas desaparecías, agilizar trámites, identificar delincuentes entre otras aplicaciones enfocadas en brindar certeza y confiabilidad. En este documento, se expone el trabajo investigativo realizado durante el semestre para el proyecto final de la asignatura análisis inteligente de señales y sistemas, en el cual se entrenó una red convolucional con el objetivo de generar caras para luego ser categorizadas por un clasificador. Inicialmente, el algoritmo utiliza dos bases de datos para generar nuevas imágenes de caras basadas en características reales, establecido como un método de pruebas que permite comprobar que el modelo de clasificador tiene la capacidad de diferenciar entre información real y la generada artificialmente.**

***Index Terms*—Sistemas, convolucional, algoritmo, clasificador , artificial.**

I. Introducción

La seguridad y legitimación de los usuarios resulta ser una necesidad básica en diferentes escenarios en donde se requiere evitar plagios a los que se encuentran sometidas personas, corporaciones o cualquier ambiente que contenga información relevante. Los sistemas biométricos de reconocimiento facial resultan tener cierta ventaja en comparación con los tradicionales, ya que el distinguir características como ojos, boca, nariz, orejas, permiten generar un método más eficaz por medio de atributos que serán siempre parte del usuario, no se pueden perder y no es necesario memorizarlos [1]. El modelo generativo entrenado, es una clase de algoritmo que utiliza aprendizaje no supervisado. Inicialmente toma un conjunto de datos que provee cierta información a partir de una distribución establecida y genera una nueva representación de información estimada, en este caso, imágenes de rostros. El clasificador binario, realiza un proceso basado en la extracción de un conjunto de características globales aplicando transformaciones de imágenes sobre la región enmarcada por el rostro, o locales, por medio del análisis particular de las diferentes características (ojos, mentón, cejas) para la conformación de un modelo representativo del rostro en su totalidad [2]. El documento esta estructurado de la siguiente manera. Primero se aborda a manera general el problema para el cual el método trazado podría ser útil, seguido de esto, se realiza una explicación de la red generativa adversarial GAN y el clasificador, después se exponen los resultados obtenidos de acuerdo con la información generada a medida que el proyecto fue avanzando. Al finalizar, se presentan unas conclusiones y se establecen algunos posibles trabajos futuros con base a lo sustentado.

II. Planteamiento del problema

Fornite es un vídeo juego desarrollado en el año 2017 por la empresa Epic Games [3]. En el modo Battle Royale, el objetivo es competir en solitario o en grupo y sobrevivir en una isla, asesinando a los oponentes con herramientas y armas esparcidas en el entorno hasta ser la ultima persona o grupo en pie. Debido a su gran acogida por los usuarios en el ultimo año, la compañía requiere incluir ciertas mejoras en la versión que se lanzara a finales del año 2019, dentro de las cuales contempla incluir mas jugadores en escena con características cada vez mas reales. El

objetivo de este proyecto es poder suplir una parte de esta necesidad creando caras por medio de la inteligencia artificial, que no pertenezcan a ninguna persona de la vida real y puedan ser utilizadas como el rostro de los personajes del juego. El clasificador seleccionara aquellas que sean aptas de acuerdo con el requerimiento de la entidad.

III. Desarrollo del proyecto

III-1. Red generativa adversarial GAN

Una red generativa adversarial GAN, es un sistema que se plantea como una competencia entre dos redes separadas: una red generadora y una segunda red discriminatoria que trata de clasificar las muestras como procedentes de la distribución verdadera o la distribución del modelo [4]. Para el problema planteado, fueron creados nuevos rostros a partir de dos bases de datos, MNIST que contiene dígitos escritos a mano, usada comúnmente para el aprendizaje de automático y CelebA que contiene imágenes de rostros celebridades. Seguido de esto, se procedió a corroborar los resultados con un discriminador que determina entre el conjunto de imágenes creadas y una base de datos de personas, cuáles de estas son caras o no, y cuales son reales o fueron creadas de manera artificial. En la figura 1, se ilustra el funcionamiento de los dos modelos que se entrenan simultáneamente, donde el modelo generador produce una muestra $\dot{y}$ dada una entrada y de una distribución de probabilidad implícita [5]. El modelo clasificador, dada una entrada produce un escalar de salida que determina a que distribución pertenece. La función de costo es una forma de medir el rendimiento del modelo basado en el entrenamiento establecido [5].

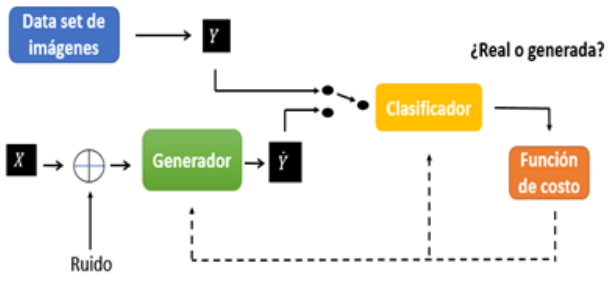

Figura 1. Red generativa adversarial GAN (archivo autor)

III-1.2 Modelo generativo

Dado un conjunto de datos representados por una distribución, un modelo generativo aprende a crear muestras pertenecientes a una nueva distribución. Por lo general, los modelos generativos utilizan la estimación de *Máxima verosimilitud* , En el cual dada una realización $x = (x_1..., x_n)$ de un vector aleatorio $X = (X_1..., X_n)$ distribuido según la función de masa de probabilidad PMF $f_X(x; \theta)$, el método estima un valor de θ que maximice la función para todo θ [6]. Para este contexto, se asume que la distribución del modelo es estimada por medio de las dos bases de datos mencionada (MNIST y CelebA). la verosimilitud $\mathcal{L}$ es la probabilidad conjunta de que todas las muestras de los datos de entrenamiento x "sucedan" en la probabilidad del modelo de distribución [5], tal y como se muestra en la ecuación 1.

$$\mathcal{L}(x, \theta) = \prod_{i=1}^{m} P_{Modelo} \tag{1}$$

Los modelos generativos son basados principalmente en dos grupos, el primero utiliza una función de densidad de probabilidad para ser entrenado y aprender por si mismo. El segundo, extrae muestras de una función de densidad que no esta directamente establecida. Las redes generativas GAN pertenecen al segundo grupo [4]. Nuestro modelo propuesto se basa en tres aspectos. El primero es tomar muestras de ciertas características de las bases de datos, el segundo es el aprendizaje de estas, y el tercero es la reconstrucción o creación de las cara. Inicialmente, las imágenes de las caras de las celebridades fueron convertidas en escala de grises, todas del mismo tamaño 28×28. El generador toma de estos datos de entrada con cierta cantidad de ruido adicionado que permite generar una mayor distribución de las muestras para obtener mejores resultados. Es posible pensar en el generador como un falsificador de dinero, entrenado para engañar al discriminador que se puede pensar Como la fuerza policial. La policía siempre está mejorando, sus técnicas de detección de falsificación al mismo tiempo que Los falsificadores están mejorando su capacidad de producir mejores falsificaciones [5].

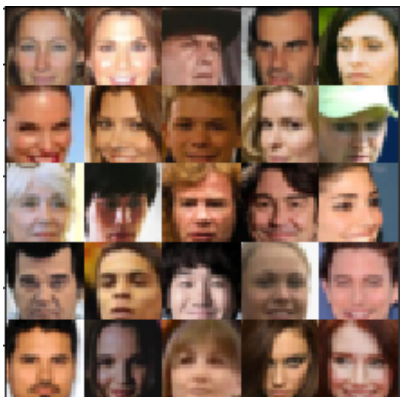

Figura 2. Imágenes de celebridades escaladas a 28x28 - Base de datos CelebA

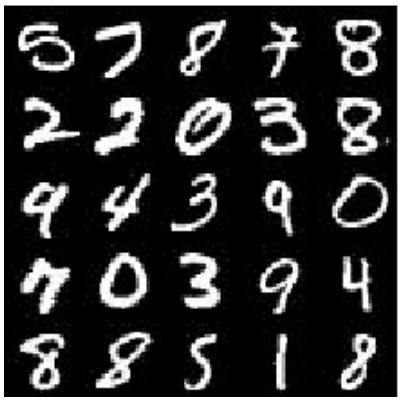

Figura 3. Imágenes de Digitos escaladas a 28x28 - libreria MNIST

III-1.2 Modelo discriminador

El modelo discriminador o clasificador, es un algoritmo previamente entrenado, en el cual, dada una entrada x, por medio de cierta cantidad de parámetros, tiene la capacidad de determinar a que distribución pertenece, si la muestra fue creada por el modelo generativo, es o no una cara, o fue tomada de la base de datos de caras reales. Asigna una etiqueta correspondiente a los datos que se requieran evaluar, ya sean los producidos por el generador o los de la base de datos establecida tal y como se muestra en la figura 1.

III-1.3 Modelo del clasificador binario

Las información generada por la red GAN (imágenes de diferentes caras) es ingresada a un clasificador binario que discrimina entre las clases cara y no cara con el fin de evaluar que tan realistas son las caras generadas por la red. Para esto, se utilizaron diferentes prepocesamientos de datos y dos métodos de clasificación con el fin de evaluar el desempeño de cada uno y usar el que demuestre un mejores resultados. El primer método usado fue una maquina de soporte vectorial (SVM) para una clase la cual puede ser vista como una regresión a un kernel radial y el segundo es isolation forest o bosque aislado el cual genera una frontera que delimita una clase específica con base en la densidad de datos en el espacio multidimensional; este ultimo método también es usado para detección de outliers. para el preprocesamiento, fueron utilizados cuatro métodos: PCA eligiendo 100 componentes, transformada de fourier con PCA de 100 componentes, filtro edge 1 que hace referencia a la aplicación del filtro a la imagen original para posteriormente reducirla a 28 x 28 y después resumir la dimensionalidad a 400 componentes. El último presposamiento usado fue el filtro edge aplicado a la imagen reducida a 28 x 28 y posteriormente utilizando PCA elegir los 100 primeros componentes principales (Edge 2). Para la validación se utilizaron imágenes de caras y de otros elementos que no fueran caras obteniendo los resultados de precisión presentados en la tabla II. Finalmente con los resultados de cada modelo se escogió el preprocesamiento con el filtro edge 2 y la máquina de soporte vectorial.

TABLA I
PRECISIÓN SEGUN PRE-PROCESAMIENTO Y ALGORITMO CLASIFICACIÓN

PRE-PROC	ALGORITMO	
	SVM	IFOREST
PCA 100	66.4 %	66.8 %
FOURIER + PCA 100	58.5 %	58 %
EDGE 1 + PCA 400	51.6 %	51.6 %
EDGE 2 + PCA 100	70.7 %	70.5 %

IV. Implementación del algoritmo y análisis de resultados

Para la elaboración del GAN, el algoritmo fue implementado en el lenguaje de programación Python entorno Jupyter Notebook, utilizando librerías como OpenCV, TensorFlow, keras ideales para realizar tareas de machine learning enfocadas en redes

neuronales y otras utilizadas para realizar análisis de imágenes y operaciones matemáticas. Tanto el modelo generado como el discriminador se componen de redes nuronales unidas de forma serie para evaluar por parte del discriminador las imágenes elaboradas por el generador. La red generadora toma un vector de números aleatorios y lo transforma en una imagen de 28x28. Esta compuesta por 4 capas de neuronas, cada una contiene 512, 256, 128 y 64 neuronas respectivamente (ver Figura 4). Cada una de estas implica una convolución de transposición, normalización de lotes y no linealidad. La biblioteca Tensorflow permite definir fácilmente cada una de estas capas [8]. El discriminador que cuenta con 4 capas de neuronas, cada capa con 58, 112, 232 y 464 neuronas (ver figura 5), toma como entrada una imagen de igual tamaño y la transforma en una probabilidad que determina a que distribución pertenece. Nuevamente se requiere de tf.slim para definir las capas convolucionales, la normalización de lotes y la inicialización de pesos [8], adicionalmente se le aplico un algoritmo conocido como Dropout [10] tanto al generador como al discriminador, el cual consiste en revisar todas las neuronas de las redes y desactivar aquellas que tengan mucha redundancia, esto se realizo con el fin de tratar de evitar el overfitting.

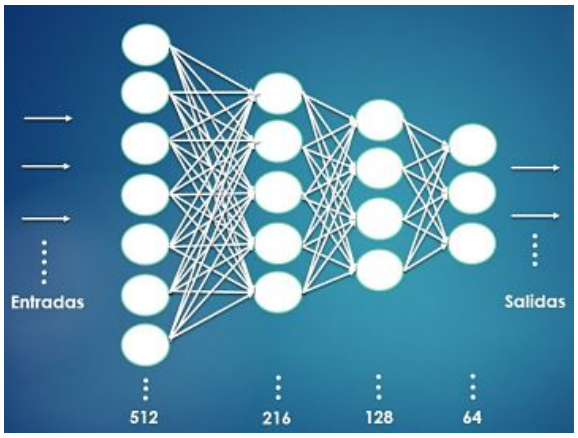

Figura 4. Estructura capas red neuronal generadora

IV.1 Resultados

La red GAN (Generative Adversarial Network) fue entrenada en varias sesiones con aproximadamente 50 épocas por entrenamiento. La optimización

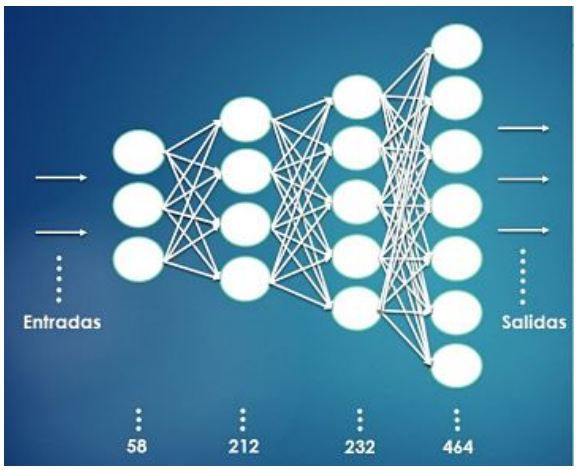

Figura 5. Estructura capas red neuronal discriminadora

del algoritmo que utiliza descenso de gradiente estocástico de primer orden, se realizo mediante una taza de aprendizaje del 0.0002, que es un parámetro el cual determina en qué medida la información recién adquirida anula la información antigua [9]. El factor beta1 influye en el decaimiento de la tasa de aprendizaje [5], este tuvo un valor de $0{,}3$. En la figura 7 se observa como al el discriminador otorga una alta probabilidad a los datos que inicialmente valida como reales, con el paso de las épocas, la probabilidad concedida disminuye considerando estas muestras como datos generados. Esto se demuestra por medio de la expresión 2, el primer termino corresponde a la optimización de la probabilidad de que los datos reales (y) tengan una calificación alta [7]. El segundo término corresponde a la optimización de la probabilidad de que los datos generados $G(\dot{y})$ tengan una calificación baja. Lo contrario el discriminador, el generador fue optimizado para que los nuevos datos formados tuviesen una probabilidad alta (ver figura 8), tal y como se ilustra en la ecuación 3

$$\nabla(\theta_g)\frac{1}{m}\sum_{i=1}^{m}\Big[\log D(y^i) + \log(1 - D(G(\dot{y}^{(i)})))\Big] \quad (2)$$

$$\nabla(\theta_g)\frac{1}{m}\sum_{i=1}^{m}\log(1 - D(G(\dot{y}^{(i)}))) \quad (3)$$

En la figura 6 se observan 32 caras generadas por el algoritmo. Cada columna corresponde a los mejores resultados por época, desde la época 1 hasta la 35 en saltos de 5 épocas.

Figura 6. Caras generadas; cada columna de izquierda a derecha representa las épocas 1, 5, 10, 15 ... hasta la época 35.

Para añadir se vio en la necesidad de implementar una tarjeta gráfica para el entrenamiento de los modelos, debido a su complejidad, se analizaron tres arquitecturas y se obtuvo los siguientes resultados:

TABLA II
ARQUITECTURA DEL PC VS TIEMPO POR ÉPOCA

Arquitectura	Tiempo en minutos
Procesador (i7 de 7 generación serie HQ)	480
Targeta grafica (Nvidia GTX 960M)	120
Targeta grafica (Nvidia Titan XP)	20

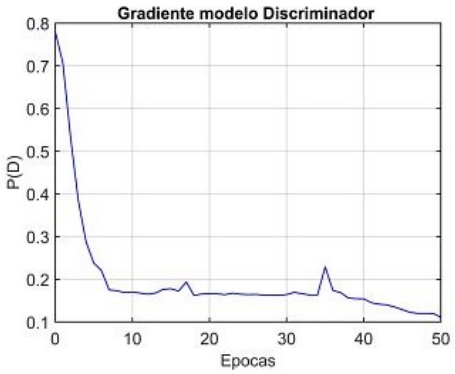

Figura 7. Gradiente discriminador

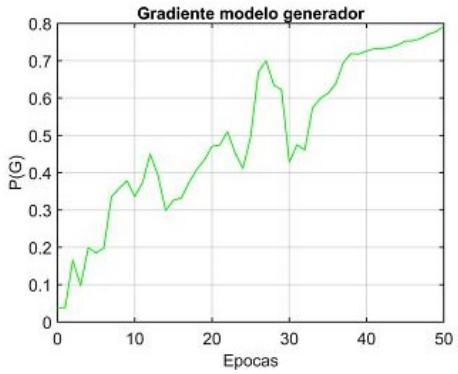

Figura 8. Gradiente Generador

V. CONCLUSIONES

A continuación se muestran las conclusiones más importantes de los resultados obtenidos luego de haber implementado la metodología descrita anteriormente.

- Se encontró que las redes GAN tienden fácilmente al overfitting, por lo que se debe implementar diversas metodologías para evitarlo, como por ejemplo el aplicado en esta implementación el Dropout.
- Se generaron caras artificialmente las cuales mezclaron características de las imágenes de entrenamiento con el fin de generar imágenes totalmente nuevas.

- Se integraron las redes GAN con el algoritmo de clasificación binario y se evaluaron las caras generadas por el primero permitiendo concluir que un 56 % de las caras generadas se clasificaban como caras reales.
- Se vio en la necesidad de utilizar una tarjeta gráfica bastante potente para la implementación de esta red, debido a lo complejo de las redes correspondientes al generador y al discriminador.
- En este documento se presentó el desarrollo de un algoritmo que tuvo como finalidad un acercamiento a la generación de caras. Como trabajo futuro se propone aplicar métodos semi-supervisados, que puedan extraer más características y mejorar el rendimiento tanto del clasificador como del generador disponiendo de datos etiquetados

REFERENCIAS

[1] K. Eun ko, k. Bo Sim. *Development of a Facial Emotion Recognition Method based on combining AAM with DBN*, School of Electrical and Electronics Engineering Chung-Ang University, International Conference on Cyberworlds, 2010.

[2] j. Valvert. *Métodos y técnicas de reconocimiento de rostros en imágenes digitales bidimensionales*, Universidad de San Carlos, Facultad de ingeniería, Guatemala, Junio 2016

[3] Fortnite (videojuego), en Wikipedia, revisado en mayo 2019, de https://es.wikipedia.org/wiki/Fortnite(videojuego)

[4] j. Baruffaldi, L. Uzal.*Redes neuronales adversarias para el reconocimiento de malezas* , Facultad de Ciencias Exactas, Ingeniería y Agrimensura. Universidad Nacional de Rosario CIFASIS, CONICET – UNR.

[5] A. Moacir, S. Leonardo, F. Ribeiro, S. Nazare. *Everything you wanted to know about Deep Learning for Computer Vision but were afraid to ask*, University of S˜ao Paulo, University of Surrey, Junio 2016

[6] P. Bertsekas, N. Tsitsiklis. *Introduction to Probability*, SECOND EDITION, Massachusetts Institute of Technology. 2008

[7] I. Goodfellow. *Generative Adversarial Networks*, NIPS 2016 Tutorial.

[8] A. Radford, S. Chintala *UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS*, ICLR 2016.

[9] Learning rate, en Wikipedia, revisado en mayo 2019, de https://en.wikipedia.org/wiki/Learningrate.

[10] Dropout explained and implementation in Tensorflow http://laid.delanover.com/dropout-explained-and-implementation-in-tensorflow/