Desarrollo de un algoritmo de Machine Learning para la detección de contenido multimedia generado por medio de DeepFake

Ballesteros Santiago, Barón Daniel, Mejía Magda, Portocarrero Larry
Universidad de los Andes, Colombia

Resumen *– En los últimos años gracias a la aparición de las redes sociales y el internet se han generado gran cantidad de datos en formato de imágenes y video. Del mismo modo en que se generan nuevos datos, han aparecido nuevas tecnologías para generar datos falsos. Una de las técnicas más recientes desarrolladas es DeepFake, la cual consiste en la creación de videos o imágenes hiperrealistas utilizando inteligencia artificial. Esta tecnología puede ser utilizada para manipular la imagen pública o suplantar la identidad de celebridades o personajes importantes. Es por eso que surge la necesidad de desarrollar un algoritmo para la detección del contenido generado por medio de DeepFake. Para desarrollar el algoritmo se utiliza el dataset proporcionado por el desafío DeepFake Detection disponible en Kaggle. Del mismo modo, se comparan los resultados de diferentes modelos construidos a partir de las redes pre-entrenadas InceptionV3 y VGG16. Por último, se escoge el modelo construido a partir de la red pre-entrenada VGG16 para desarrollar una aplicación en la cual el usuario final pueda detectar si un video fue modificado o no por medio de DeepFake. Lo anterior debido a que este modelo fue el que generó mejores resultados sobre el conjunto de prueba.*

Índice de Términos *– DeepFake, InceptionV3, VGG16, Kaggle, dataset, entrenamiento, validación, prueba, imágenes, videos.*

I. INTRODUCCIÓN

La cantidad de datos creados en todo el mundo en 2018 alcanzó los 33 ZB (un ZetaByte (ZB) equivale a 1.000 millones de TeraBytes (TB)), 16,5 veces más que solo hace nueve años. No obstante, gracias a los nuevos avances tecnológicos como el internet de las cosas y las redes sociales se estima que la cantidad de información digital generada en 2035 ascenderá a los 2.142 ZB [1]. Por otro lado, gracias a la facilidad con la cual la información es compartida por medio del internet y las redes sociales, ha surgido un gran inconveniente relacionado con la veracidad de dicha información. Una encuesta de alrededor de 25.000 participantes realizada por Ipsos en nombre del *Centre for International Governance Innovation* (CIGI) revelan que alrededor del 86% de los encuestados creen que han estado expuestos a noticias falsas. Las principales fuentes de información falsa son redes sociales como Facebook y YouTube e inclusive la televisión y los sitios web [2].

Del mismo modo, gran parte del contenido disponible se encuentra en formato de imágenes o vídeos, y gracias a la aparición de tecnologías como la Inteligencia Artificial, se han desarrollado nuevas técnicas para crear contenido multimedia falso. Una de las técnicas más recientes desarrolladas es DeepFake, la cual consiste en la creación de videos o imágenes hiperrealistas utilizando inteligencia artificial [3]. El contenido multimedia generado por medio de DeepFake consiste en videos de personas haciendo y diciendo cosas irreales, a tal punto que es difícil identificar si el contenido es real o generado por DeepFake. Un ejemplo del resultado de la alteración de imágenes por medio de DeepFake se puede observar en la figura 1.

Figura 1. (I) Imagen original de Allison Brie. (D) Imagen modificada usando DeepFake para poner la cara de Jim Carrie en lugar de la cara de Allison Brie [23].

II. PLANTEAMIENTO DEL PROBLEMA

La generación de contenido multimedia falso por medio de DeepFake tiene un alto potencial dañino en el uso y manipulación de la imagen pública de la persona afectada debido a la calidad del contenido creado, que llega al punto de que es casi imposible para un ser humano diferenciar si un video es real o fue creado por

medio de DeepFake. Por ejemplo, en 2017 una celebridad se enfrentó a una situación controversial debido a la circulación de un video pornográfico falso en el ciberespacio [4], y hoy en día el uso de esta tecnología está en manos del público por lo cual cualquier persona que sepa usar herramientas como DeepFaceLab [5], está en la capacidad de generar DeepFakes sobre actores, personas famosas, y en general cualquier persona vulnerando sus derechos personales y derechos de propiedad intelectual [6]. Debido al potencial dañino que tiene DeepFake es importante tener una herramienta para detectar contenido multimedia generado por medio de DeepFake.

III. Estado del arte

El contenido generado a partir de los DeepFakes es cada vez más perjudicial para la privacidad, la seguridad de la sociedad e incluso la democracia [7]. Se han propuesto métodos para detectar DeepFakes desde el inicio de esta amenaza. Los primeros métodos se basaron en características evidentes obtenidas de inconsistencias del proceso de síntesis del video falso. Los métodos más recientes aplican el aprendizaje profundo para extraer características que revelan la presencia de alteraciones en un vídeo [8].

La detección de contenido generado a partir de DeepFake se ha considerado como un problema de clasificación binaria, estableciendo las clases de vídeos auténticos y manipulados. Los modelos basados en una clasificación binaria requieren una gran cantidad de datos en su fase de entrenamiento, estos datos están representados en datasets de videos reales y manipulados. El inicio de estos dataset fue dado por Korshunov y Marcel [9], quienes crearon un conjunto de datos de DeepFake que incluía 620 videos basados en el modelo GAN utilizando la herramienta de faceswap-GAN de código abierto [10]. Este conjunto de datos se alimentó de videos de la base de datos de VidTIMIT disponible de manera pública [11] para generar videos alterados de tipo DeepFake de baja y alta calidad, imitando expresiones faciales, movimientos de la boca y parpadeo de ojos.

Con estos videos disponibles se ponen a prueba varios métodos de detección de DeepFake. Los resultados de estas pruebas muestran que populares sistemas de reconocimiento facial basados en VGG [12] y Facenet [13][14] son incapaces de diferenciar rostros generados por DeepFake de un rostro real.

Así mismo otros métodos, como los de enfoques de sincronización de labios [15][16] y las métricas de calidad de imagen con máquina de vectores de soporte (SVM)[17], producen tasas de error muy altas en la detección de videos DeepFake del conjunto de datos de Korshunov y Marcel.

En intentos más recientes, se encuentra el método basado en la observación de la coherencia temporal de un video de Sabir et al [18]. Este método realiza la comparación de características de las secuencias cuadro por cuadro de un video, identificando manipulaciones y así detectar DeepFakes. Se propuso un modelo convolucional recurrente (RCN) basado en la integración de la red convolucional DenseNet [19] y celdas unitarias recurrentes [20] y así identificar las diferencias temporales entre cuadros. (Figura 2). Este modelo fue probado con un nuevo dataset, FaceForensics++[21], el cual incluye 1000 videos y se obtiene una mejora comparado con los anteriores modelos.

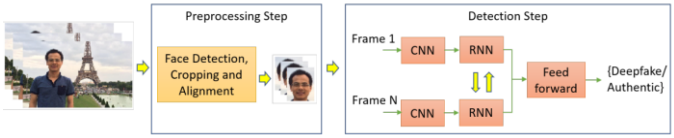

Figura 2. Método basado en la detección de modificaciones en una secuencia de fotogramas.[21]

IV. Descripción de la solución

Para desarrollar el algoritmo de detección de contenido multimedia generado por medio de DeepFake se utilizará un conjunto de datos proporcionado por el desafío DeepFake Detection Challenge disponible en Kaggle. El conjunto de datos consiste en 470 GB de videos en formato MP4 etiquetados como REAL o FAKE, en donde la categoría REAL consiste en videos reales y la categoría FAKE consiste en videos generados por medio de DeepFake. Los videos tienen una duración fija de 10 segundos y una resolución de 1920x1080 píxeles. Asimismo, Kaggle proporciona un archivo de metadatos que contiene el nombre del archivo, la etiqueta (REAL o FAKE) y la distribución (train o test).

Debido al peso elevado de los datos, Kaggle dividió el conjunto de datos en pequeños Chunks que tienen un peso aproximado de 10 GB. Para construir el dataset definitivo que fue utilizado en el desarrollo de los diferentes modelos de aprendizaje se utilizaron 5 Chunks de datos: el Chunk 0, Chunk 2, Chunk 4, Chunk 6 y Chunk 8 con un peso total de 49.3 GB.

Del mismo modo, debido al peso elevado de los datos, se utilizó una plataforma On-Cloud para el entrenamiento y validación de los diferentes modelos construidos. En específico, se utilizó un clúster en la herramienta AI Platform de Google Cloud Platform (GCP). Dicho clúster contaba con 64 GB de RAM, 8 CPUS y una GPU Tesla T4.

El desarrollo del proyecto se dividió en 4 etapas: la etapa de balanceo de datos, la etapa de extracción de características, la etapa de entrenamiento y validación de los modelos y la etapa de despliegue del modelo en una interfaz web.

A. Etapa de balanceo de datos

Como se describió anteriormente, el dataset proporcionado por Kaggle cuenta con un alto desbalance de las clases, por ende, para no inducir *Sampling Bias* al modelo fue necesario incluir una etapa de balanceo de los datos. Por otro lado, debido al límite de recursos para el procesamiento de los datos, no fue posible trabajar con el dataset completo y fue necesario extraer una pequeña muestra de los datos correspondiente a 2400 videos, 1200 videos para la clase Real y 1200 videos para la clase Fake.

Por otro lado, para extraer los videos de las diferentes clases se seleccionaron 5 Chunks de datos del dataset completo proporcionado por Kaggle. Asimismo, se tomaron todos los videos pertenecientes a la clase Real de estos Chunks y se hizo un muestreo aleatorio para extraer la misma cantidad de videos para la clase Fake.

B. Etapa de extracción de características

Una vez se seleccionaron los videos pertenecientes al conjunto de datos definitivo, era necesario extraer las características relevantes para entrenar un modelo de aprendizaje. Cada video tiene una duración de 10 s y una resolución de 1920x1080 píxeles. Asimismo, cada video tiene en promedio 300 frames y tres canales de color (RGB) por frame. Para reducir el número de características, se seleccionaron 30 frames espaciados linealmente dentro del conjunto de 300 frames, con el fin de obtener una muestra representativa de los datos.

Por otra parte, debido a que los DeepFakes afectan principalmente el rostro de la persona que aparece en el video, era necesario resaltar esa región de los frames. Por ende, se utilizó un modelo pre-entrenado para la extracción de rostros para los diferentes frames escogidos del video. El modelo pre-entrenado utilizado para la extracción de los rostros fue MTCNN (Multi-Task Cascaded Convolutional Networks) debido a que presentó buen rendimiento en cuanto a tiempo de ejecución comparado con librerías de Python enfocadas para dicha tarea.

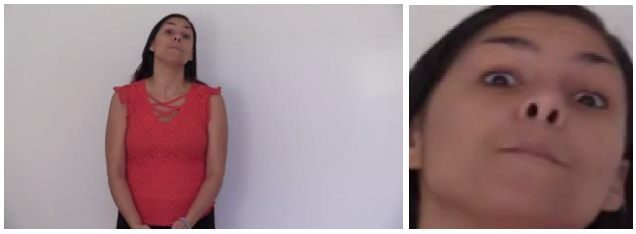

Figura 3. (I) Frame extraído del video. (D) Rostro extraido a partir del frame usando MTCNN

En la figura 3 se pueden observar las imágenes resultantes luego de la extracción de características de un video de muestra. En la imagen izquiera se puede observar un Frame extraído a partir del video. Por otro lado, en la imagen derecha se puede observar la extracción del rostro a partir del frame proporcionado.

Para cada video del conjunto de datos, se construyó un vector con las imágenes de los rostros extraídos a partir de los 30 frames espaciados linealmente. Este vector de imágenes fue utilizado como insumo para el entrenamiento y validación de los modelos construidos.

C. Etapa de entrenamiento y validación de los modelos

En la etapa de entrenamiento y validación se construyeron 3 modelos, un modelo baseline, un modelo a partir de la red pre-entrenada InceptionV3 y un modelo a partir de la red pre-entrenada VGG16.

Modelo Baseline:

El modelo Baseline se construyó a partir de una red Convolucional simple. Lo anterior con el fin de establecer un punto de comparación a partir del cual proponer modelos más complejos y con mejores resultados. La estructura del modelo Baseline se observa en la siguiente tabla.

Tabla 1. Estructura del modelo Baseline.

Modelo Baseline				
Layer	**Type**	**Neuronas**	**Padding**	**Activación**
Capa convolucional 1	Conv2D	128	same	ReLU
Capa maxpooling 1	MaxPooling2D			
Capa convolucional 2	Conv2D	16	same	ReLU
Capa maxpooling 2	MaxPooling2D			
Capa convolucional 3	Conv2D	8	same	ReLU
Capa de aplanamiento	Flatten			
Capa Densa 1	Dense	50		ReLU
Capa Densa 2	Dense	30		ReLU
Capa Salida	Dense	2		Sigmoid

Modelo a partir de la Red Pre-entrenada InceptionV3:

A partir de los resultados del modelo Baseline, se pudo observar que los mapas de características generados por este modelo no eran suficientes como para aprender a clasificar los videos. Es por eso que se escogió la red pre-entrenada InceptionV3 debido a que tiene una estructura más compleja y, por ende, es capaz de captar patrones en las imágenes mucho más complejos, como por ejemplo los generados por medio de DeepFakes. La red pre-entrenada InceptionV3 cuenta con una serie de capas convolucionales, capas de Pooling, capas de Dropout y Concatenación. Para este caso, se eliminó la capa final, debido a que esta capa debe ser construida de acuerdo a los requerimientos del problema a solucionar. Para la etapa de clasificación, se construyó un Perceptrón Multicapa con una estructura muy simple. La estructura del perceptrón Multicapa puede observarse en la tabla 2.

Tabla 2. Estructura del perceptrón Multicapa utilizado para la clasificación.

Perceptrón Multicapa				
Layer	**Type**	**Neuronas**	**Activación**	**Dropout**
Capa Global Average Pooling 1	GlobalAveragePooling2D			
Capa Densa 1	Dense	512	ReLU	0,1
Capa Densa 2	Dense	256	ReLU	0,1
Capa de Salida	Dense	2	Sigmoid	

La estructura definitiva del modelo se observa a continuación.

Figura 4. Estructura del modelo construido a partir de la red pre-entrenada InceptionV3 y el Perceptrón Multicapa.

Modelo a partir de la Red Pre-entrenada VGG16:

Este modelo fue construido a partir de la red pre-entrenada VGG16, y al igual que el modelo anterior, se agregó un Perceptrón Multicapa para la clasificación de las características generadas a partir de la red pre-entrenada.

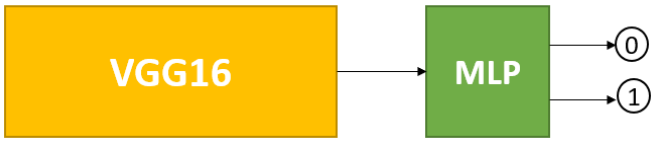

Figura 5. Estructura del modelo construido a partir de la red pre-entrenada VGG16 y el Perceptrón Multicapa.

Para entrenar los diferentes modelos se utilizó una distribución para los conjuntos de entrenamiento y prueba de 0.5. Lo anterior debido a que fue la mayor cantidad de datos posibles antes de que se generará un error de memoria en el clúster de GCP.

Del mismo modo, la distribución entre el conjunto de entrenamiento y validación fue de 0.2. A continuación se observa en detalle la distribución de los datos para el conjunto de entrenamiento, validación y prueba.

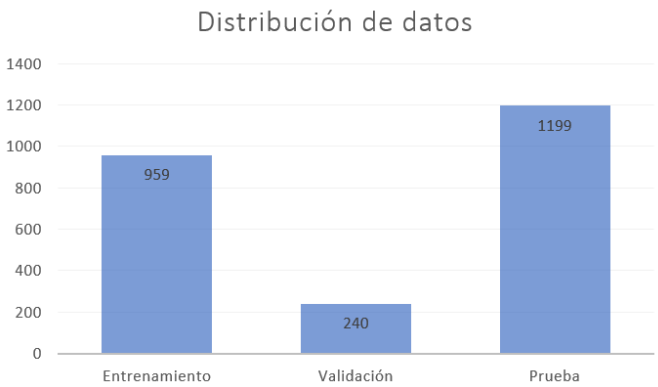

Figura 6. Distribución de los conjuntos de entrenamiento, validación y prueba.

D. Etapa de despliegue del modelo

Para dar paso a la interacción de usuarios externos con el algoritmo y el modelo con mejor desempeño a la hora de clasificar los videos se realizó una aplicación en Python mediante el *framework* Flask. Este *framework* se parametrizó para que funcione a partir de un modelo cargado, con el fin de que sea escalable y pueda ser cambiado más adelante si se obtiene un modelo con mejores resultados. Esta aplicación permite al usuario subir o cargar un video y procesarlo para indicar si los *frames* del video son reales o generados por medio de DeepFake.

Los resultados generados por el modelo se pueden apreciar mediante la etiqueta *Real* y *DeepFake*. A continuación, se muestra un ejemplo del caso de uso con un video ya cargado (parte superior) y su respectivo resultado (parte inferior).

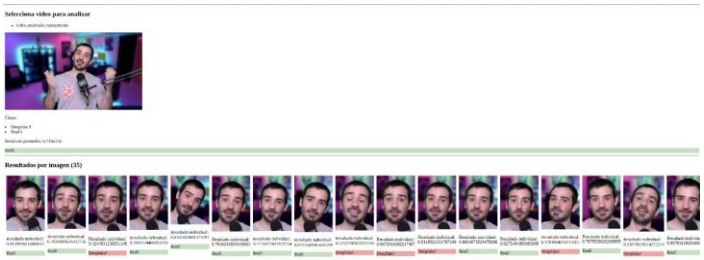

Figura 5. Aplicación construida para la detección de videos generados por medio de DeepFake.

V. RESULTADOS

Luego de entrenar los diferentes modelos se obtuvo los siguientes resultados:

Tabla 3. *Valor de Precisión para los diferentes modelos.*

Modelo	Clase	Precisión
Modelo Baseline	0	0,69
	1	0,66
Modelo Red Pre-entrenada InceptionV3 + MLP	0	0,80
	1	0,85
Modelo Red Pre-entrenada VGG16 + MLP	0	0,85
	1	0,82

En la tabla 3 se puede observar los valores de precisión para los diferentes modelos y las diferentes clases. En este problema la clase de interés es la clase FAKE, por ende, se busca maximizar las estadísticas para esta clase. El modelo que obtuvo mejores resultados de Precisión para la clase de interés fue el modelo basado en la red pre-entrenada InceptionV3.

Tabla 4. *Valor de RECALL para los diferentes modelos.*

Modelo	Clase	RECALL
Modelo Baseline	0	0,53
	1	0,80
Modelo Red Pre-entrenada InceptionV3 + MLP	0	0,83
	1	0,82
Modelo Red Pre-entrenada VGG16 + MLP	0	0,78
	1	0,88

En la tabla 4 se puede observar los valores de RECALL para los diferentes modelos construidos. El modelo que maximiza este valor para la clase de interés fue el modelo basado en la red pre-entrenada VGG16.

Tabla 5. Valor de F1-Score para los diferentes modelos.

Modelo	Clase	F1-Score
Modelo Baseline	0	0,60
	1	0,72
Modelo Red Pre-entrenada InceptionV3 + MLP	0	0,82
	1	0,84
Modelo Red Pre-entrenada VGG16 + MLP	0	0,81
	1	0,85

En la tabla 5 se puede observar los valores de F1-Score para los diferentes modelos construidos. El modelo que maximiza este valor para la clase de interés fue el modelo basado en la red pre-entrenada VGG16.

Con base en los resultados anteriores, y con base en el contexto del problema, se desea que el modelo escogido tenga el mayor valor posible en la métrica RECALL, ya que es más importante predecir la mayor cantidad de datos como FAKE, inclusive si algunos de estos videos no pertenecen a la categoría FAKE. Lo anterior debido a que clasificar un video como REAL y que su verdadera etiqueta sea FAKE podría afectar la imagen pública de alguna persona o involucrarlo en algún escándalo. Es por eso que la métrica más importante a la hora de escoger el mejor modelo es el RECALL.

Por otro lado, se observó que una Red Neuronal Convolucional simple como la utilizada para construir el Modelo Baseline no es capaz de clasificar con alta exactitud los videos modificados por medio de DeepFake.

Con base en lo anterior, el modelo que obtuvo mejores resultados y con el cual se construirá la aplicación para la interacción con los usuarios finales es el modelo basado en la red pre-entrenada VGG16.

VI. CONCLUSIONES

Se observó que las redes pre-entrenadas ofrecen un rendimiento elevado a la hora de extraer características complicadas como las que se generan por medio de DeepFake. Incluso para un ser humano es difícil concluir sobre la veracidad de un video modificado por medio de DeepFake, es por eso que con una Red Neuronal Convolucional simple como la construida para el Modelo Baseline fue incapaz de obtener buenos resultados a la hora de clasificar los videos. Por otro lado, la cantidad de datos utilizada para entrenar los diferentes modelos no fue la más adecuada debido al peso computacional que genera el procesamiento y la creación de los diferentes modelos. Si se tuvieran recursos adecuados para utilizar gran parte del dataset proporcionado por el challenge de Kaggle, se hubieran obtenido mejores resultados comparado con los obtenidos en este proyecto. Asimismo, para aumentar el desempeño de los diferentes modelos es necesario buscar una forma de procesar los videos en donde se detectan mas de un rostro, ya que el DeepFake únicamente es aplicado a un rostro por video. Por ende, al utilizar un rostro que no fue procesado por DeepFake y que tiene etiqueta FAKE, se obtendrían características erróneas y el modelo tendría *Bias*. Por último, se comprobó que por medio de modelos de Machine Learning y Deep Learning es posible construir una aplicación precisa para la detección de DeepFakes, lo cual puede ser de gran utilidad en la comprobación de contenido compartido en redes sociales o en internet.

VII. REFERENCIAS

[1] Statista. (2019, April 17). Infografía: A la espera de un big bang de datos. Statista Infografías. Retrieved October 7, 2021, from https://es.statista.com/grafico/17734/cantidad-real-y-prevista-de-datos-generados-en-todo-el-mundo/.

[2] Ipsos. (2019, June 11). Fake News: A Global Epidemic Vast Majority (86%) of Online Global Citizens Have Been Exposed to it. Ipsos. Retrieved October 7, 2021, from https://www.ipsos.com/en-us/news-polls/cigi-fake-news-global-epidemic.

[3] Westerlund, M. (2019). The emergence of deepfake technology: A review. Technology Innovation Management Review.

[4] Masood, M., Nawaz, M., Malik, K. M., Javed, A., & Irtaza, A. (2021). Deepfakes Generation and Detection: State-of-the-art, open challenges, countermeasures, and way forward. arXiv preprint arXiv:2103.00484.

[5] Colaner, N. & Quinn, .M.J (2020). Deepfakes and the Value-Neutrality Thesis Retrieved October 7, 2021 from https://www.seattleu.edu/ethics-and-technology/viewpoints/deepfakes-and-the-value-neutrality-thesis.html

[6] Colak, B. (2021). Legal Issues of DeepFakes. Retrieved October 7, 2021 from https://www.internetjustsociety.org/legal-issues-of-deepfakes

[7] Chesney, R., and Citron, D. K. (2018). Deep fakes: a looming challenge for privacy, democracy, and national security. https://dx.doi.org/10.2139/ssrn.3213954.

[8] De Lima, O., Franklin, S., Basu, S., Karwoski, B., and George, A. (2020). Deepfake detection using spatiotemporal convolutional networks. arXiv preprint arXiv:2006.14749.

[9] Korshunov, P., and Marcel, S. (2019). Vulnerability assessment and detection of deepfake videos. In The 12th IAPR International Conference on Biometrics (ICB), pp. 1-6.

[10] Faceswap-GAN. Available at https://github.com/shaoanlu/faceswap-GAN.

[11] VidTIMIT database. Available at http://conradsanderson.id.au/vidtimit/

[12] Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015, September). Deep face recognition. In Proceedings of the British Machine Vision Conference (BMVC) (pp. 41.1-41.12).

[13] FaceNet. Available at https://github.com/davidsandberg/facenet.

[14] Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 815-823).

[15] Chung, J. S., Senior, A., Vinyals, O., and Zisserman, A. (2017, July). Lip reading sentences in the wild. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 3444-3453).

[16] Korshunov, P., and Marcel, S. (2018, September). Speaker inconsistency detection in tampered video. In 2018 26th European Signal Processing Conference (EUSIPCO) (pp. 2375-2379). IEEE.

[17] Galbally, J., and Marcel, S. (2014, August). Face anti-spoofing based on general image quality assessment. In 2014 22nd International Conference on Pattern Recognition (pp. 1173- 1178). IEEE.

[18] [18] Sabir, E., Cheng, J., Jaiswal, A., AbdAlmageed, W., Masi, I., and Natarajan, P. (2019). Recurrent convolutional strategies for face manipulation detection in videos. In Proceedings of the IEEE Conference on Comp

[19] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4700-4708).

[20] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014, October). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1724-1734)

[21] Hsu, C. C., Lee, C. Y., and Zhuang, Y. X. (2018, December). Learning to detect fake face images in the wild. In 2018 International Symposium on Computer, Consumer and Control (IS3C) (pp. 388-391). IEEE.