Introducción a los Sistemas Inteligentes
Reporte - Proyecto Final

Evaluación de la generalización de un detector de imágenes falsas entrenado como parte de una red GAN convolucional

Andrés Ricardo Pérez Rojas - riperezro@unal.edu.co
CC 1020843903
Febrero 9 de 2022

Link del video con la presentación

El video está disponible en: https://youtu.be/kYnFEPmVqds

1. Business Understanding

Para este paso, debemos plantear un contexto hipotético en el cual se desarrolla el proceso de Data Mining. La situación hipotética planteada para este proyecto es la de una aplicación en la que los usuario publican y venden artículos de ropa. Los vendedores suben fotografías de sus artículos en venta para que los potenciales compradores puedan verlas y tomar una decisión.

Después de un tiempo, la aplicación empezó a tener problemas con usuarios que estaban modificando las fotos por diversos medios para engañar a los compradores, incluso usando imágenes generadas por inteligencia artificial de productos que no existen realmente. Ante esto, el equipo detrás de la aplicación debe tomar una decisión sobre cómo manejar la situación. Contratar moderadores que analicen detalladamente todas las fotos subidas por los usuarios es ineficiente, costoso, limitaría el crecimiento que hasta ahora ha venido teniendo la aplicación y afectaría la experiencia de los usuarios al hacer mucho más lento el proceso de publicar un nuevo artículo.

Esto lleva al equipo a pensar en posibles soluciones automatizadas para detectar imágenes generadas o adulteradas. Una de las opciones para construir este detector es mediante las redes neuronales GAN (Generative Adversarial Networks). El objetivo del proceso de Data Mining en este caso es determinar la viabilidad de esta opción para filtrar las fotos que no sean reales.

Determining Business Objectives

A partir del contexto planteado, podemos formular los posibles objetivos de negocio del proyecto:

- Evaluar la posibilidad de automatizar este proceso de verificación de autenticidad de las imágenes subidas por los usuarios.
- Determinar la viabilidad y efectividad de un detector entrenado como parte de una GAN junto a un generador que busca emular imágenes reales de prendas de ropa.

Como posibles criterios de evaluación de éxito del proyecto:

- Entrenar un detector en como parte de una GAN para ver sus resultados y poder medir su desempeño.
- Evaluar mediante métricas de desempeño la generalización del modelo a partir del aprendizaje en un subconjunto de pruebas.
- A partir de las métricas, realizar un análisis de la posibilidad y practicidad de aplicarlo para detectar automáticamente imágenes generadas o modificadas.
- Plantear posibles alternativas o siguientes pasos a partir de los resultados encontrados en el proyecto.

Assessing the Situation

A modo de experimento, se usará el dataset de Fashion MNIST, disponible en la plataforma Kaggle. Este dataset consta de imágenes de 28 x 28 píxeles en escala de grises. Las imágenes pertenecen a una de 10 clases, siendo cada clase un tipo de artículo de ropa (como camiseta, vestido, saco, bolso, entre otros). El dataset viene de la compañía de e-commerce Zalando.

Dado que el objetivo del proyecto es determinar la viabilidad de este método, partiremos de una arquitectura de red establecida en un notebook publicado en Kaggle para el montaje experimental. El notebook elegido para esta tarea es “Introduction to GANs on Fashion MNIST Dataset”. Este notebook tiene como objetivo la generación de imágenes imitando al dataset de Fashion MNIST. Primero entrena una red convencional con capas densas, la cual al final es capaz de generar imágenes similares pero no muy claras. Después pasa a usar redes convolucionales, lo cual el ayuda al modelo a aprender mejor las estructuras de las imágenes y la relación entre los píxeles adyacentes, obteniendo resultados significativamente mejores.

Se realizará una comparación de los resultados obtenidos con esta arquitectura contra los resultados obtenidos usando otra arquitectura de red convolucional usada en un notebook de Kaggle, en este caso “Introduction to GANs with Keras”. En este notebook solo se entrena la red con una sola clase. Podremos observar cómo se comporta esta arquitectura con un espacio de entrada más amplio y diverso.

Parte de los riesgos de construir el detector de esta forma es la falta de datos manipulados por usuarios en este escenario hipotético. Idealmente en un caso real se podrían recopilar una cantidad significativa de imágenes adulteradas subidas a la plataforma para evaluar mejor la generalización del modelo.

Determining Data Mining Goals

El objetivo principal es lograr entrenar un detector capaz de diferenciar las imágenes reales de las adulteradas o generadas. Sí funciona de manera aceptable con este dataset experimental, se puede pensar en pasar a usar imágenes verdaderas y analizar el desempeño de la red entrenada con estos datos.

Realizar el ejercicio con los datos de prueba de Fashion MNIST se puede utilizar para validar el potencial de seguir con esta alternativa para automatizar el proceso de detección y analizar qué arquitecturas llevan a mejores resultados.

2. Data Understanding

Describing Data

El dataset de Fashion MNIST cuenta con 70 mil imágenes de 28 x 28 en escala de grises. Cada una de las imágenes pertenece a una de 10 clases, que representan diferentes artículos de ropa y accesorios. Es una cantidad significativa de datos que nos permite tomar 3 clases como anomalías para evaluar la generalización del discriminador entrenado, además de repartir los datos que de las clases que se usarán para el entrenamiento en un subconjunto de entrenamiento y un subconjunto de pruebas.

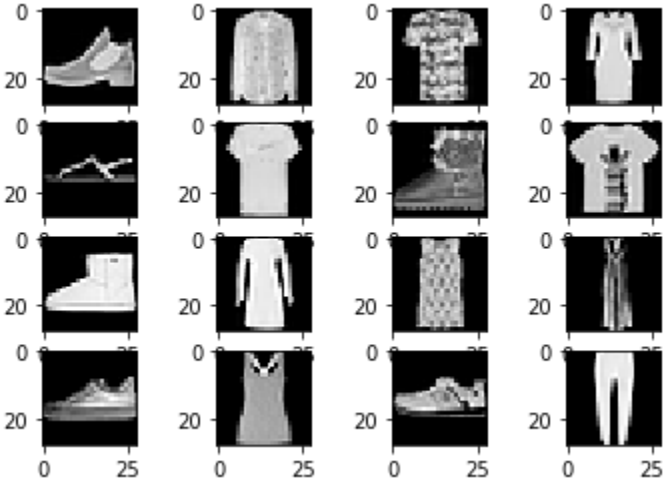
El formato de las imágenes del dataset es un vector de 784 valores entre 1 y 255, y adicionalmente un entero entre 0 y 9 que representa la clase. Las imágenes deben ser convertidas a una matriz de 28 x 28 para poder utilizarlas con redes convolucionales y que sus filtros funcionen correctamente.

Describing Data

La cantidad de muestras de cada clase es igual. Las clases incluidas en el dataset de Fashion MNIST son:

0. Camiseta / Top
1. Pantalón
2. Pullover
3. Vestido
4. Saco
5. Sandalia
6. Camisa
7. Zapatilla deportiva
8. Bolso
9. Bota

A continuación se muestran algunas de las imágenes del dataset:



Algunas de las imágenes de escala de grises del dataset Fashion MNIST

3. Data Preparation

Selecting Cleaning, and Formatting Data

El dataset se compone completamente de imágenes válidas, por lo que en este caso no es necesario realizar una selección de filas para eliminar muestras vacías o inválidas. Sí se debe hacer una selección de columnas antes de realizar el entrenamiento del modelo, dado que las redes GAN no necesitan la columna con la clase de la muestra. El generador recibe como entrada ruido y da como salida una imágen generada. El discriminador recibe como entrada una imagen y, sin necesidad de conocer la etiqueta, trata de distinguir sí la imagen es real o generada por el generador. Por ende, después de separar las clases anómalas, se puede suprimir la columna de clase de las muestras.

4. Modeling

Selecting Modeling Techniques

El modelo elegido para el proyecto son las redes neuronales GAN. En esta arquitectura, se crean dos redes neuronales que compiten entre sí. Para el caso de generación de imágenes, la primera red, llamada generador, recibe ruido aleatorio como entrada y genera imágenes como salida. La segunda red, el discriminador, recibe imágenes como entrada y da como resultado un número entre 0 y 1, la confianza que tiene en que la imágen sea real y no sea generada por el generador. De esta forma las redes compiten entre sí, el generador tratando de engañar al discriminador; y el discriminador tratando de no ser engañado por el generador.

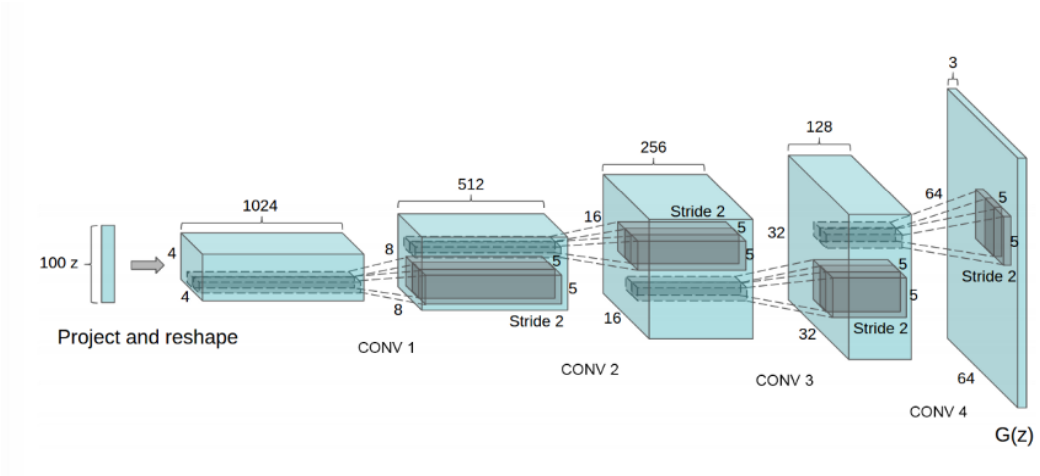
El entrenamiento de estas redes se hace en conjunto. El discriminador se entrena tomando muestras del conjunto de entrenamiento con las imágenes que se quieren imitar, que en este caso son imágenes del dataset de Fashion MNIST de las clases que no fueron elegidas como anomalías. Se entrena el discriminador de manera supervisada dado que sabemos que la salida para estas imágenes idealmente debe ser 1, dado que son reales. Además, se generan imágenes con el generador actual, y se entrena de manera similar el discriminador con estas entradas, con la etiqueta 0 para todos estos casos dado que son imágenes generadas.

El generador se entrena de manera un poco diferente. Primero, se debe configurar el discriminador como no entrenable. Con esta configuración, sí entrenamos la GAN completa, solo se modifican los pesos del generador. Desde el punto de vista del generador, el comportamiento deseado es que la GAN reciba como entrada ruido aleatorio, y de como salida 1. Es decir que la imágen generada por el generador, que es la entrada dada al discriminador, es considerada como real por el discriminador. Por ende, para el entrenamiento se le da como entrada varios vectores de ruido aleatorio, y se entrena de manera supervisada con la etiqueta 1 para todas las muestras. Los pesos del discriminador no cambian, por lo que el resultado es que el generador se vuelve mejor en engañar la versión actual del discriminador.

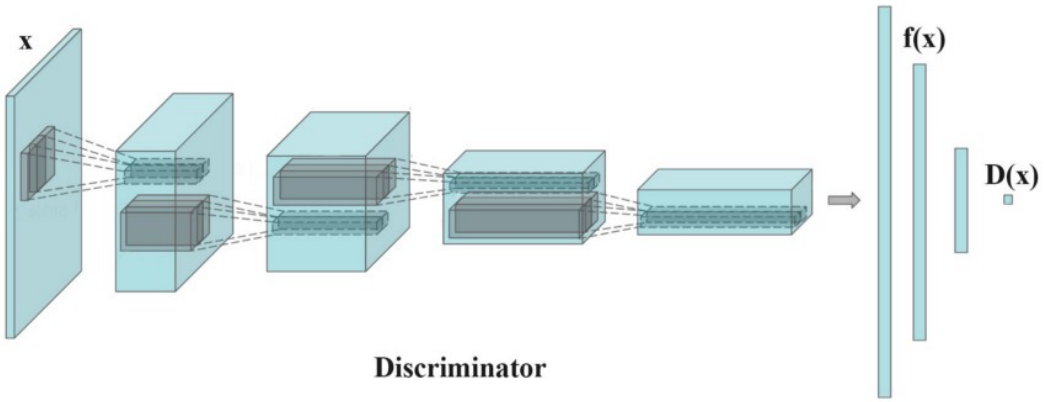
Este proceso de entrenamiento se repite varias veces y de esta manera el generador y el discriminador aprenden y mejoran su desempeño juntos. A medida que el generador mejora el discriminador debe mejorar también para poder distinguir las imágenes generadas de las reales. A su vez, a medida que el discriminador mejora, el generador debe mejorar también generando imágenes más parecidas a las del dataset para lograr engañar al discriminador.

Adicionalmente, para el proyecto se eligieron redes GAN convolucionales, conocidas también como DCGANs. Cuando se trabaja con imágenes, se suelen utilizar capas convolucionales, que utilizan filtros que le permiten a la red comprender mejor las estructuras de las imágenes al relacionar pixeles vecinos. En este caso, tanto el generador como el discriminador tienen capas convolucionales de 2 dimensiones, sin necesidad de canales dado que las imágenes del dataset de Fashion MNIST solo tienen un único canal, la intensidad en escala de grises.

A continuación, se presentan ilustraciones generales de cómo se estructuran las redes de este tipo. No corresponden a las arquitecturas utilizadas en este proyecto.



Estructura posible para un generador convolucional [6]



Estructura posible para un discriminador convolucional [8]

5. Evaluation

Para evaluar la generalización del discriminador entrenado con cada arquitectura, se escogieron inicialmente 3 de las 10 clases para que fueran anomalías. Las muestras de las clases restantes fueron divididas en subconjuntos de entrenamiento y de prueba. De esta forma, se tienen 3 tipos de muestras en las cuales evaluar el desempeño del discriminador como detector de imágenes generadas artificialmente:

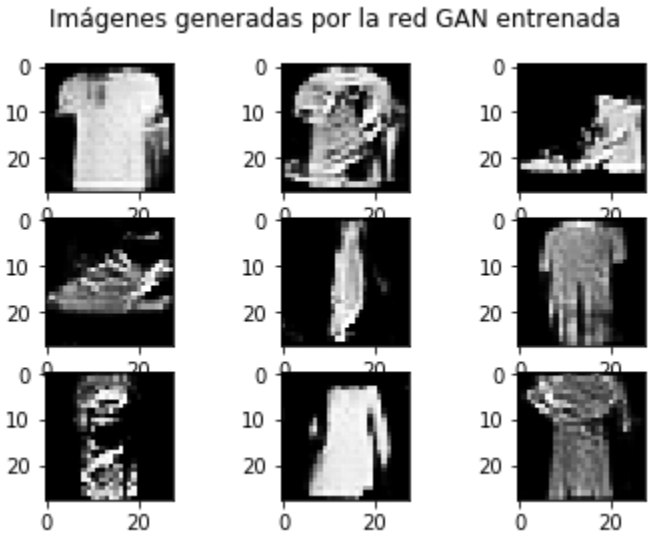
- Imágenes generadas por el generador entrenado junto al detector
- Imágenes reales de las clases anómalas en las cuales el detector no fue entrenado
- Imágenes reales de las clases no anómalas en las cuales el detector sí fue entrenado

En términos de métricas de desempeño, podemos empezar calculando la matriz de confusión general con todas las muestras de prueba. Es especialmente importante el recall cuando se toma imagen generada como la clase positiva, dado que esto indica el desempeño del detector cuando se le presenta una imagen generada. También es importante la precisión para saber

cuántos de los reportes de imágenes generadas realmente corresponden a imágenes falsas. Además, el recall tomando las clase positiva como imagen real es relevante en la práctica para el caso de uso planteado, dado que sí muchas imágenes reales son detectadas como falsas, muchos usuarios que están siguiendo las reglas y subiendo imágenes reales se verán afectados por demoras mientras su caso es revisado y se les da el visto bueno para hacer su publicación.

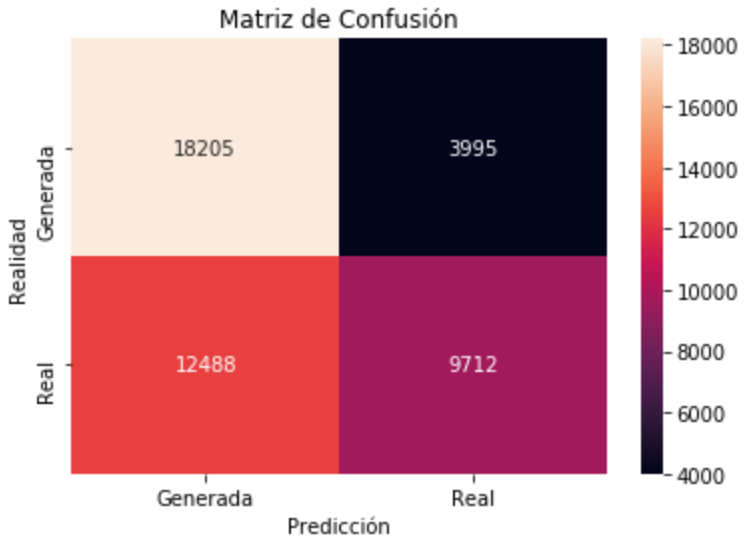
Resultados de la primera red

Algunas imágenes generadas por el generador de la primera red:



Matriz de confusión y métricas generales de desempeño:

	Precision	Recall	F1 score	Support
Generadas	0.59	0.82	0.69	22200
Reales	0.71	0.44	0.54	22200
Accuracy			0.63	44400
Macro avg	0.65	0.63	0.61	44400
Weighted avg	0.65	0.63	0.61	44400



Uno de los resultados más importantes para el caso de uso es el recall para las imágenes generadas, dado que indica qué porcentaje de las imágenes falsas son detectadas correctamente. Este primer modelo tiene resultados satisfactorios en esta medida, detectando el 82% de las imágenes generadas.

Pero, al analizar estos resultados se puede visualizar un problema. La precisión para las imágenes generadas es baja (59%), y a su vez el recall para las imágenes reales es bajo también (44%). Esto nos dice que muchas imágenes reales están siendo clasificadas también como falsas, lo cual trae problemas para el caso de uso. La idea de automatizar el proceso es minimizar las revisiones que se hacen a imágenes reales, dado que esto ralentiza la experiencia del usuario. Esto puede traer problemas para aplicar el modelo en producción.

Resultados con imágenes reales anómalas:

	Precision	Recall	F1 score	Support
Reales	1.00	0.43	0.60	18000
Accuracy			0.43	18000

En este caso la precisión no da información adicional dado que todas las imágenes son reales. El modelo no tiene un desempeño satisfactorio con las imágenes reales de las clases en las que no fue entrenado, lo cual da una mala señal de su generalización y aplicación en muestras diferentes a las de su entrenamiento

Resultados con imágenes reales no anómalas:

	Precision	Recall	F1 score	Support

Reales	1.00	0.46	0.63	4200
Accuracy			0.46	4200

En este caso la precisión no da información adicional dado que todas las imágenes son reales. El modelo no tiene un desempeño satisfactorio con las imágenes reales de las clases en las que fue entrenado, con muestras que no fueron usadas en su entrenamiento. Esto puede ser indicativo de un posible sobreajuste a las muestras utilizadas en el entrenamiento, o que el generador no logró mejorar lo suficiente para exigir al discriminador.

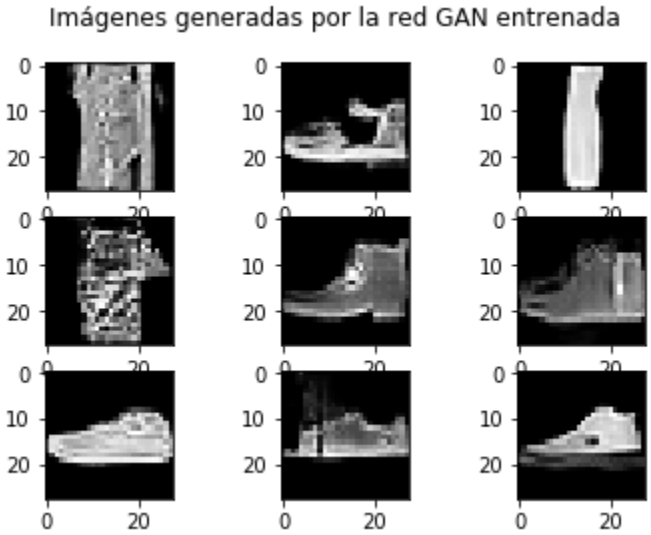
Resultados con imágenes generadas por la otra red GAN:

	Precision	Recall	F1 score	Support
Generadas	1.00	0.79	0.88	10000
Accuracy			0.79	10000

En este caso la precisión no da información adicional dado que todas las imágenes son generadas. El modelo no tiene un desempeño alto con las imágenes que genera el generador de la segunda red GAN. Esto indica que potencialmente la detección de imágenes generadas sí es generalizable, pero en conjunción con los demás resultados, los problemas surgen al enfrentarse a muestras que sean imágenes reales.

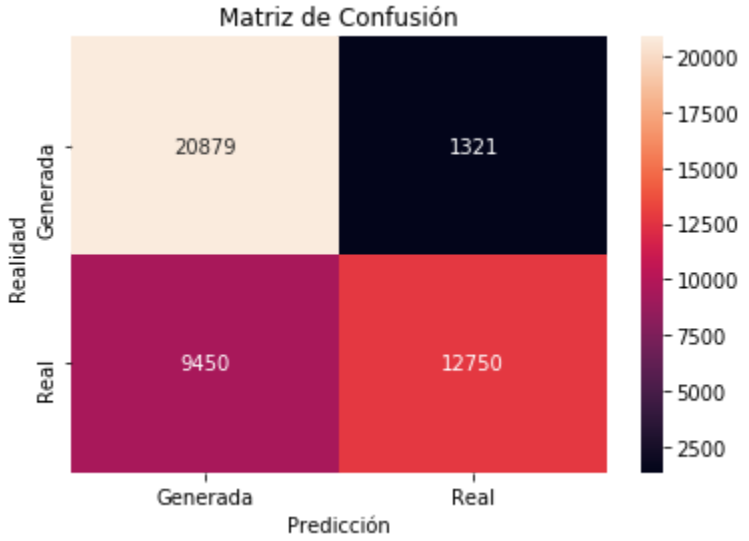
Resultados de la segunda red

Algunas imágenes generadas por el generador de la segunda red:



Matriz de confusión y métricas generales de desempeño:

	Precision	Recall	F1 score	Support
Generadas	0.69	0.94	0.79	22200
Reales	0.91	0.57	0.70	22200
Accuracy			0.76	44400
Macro avg	0.80	0.76	0.75	44400
Weighted avg	0.80	0.76	0.75	44400



Para el caso de la segunda red, el recall para las imágenes generadas es incluso mejor, llegando al 94% en las pruebas realizadas. En este caso se tiene una mejor precisión con las imágenes generadas (69%) y a su vez se tiene un mejor recall con las imágenes reales (57%).

Estos resultados son más prometedores que los de la red anterior para llevar a producción, aunque siguen teniendo problemas. Se tiene una buena cobertura de los casos con imágenes generadas, no muchas están siendo clasificadas como reales. Pero una cantidad significativa de imágenes reales siguen siendo clasificadas como generadas, que en el caso de uso planteado puede generar problemas. Al menos esta vez la mayoría de las imágenes reales son clasificadas correctamente. Una fortaleza del modelo es que sí decide clasificar una imagen como real, aparentemente se puede tener bastante certeza de que en efecto es una imágen real.

Resultados con imágenes reales anómalas:

	Precision	Recall	F1 score	Support
Reales	1.00	0.57	0.72	18000
Accuracy			0.57	18000

En este caso la precisión no da información adicional dado que todas las imágenes son reales. Los resultados con las clases anómalas no son impresionantes, pero sí representan una mejora considerable con respecto a la primera red. Podemos intuir que la red sí aprendió a distinguir algunos rasgos de las imágenes falsas que no están en una imágen verdadera, y puede reconocer en la mayoría de casos que la imágen es real a pesar de que pertenezca a una clase que no vio en el entrenamiento, dando pie a una posible generalización adecuada.

Resultados con imágenes reales no anómalas:

	Precision	Recall	F1 score	Support
Reales	1.00	0.61	0.75	4200
Accuracy			0.61	4200

En este caso la precisión no da información adicional dado que todas las imágenes son reales. Los resultados con las clases que sí vio en el entrenamiento, pero con imágenes que no estaban presentes en el dataset de entrenamiento, son buenos pero no excelentes. Logra identificar significativamente más de la mitad de estas muestras como imágenes reales. Estos resultados junto a los obtenidos con las clases anómalas sugieren que se tiene cierto grado de generalización incluso con imágenes reales, a diferencia de la red anterior.

Resultados con imágenes generadas por la otra red GAN:

	Precision	Recall	F1 score	Support
Generadas	1.00	1.00	1.00	10000
Accuracy			1.00	10000

En este caso la precisión no da información adicional dado que todas las imágenes son generadas. La segunda red fue capaz de identificar la totalidad de las imágenes generadas por el primer generador como falsas. Estos resultados, unidos con los resultados de la primera red, nos llevan a pensar que la identificación de las imágenes falsas es más fácil de aprender para los modelos planteados que la identificación de imágenes reales.

6. Deployment

El objetivo de este proyecto es evaluar el desempeño de un discriminador entrenado como parte de una red GAN para distinguir imágenes reales de imágenes generadas. Dado que el dataset de Fashion MNIST es un dataset sencillo, realmente no se puede llevar a producción realmente.

Sin embargo, los resultados son prometedores, en especial con la segunda arquitectura probada en este proyecto. Refinando un poco más la arquitectura, y recolectando datos reales con más canales, como suelen ser las imágenes subidas a plataformas en internet como la del caso de uso sugerido, se pueden llegar a tener resultados positivos, utilidad para la compañía y una mejor experiencia en general para los usuarios. Incluso con estas arquitecturas sencillas, se tiene una reducción en la cantidad de casos a ser revisados manualmente en comparación a revisar la totalidad de las imágenes subidas por los usuarios a la plataforma. Es necesario experimentar con imágenes con más canales antes de llegar a la conclusión de que es posible llevar un servicio de este caso en un ambiente real, pero los resultados encontrados hasta ahora sugieren que es posible y tiene potencial utilidad.

También existe la posibilidad de usar una arquitectura diferente para incorporar a los datos de entrenamiento del detector imágenes falsas generadas con varias de las arquitecturas de GAN más ampliamente utilizadas. Como pudimos ver, arquitecturas diferentes pueden generar las imágenes de manera diferente. Una arquitectura así se expone en Hsu et al., obteniendo resultados satisfactorios.

Referencias

[1]Zalando Research, "Fashion MNIST", Kaggle.com, 2017. [Online]. Available: https://www.kaggle.com/zalando-research/fashionmnist?select=fashion-mnist_train.csv. [Accessed: 15- Jan- 2022].

[2]Sayak, "Introduction to GANs on Fashion MNIST Dataset", Kaggle.com, 2020. [Online]. Available: https://www.kaggle.com/sayakdasgupta/introduction-to-gans-on-fashion-mnist-dataset. [Accessed: 15- Jan- 2022].

[3]A. Goel, "Introduction to GANs with Keras", Kaggle.com, 2020. [Online]. Available: https://www.kaggle.com/yushg123/introduction-to-gans-with-keras. [Accessed: 15- Jan- 2022].

[4]"Deep Convolutional Generative Adversarial Network | TensorFlow Core", TensorFlow, 2022. [Online]. Available: https://www.tensorflow.org/tutorials/generative/dcgan. [Accessed: 06- Feb- 2022].

[5]IBM, "IBM SSPS Modeler CRISP-DM Guide", Ibm.com, 2021. [Online]. Available: https://www.ibm.com/docs/en/spss-modeler/SaaS?topic=guide-introduction-crisp-dm. [Accessed: 15- Jan- 2022].

[6] A. Radford, L. Metz, S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. 2015. [Online]. Available: https://arxiv.org/abs/1511.06434. [Accessed: 06- Feb- 2022].

[7]C. Shorten, "Deeper into DCGANs", Medium, 2019. [Online]. Available: https://towardsdatascience.com/deeper-into-dcgans-2556dbd0baac. [Accessed: 06- Feb- 2022].

[8]J. Hui, "GAN — Ways to improve GAN performance", Medium, 2018. [Online]. Available: https://towardsdatascience.com/gan-ways-to-improve-gan-performance-acf37f9f59b. [Accessed: 06- Feb- 2022].

[9]C. Hsu, C. Lee and Y. Zhuang, Learning to Detect Fake Face Images in the Wild, 3rd ed. 2018. [Online]. Available: https://arxiv.org/abs/1809.08754. [Accessed: 06- Feb- 2022].