Detección de recapturas de pantalla en *onboarding* digital: estrategias para entornos de datos restringidos

Ángela Barriga Rodríguez
Mobbeel Solutions SL
Cáceres, España
abarriga@mobbeel.com

Álvaro Hernández Martín
Mobbeel Solutions SL
Cáceres, España
ahernandez@mobbeel.com

Belén González Sánchez
Mobbeel Solutions SL
Cáceres, España
bgonzalez@mobbeel.com

Javier Nieto Acero
Mobbeel Solutions SL
Cáceres, España
jnieto@mobbeel.com

Daniel Santos Anés
Mobbeel Solutions SL
Cáceres, España
dsantos@mobbeel.com

***Resumen*—Con la creciente expansión de la transformación digital, la importancia de garantizar la seguridad en los procesos de generación de identidad digital ha aumentado significativamente. En este artículo, abordamos la verificación de documentos de identidad en el contexto de las soluciones de registro remoto de clientes (*onboarding* digital). En concreto, nos centramos en el desafío de entrenar modelos de aprendizaje automático para detectar ataques de recaptura de pantalla en entornos de datos restringidos. Este escenario surge debido a la necesidad de cumplir con la privacidad y las políticas de protección de datos en documentos pertenecientes a individuos reales. Para afrontar esta limitación, presentamos un flujo de trabajo destinado a la creación de bases de datos de imágenes de documentos que simulan ser capturas de pantalla a partir de especímenes disponibles públicamente. Esta propuesta involucra el empleo de técnicas de transferencias de texturas, así como la introducción de una métrica para evaluar la probabilidad de que una imagen presente trazas de recaptura, con el objetivo de optimizar el proceso de generación de datos sintéticos. Asimismo, exploramos el impacto de la dimensión temporal en los datos, generando secuencias de *frames* de vídeo que imitan el movimiento de artefactos de pantalla. Finalmente, evaluamos nuestra propuesta en un entorno real, demostrando su capacidad de generalización. Los resultados obtenidos muestran que nuestra propuesta mejora la capacidad de los modelos de clasificación para distinguir entre imágenes reales y recapturas.**

***Index Terms*—verificación de documentos, recapturas de pantalla, clasificación, aprendizaje profundo, aprendizaje automático, visión por computador, *onboarding* digital**

Tipo de contribución: *Transferencia*

I. Introducción

En los últimos años ha surgido un creciente interés por parte de empresas y organismos públicos en la digitalización de sus procesos. Esta transformación digital se ha acelerado aún más a partir de la pandemia producida por el SARS-CoV-2 [1], haciendo que incluso operativas clave puedan efectuarse sin necesidad de desplazamientos.

En este nuevo entorno es fundamental desarrollar tecnologías que permitan certificar la identidad de las personas de forma segura. El concepto de identidad digital abarca los datos que representan a una persona en internet. Certificar la autenticidad de estas identidades implica vincular la identidad real o analógica con la digital, un proceso facilitado por las soluciones de registro remoto de clientes (también conocidas como *onboarding* digital). Estos sistemas permiten generar una identidad digital fácilmente a partir de documentos de identidad oficiales y tecnologías de biometría.

Las soluciones de *onboarding* digital son cada vez más comunes en sectores como el financiero, seguros, juegos en línea y comercio digital. Además, se utilizan también para operaciones altamente sensibles como la emisión de certificados cualificados [2]. Esta proliferación plantea el desafío de garantizar un nivel de seguridad equiparable a la presencia física, lo que representa un importante desafío en la gestión y detección de fraudes.

Un aspecto crítico en términos de seguridad es el análisis del documento de identidad utilizado durante el proceso de registro. Los sistemas de *onboarding* deben ser capaces de identificar cualquier intento de fraude, como el uso de documentos fotocopiados, alterados física o digitalmente, o la presentación de una imagen del documento en una pantalla en lugar del original (en adelante recapturas de pantalla). Este último caso es especialmente preocupante, ya que constituye un vector de ataque que puede llevarse a cabo con facilidad y tener consecuencias potencialmente perjudiciales si el suplantador logra obtener una copia de la imagen del documento de identidad de otra persona.

Los recientes avances en el ámbito de la visión por computador, impulsados por el aprendizaje profundo, han dado lugar al desarrollo de sistemas de clasificación automática de imágenes con alta precisión. Sin embargo, para entrenar y generalizar estos modelos a diversos entornos, se requiere un extenso conjunto de datos. En el caso de los documentos de identidad, la ausencia de bases de datos públicas debido a consideraciones obvias de privacidad y protección de datos [3] representa un desafío inicial. Además, la obtención de muestras de imágenes de recapturas es un proceso costoso, especialmente para garantizar la generalización del sistema a diferentes entornos.

Este es el contexto que impulsa nuestra investigación. En este artículo, nos enfocamos en la detección de recapturas de

ISBN:978-84-09-62140-8

pantalla en un entorno de datos limitado, específicamente en documentos de identidad dentro del marco de soluciones de *onboarding* digital. Nuestra investigación se centra en desarrollar un marco de trabajo innovador para crear una base de datos de muestras de recapturas sintéticas, utilizando ejemplos de documentos de diferentes países (especímenes o sintéticos) evitando de esta manera el uso de documentos pertenecientes a personas reales. Hemos empleado diversas técnicas de visión por computador para generar estas recapturas aplicando texturas, buscando replicar los artefactos y aberraciones comunes en las pantallas. Para aumentar la calidad de las recapturas generadas, proponemos una métrica con la que medir cómo de probable es que una imagen se corresponda con una recaptura. Hemos utilizado esta métrica para filtrar las texturas aplicadas. Además, hemos ampliado el alcance del sistema para incluir secuencias de vídeo, aprovechando la dimensión temporal para mejorar la distinción entre imágenes reales y recapturas. Finalmente, hemos utilizado esta base de datos para entrenar varios modelos de clasificación, evaluándolos con un conjunto de datos reducido de documentos reales para determinar su capacidad de generalización a un entorno real.

El artículo sigue la siguiente estructura: la Sección II ahonda en el concepto de recapturas de pantalla y analiza cuál es el estado del arte respecto a bases de datos que incluyen este tipo de ataques. A continuación, en la Sección III, presentamos nuestro flujo de trabajo para generar una base de datos que incluya imágenes de documentos reales y recapturas de pantalla sintéticas. Tras ello, en la Sección IV presentamos una evaluación en la que entrenamos y evaluamos varios modelos de clasificación con dos bases de datos, con y sin dimensión temporal, utilizando diferentes modelos de clasificación. Siguiendo en esta línea, la Sección V presenta una evaluación con datos capturados en un entorno real, demostrando la transferencia de nuestro estudio. Finalmente, la Sección VI enumera las conclusiones de nuestro trabajo y presenta líneas de trabajo futuro.

II. Recapturas de pantalla

En esta sección ahondamos en el concepto de ataque mediante recaptura de pantalla y analizamos las bases de datos existentes para trabajar en la detección de este tipo de fraude.

Los ataques de recaptura de pantalla consisten en la presentación [4] de una pantalla con la imagen de un documento de identidad para engañar a un sistema de verificación. El objetivo del atacante es hacer pasar la imagen presentada como un documento de identidad legítimo, con el fin de suplantar la identidad de otra persona.

La detección de este tipo de ataques es un reto debido a la variedad de dispositivos y pantallas que pueden utilizarse tanto para mostrar el documento de identidad como para capturarlo. Además, dado que los sistemas de verificación de identidad están hechos para su uso en entornos cotidianos, las imágenes de los documentos pueden presentar condiciones lumínicas muy variables, así como alteraciones en la calidad de la imagen dependiendo de la pericia del usuario final cuando hace uso de la tecnología.

En paralelo a estos problemas, el desarrollo de sistemas de detección automática de fraude se complica por la necesidad de disponer de bases de datos con una cantidad suficiente de imágenes para llevar a cabo un entrenamiento y una evaluación efectivas [5], [6].

Así, se han creado bases de datos como la presentada por Polevoy et al. en [6], incluyendo documentos de identidad y pasaportes de diferentes países. Consta de más de mil secuencias de vídeo capturadas en una amplia gama de condiciones lumínicas, diversidad de fondos, ángulos, etc. Los vídeos contienen diferentes tipos de ataques con documentos, incluyendo recapturas de pantalla. Esta base de datos se ha construido a partir de documentos generados sintéticamente. Tras crear impresiones de alta calidad de los documentos se ha procedido a crear sus recapturas a mano, mostrándolos en pantallas de escritorio y de portátiles LCD y grabándolos con diferentes modelos de móviles.

Otra base de datos disponible públicamente es la descrita en [7] por Chen et al. En este caso, se ha generado sintéticamente un conjunto de documentos que imitan a tarjetas de estudiantes de diversos organismos universitarios de China. A partir del conjunto sintético se han creado diferentes ataques a mano, incluyendo recapturas de pantalla, utilizando diversos dispositivos tanto para la captura como la muestra. En total, la base de datos incluye más de 2000 imágenes. Esta base de datos se caracteriza por incluir imágenes de muy alta calidad.

En el momento de escribir este artículo, no hemos podido encontrar más bases de datos disponibles públicamente que incluyesen muestras de recaptura de pantallas. Ambas bases de datos presentan problemas similares. En primer lugar, el número de muestras de ataques de recaptura de pantalla es limitado y, dependiendo de la arquitectura de aprendizaje automático a entrenar, no presentan un número de muestras suficientes para llevar a cabo un entrenamiento exitoso. Este número de muestras bajo se debe, en gran medida, al hecho de que los ataques se han creado a mano uno a uno, convirtiéndose en una tarea costosa en términos de tiempo.

En segundo lugar, para lograr una buena tasa de acierto en la detección de fraude en entornos reales, es necesario que el sistema de clasificación tenga suficiente capacidad de generalización. Es decir, ha de ser capaz de clasificar correctamente documentos que no haya visto antes en su entrenamiento, por ejemplo: nuevas versiones o de diferentes países. Para alcanzar una buena generalización, en un escenario de entrenamiento ideal, se usarían muestras del mayor número posible de tipos diferentes de documentos, incluyendo imágenes reales y ataques, a los que el clasificador fuera a enfrentarse.

Otra perspectiva posible consiste en reentrenar un sistema de clasificación especializado en ciertos tipos de documentos para ser capaz de clasificar correctamente otros distintos, en lo que se conoce como adaptación de dominio. En este escenarios necesitaríamos un número de muestras suficientes del nuevo tipo o tipos de documentos.

Así, en [6] se incluyen muestras sintéticas a partir de documentos de identificación de 5 países (Albania, España, Estonia, Finlandia y Serbia) y de pasaportes de otros 5 países (Azerbayán, Grecia, Letonia, Rusia y Eslovaquia). Mientras que en [7] solo aparecen muestras sintéticas de tarjetas universitarias, todas de China. La variedad que presentan ambas bases de datos resulta insuficiente a la hora de enfrentar un escenario real de verificación de identidad.

Para solventar algunos de estos problemas, podemos en-

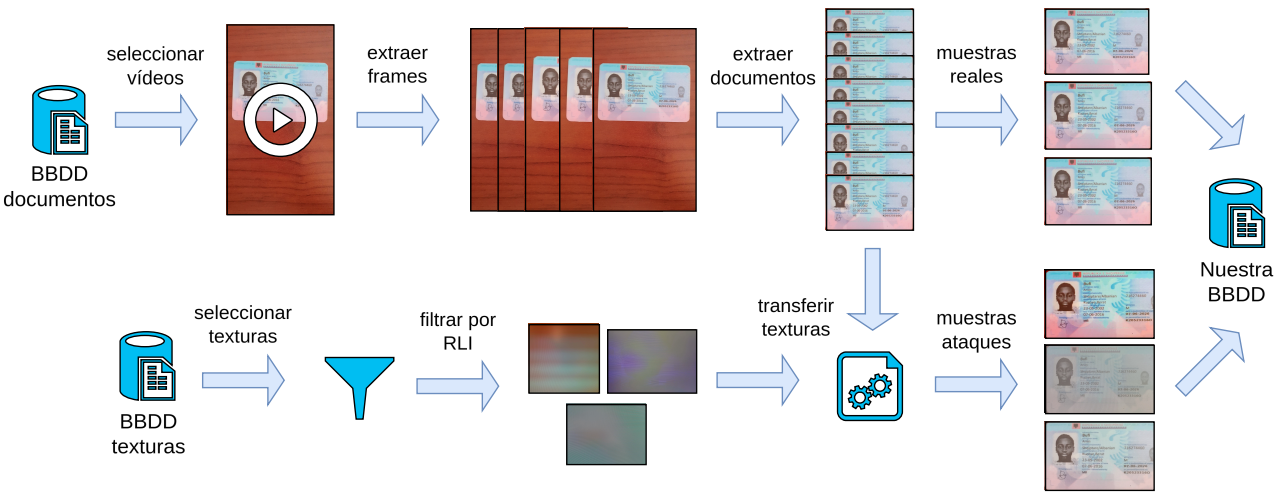

Figura 1: Flujo de preprocesado y creación de nuestra base de datos.

contrar trabajos comos los presentados en [5], [8], donde se proponen técnicas de generación de recapturas de documentos a partir de texturas y artefactos que imitan a pantallas, papel, etc. Ambos trabajos utilizan redes adversarias generativas (*Generative Adversarial Network*, GAN) y técnicas de modificación de imágenes para transferir texturas. Estos trabajos se centran en verificar la calidad de las recapturas creadas. Para ello, calculan la distancia de Fréchet de incepción (*Fréchet Inception Index*, FID [9]) y comprueban la calidad de la composición de las recapturas mediante su clasificación con arquitecturas de aprendizaje automático. Asimismo, presentan estudios sobre cómo el uso de estos datos generados sintéticamente podría afectar a la detección de ataques. Ambos estudios concluyen que, pese a que para recapturas en papel la detección de ataques mejora, hay un ligero empeoramiento a la hora de la detección de recapturas de pantalla. Sin embargo, el objetivo de ambos trabajos se encuentra enfocado a la generación de datos sintéticos y a estudiar su calidad, y no a realizar un estudio sobre el impacto del uso de este tipo de datos en la detección de ataques.

Como respuesta a los problemas que presentan las bases de datos disponibles públicamente [6], [7] y con el objetivo de realizar un estudio sobre el impacto que tiene la adición de artefactos de pantalla en vídeos en lugar de en imágenes independientes, en la siguiente sección proponemos nuestra solución para generar bases de datos que facilite la detección de recapturas de pantalla en entornos de datos restringidos.

III. Creación de la base de datos

En esta sección, describimos el proceso que hemos seguido para construir nuestra base de datos. Iniciamos el proceso con una base de datos preexistente formada por documentos sintéticos. Posteriormente, empleamos técnicas de transferencia de texturas para generar muestras que simulan ataques de recaptura de pantalla. La Fig. 1 presenta un resumen del flujo de trabajo realizado para obtener las recapturas de pantalla. Para conluir la sección analizamos las texturas utilizadas para la generación de imágenes, así como los resultados.

III-A. *Fuente y preprocesado*

Como punto de partida, hemos seleccionado la base de datos ofrecida por Bulatovich et al. en [10], llamada MIDV-2020. Esta base de datos cuenta con documentos de identidad y pasaportes generados sintéticamente, de un total de 10 países, correspondientes con los enumerados en la sección previa al presentar la base de datos [6].

MIDV-2020 contiene vídeos de los documentos y pasaportes, sus respectivos *frames*, fotos, escaneos y sus imágenes originales en alta calidad. Dado que nuestro objetivo es la creación de una base de datos con ataques en vídeo, procedemos a trabajar con las muestras de vídeo.

III-B. *Creación de recapturas*

A continuación, utilizamos la técnica de generación de imágenes mediante transferencia de texturas presentada en [8]. En esta técnica, los autores proponen una biblioteca de imágenes de texturas de papel y pantalla que se pueden transferir para crear imágenes de ataques sintéticos, mediante la combinación de varias imágenes modificando su opacidad.

De las 5000 texturas propuestas en [8], seleccionamos las que crean artefactos de pantalla más notables en las imágenes de los documentos, obteniendo 200 texturas. En la siguiente subsección analizaremos más a fondo el criterio de selección de texturas. La razón tras esta selección viene por la propia naturaleza de los documentos y sus elementos de seguridad: surcos, texturas 3D y hologramas. Si los artefactos de la textura a aplicar son relativamente débiles o poco visibles en la imagen, al combinarse con los elementos de seguridad del documento pasarán desapercibidos, especialmente si incide sobre ellos la luz, dificultando la diferenciación entre ataque y documento real y por ende teniendo un impacto negativo en el entrenamiento de modelos de clasificación.

En nuestro caso, con el objetivo de aplicar las texturas a vídeos, por cada conjunto de *frames* pertenecientes al mismo vídeo, utilizamos una única textura a la que añadimos modificaciones de movimiento, opacidad, saturación y contraste, con el fin de simular las alteraciones reales que sufren los artefactos de pantalla al ser grabados en vídeo.

En concreto el movimiento se aplica siguiendo:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & \text{dx} \\ 0 & 1 & \text{dy} \end{bmatrix} \quad (1)$$

Donde $\mathbf{T}$ es una matriz de transformación que simula el efecto de desplazamiento en una textura de pantalla debido a la tasa de refresco de la pantalla. Los valores dx y dy representan los desplazamientos horizontales y verticales, respectivamente, y son variables que toman valores aleatorios dentro del rango $[0, 20]$ para dx y $[0, 10]$ para dy.

La opacidad de la textura aplicada varía dinámicamente en función de la diferencia entre *frames* consecutivos. Esta variación se calcula como la diferencia absoluta media μ entre *frames* consecutivos normalizada en el intervalo $[0, 1]$. Así, la opacidad de la textura α viene determinada por:

$$\alpha = \text{mín}\left(\text{máx}\left(0{,}2, \frac{\mu}{255}\right), 0{,}8\right) \quad (2)$$

Aquí, α asegura que la opacidad se mantiene dentro de un rango razonable para mantener la visibilidad del documento subyacente al añadir la textura. Los valores 0.2 y 0.8 aseguran que la textura sea perceptible sin obscurecer demasiado la imagen original.

Tras ello, realizamos una combinación lineal del documento con la textura modificada tras usar las operaciones anteriores. Para ello, se aplica:

$$\text{dst}(x, y) = \text{src1}(x, y) \times \alpha + \text{src2}(x, y) \times \beta + \gamma \quad (3)$$

Donde src1 sería el documento y src2 la textura modificada. β es 1 - α representando el peso de la imagen original y γ es 0, indicando que no se añade ningún escalar a la suma, ya que no nos interesa modificar el brillo de la textura.

Finalmente, se altera la saturación y el contraste de la imagen texturizada de manera aleatoria. La saturación y el contraste se ajustan por un factor muestreado uniformemente en el rango [0.5, 1.5]. Este rango se elige para proporcionar ajustes de saturación y contraste que mejoren la apariencia de las imágenes texturizadas sin alterarlas excesivamente.

Para eliminar información innecesaria, este proceso se realiza sobre los documentos recortados de los *frames* de cada vídeo (ver Fig. 1). Para ello, utilizamos un detector que localiza las cuatros esquinas de un documento en la imagen, y con ellas, se crea una nueva imagen que solo contiene el documento. MIDV-2020 contiene imágenes de documentos en diferentes ángulos y condiciones de luminosidad. Con el fin de simular un entorno real de lo que sería aceptable en un proceso de *onboarding* digital, recortamos solo aquellos documentos que presentan una inclinación frontal suficiente y las condiciones lumínicas necesarias para que los datos del documento sean legibles. Tras texturizar los documentos recortados, contamos con 70282 documentos, la mitad de los cuales se corresponden con muestras reales y la otra mitad con recapturas.

III-C. Análisis de texturas y resultados

Con el fin de asegurar que las imágenes sintéticas generadas son adecuadas para el proceso de entrenamiento posterior, procedemos a analizarlas para evaluar su calidad.

Para ello, estudiamos cómo se aprecia la textura de pantalla en las imágenes. Las texturas de pantalla pueden ocasionar una variedad de artefactos visuales, los cuales pueden incluir, entre otros, la pixelación, que se manifiesta como una pérdida de detalle y suavizado de los bordes de los objetos, especialmente cuando la imagen es ampliada o la resolución de la pantalla es baja. Además, las pantallas con baja profundidad de color o problemas de reproducción pueden generar bandas de color visibles. Asimismo, pueden surgir reflejos y brillos no deseados en pantallas brillantes o bajo ciertas condiciones lumínicas. Uno de los artefactos más destacados es el conocido como efecto Moiré [11]. Este se produce cuando dos patrones superpuestos interactúan, creando un tercer patrón irregular debido a la falta de alineación entre sus frecuencias espaciales. Esto precisamente es lo que ocurre al capturar la imagen de una pantalla a través de una cámara.

Para una comprensión más completa del efecto Moiré, es importante considerar el fenómeno del *aliasing* y la frecuencia de *Nyquist* [12]. Así, el *aliasing* se da cuando se muestrea una señal analógica a una frecuencia insuficiente, lo que lleva a una incorrecta representación de la señal original en la versión digitalizada. Esto es debido a que las frecuencias altas de la señal analógica no pueden ser correctamente capturadas por el proceso de muestreo, resultando en la aparición de frecuencias incorrectas en la señal digital.

La frecuencia de *Nyquist*, por otro lado, es el límite superior de frecuencia que puede ser representado de manera adecuada en una señal digital. Se corresponde con la mitad de la frecuencia de muestreo y representa la frecuencia máxima que puede ser capturada sin introducir *aliasing*.

El efecto Moiré, que puede ser considerado como un tipo de *aliasing* visual, se manifiesta cuando las frecuencias espaciales de los patrones superpuestos están por debajo de la frecuencia de *Nyquist*. Por lo tanto, podemos definir formalmente el efecto Moiré de la siguiente manera:

$$Moire = |f_1 - f_2| < f_{Nyquist} \quad (4)$$

Donde f_1 y f_2 representan las frecuencias espaciales de los patrones superpuestos, y $f_{Nyquist}$ es la frecuencia de *Nyquist*, es decir, la mitad de la frecuencia de muestreo. Cuando la diferencia entre f_1 y f_2 es menor que la frecuencia de *Nyquist*, se produce el efecto Moiré.

Para calcular cuán fuerte es el efecto Moiré en las imágenes, calculamos la transformada de Fourier (TF) [13] de cada imagen $I(x, y)$ (fórmula 5), filtrando sus componentes para quedarnos solo con aquellos de alta frecuencia (fórmula 6), que están asociados usualmente a la aparición del efecto Moiré [11]. Con la suma de la magnitud (fórmula 7) de los componentes filtrados obtenemos un número que utilizamos como puntuación de cuán fuerte es el efecto Moiré en la imagen (dividido entre mil para su mejor legibilidad). A esta puntuación la denominamos índice de similitud de recaptura (*Recapture Likeness Index*, RLI), asociando valores más altos con una mayor posibilidad de que una imagen se corresponda con una recaptura. Las siguientes fórmulas desglosan el cálculo del RLI:

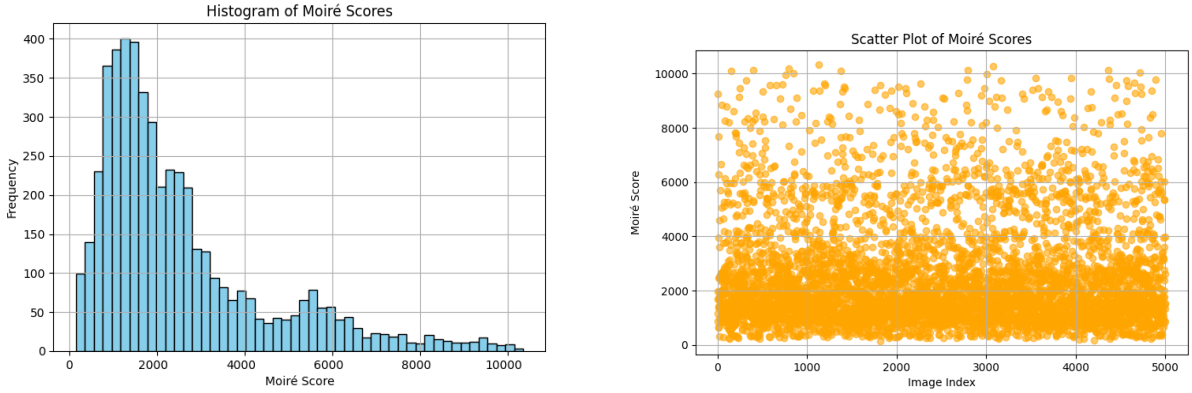

Figura 2: Histograma y diagrama de dispersión de las puntuaciones del efecto Moiré en texturas de pantalla.

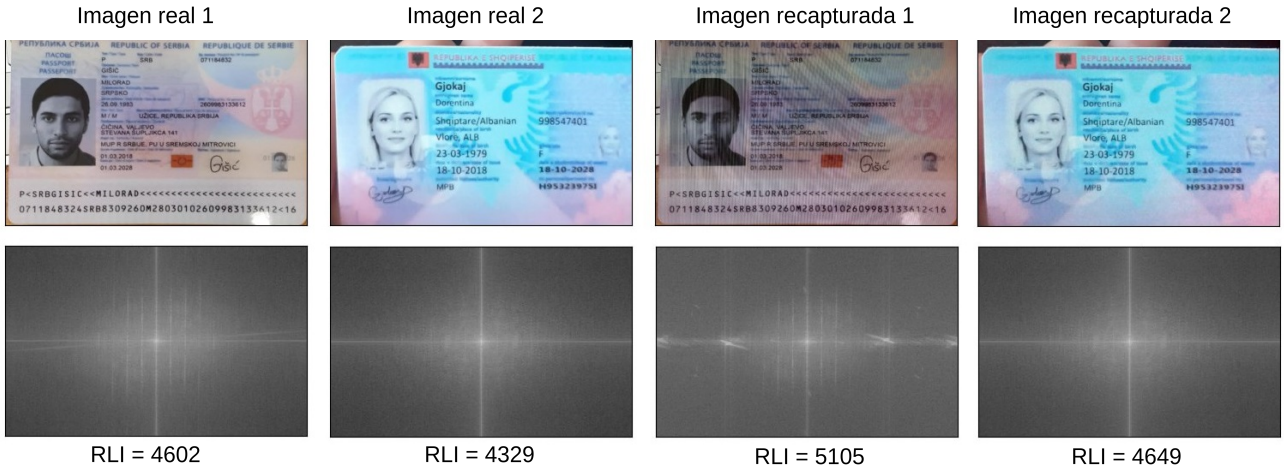

Figura 3: Comparativa entre imágenes reales y sus recapturas, incluyendo sus espectros de frecuencia.

$$\begin{aligned}
\mathrm{FFT}(I(x,y)) &= F(u,v) \\
&= \iint_{-\infty}^{\infty} I(x,y)\cdot e^{-2\pi i(ux+vy)}\,dx\,dy \qquad (5)\\
I_h(x,y) &= \mathrm{IFFT}(F_h(u,v)) \\
&= \iint_{-\infty}^{\infty} F_h(u,v)\cdot e^{2\pi i(ux+vy)}\,du\,dv \qquad (6)\\
M(u,v) &= \log(|F_h(u,v)|+1) \qquad (7)\\
\mathrm{RLI} &= \frac{1}{1000}\sum_{(u,v)\in \mathrm{ROI}} M(u,v) \qquad (8)
\end{aligned}$$

Así, utilizando el cálculo del RLI, hemos selecionado las texturas situadas en la mitad superior del primer decil, con el objetivo de utilizar texturas que darán lugar a recapturas de pantalla con un efecto Moiré muy marcado. En la Fig. 2 podemos observar como la mayoría de las texturas presentan puntuaciones más bajas, mientras que tan solo un pequeño grupo presenta puntuaciones altas. Las texturas excluidas en el proceso explicado en la Sección III presentan un RLI medio de 2197, mientras que las texturas seleccionadas presentan una media de 7230, confirmando así que las texturas seleccionadas presentan un efecto Moiré más marcado.

En la Fig. 3 podemos ver un ejemplo de dos imágenes de documentos reales y dos de las recapturas de pantalla generadas. Las imágenes reales 1 y 2 presentan un RLI de 4602 y 4329 respectivamente, mientras que sus análogas recapturadas puntuan con 5105 y 4649. Podemos ver cómo ambas recapturan presentan un RLI más alto a causa de la presencia del efecto Moiré, y que en concreto la imagen recapturada 2 presenta un índice menor que la imagen recapturada 1, ya que los artefactos de pantalla presentes en ella son menos visibles. En la parte inferior de la Fig. 3 presentamos las imágenes espectrales de la TF de estos mismos documentos con sus respectivas puntuaciones. En la figura se puede apreciar una diferencia más marcada entre las imágenes real y recapturada 1, mientras que la real y recapturada 2 reflejan una mayor similitud entre ellas.

IV. Evaluación

En esta sección procedemos a evaluar la base de datos que hemos generado en la sección anterior. Para ello, entrenamos y evaluamos varios modelos de aprendizaje automático. El objetivo es evaluar si, teniendo en cuenta la dimensión tem-

poral de los datos, se mejoran los resultados obtenidos con la propuesta de transferencia de texturas de [5], [8].

IV-A. Bases de datos

Para realizar la evaluación, además de la base de datos presentada en la Sección III (de ahora en adelante $BBDD_{seq}$), generamos otra base de datos en la que, para cada uno de los *frames* de vídeo aplicamos las texturas propuestas en [8] aleatoriamente, superponiéndolas a cada documento con diferentes niveles de opacidad ($BBDD_{ran}$).

En $BBDD_{seq}$ aplicamos las 200 texturas que presentan un RLI más alto, utilizando técnicas que imitan cómo se comportaría la textura de pantalla si estuviese siendo grabada en vídeo. Así, las texturas se aplican teniendo en cuenta los *frames* que corresponden a un mismo vídeo. Por otra parte, en $BBDD_{ran}$ aplicamos las 5000 texturas propuestas en [8] a los documentos, sin tener en cuenta qué *frames* corresponden al mismo vídeo y sin filtrar las texturas. Ambas bases de datos cuentan con el mismo número de documentos: 70282. De los cuales la mitad, es decir 35141 se corresponden con documentos reales y con recapturas respectivamente.

Conjunto de datos	**Cantidad**
Entrenamiento	50328
Test	5591
Evaluación	14365
Total	70282

Tabla I: Resumen de la distribución de datos.

IV-B. Arquitecturas

Seleccionamos dos arquitecturas para nuestra evaluación: una *MobileNetV2* como la utilizada en [5], [8] y una adaptación de la arquitectura *Sequencer2D* presentada en [14].

En el caso de la *MobileNetV2*, importamos la arquitectura *mobilenet_v2* proporcionada por *Torch Vision* en *Python*, al que le añadimos una capa lineal con la que realizar la clasificación binaria entre real y ataque. El tamaño de entrada a la red es de 224 x 224 píxeles, por lo que cada imagen de documento es transformada antes de entrar en ella.

Respecto a *Sequencer2D*, es una arquitectura basada en bloques de memoria a largo y corto plazo (*Long Short-Term Memory*, LSTM [15]) que permite el aprendizaje de dependencias temporales en datos secuenciales. *Sequencer2D* incluye LSTMs que extraen características de dos formas diferentes, vertical y horizontal, mejorando los resultados de la arquitectura. En nuestro caso, puede aprender a extraer características que diferencien con mayor eficacia un documento real de una recaptura de pantalla a través de los cambios de texturas de pantalla que se producen en los vídeos de los documentos. El tamaño de entrada a la red es idéntico al de *MobileNetV2*, 224 x 224 píxeles. De nuevo, añadimos una capa lineal con la que realizar la clasificación binaria.

IV-C. Entrenamiento

Cada arquitectura se entrena dos veces desde cero, una con $BBDD_{seq}$ y otra con $BBDD_{ran}$, obteniendo así cuatro modelos: $MobileNetV2_{seq}$, $MobileNetV2_{ran}$, $Sequencer2D_{seq}$ y $Sequencer2D_{ran}$.

Tal y cómo se presentó en la Sección II, la base de datos incluye documentos de identificación sintéticos de 5 países (Albania, España, Estonia, Finlandia y Serbia) y de pasaportes de otros 5 países (Azerbayán, Grecia, Letonia, Rusia y Eslovaquia). Para entrenar utilizamos todos los datos menos los correspondientes a los documentos de Eslovaquia y los pasaportes de Serbia, quedándonos así con con 55919 documentos. De los datos de entrenamiento, un 10 % se destina a probar (test) en tiempo real el proceso de entrenamiento. El grupo de datos de test se toma directamente del de entrenamiento aleatoriamente, sin tener en cuenta la distribución por países. Detalles de la distribución de datos se pueden ver en la Tabla I.

Todos los entrenamientos se llevan a cabo en Ubuntu 22.04.2 LTS, utilizando una *GeForce RTX* 3060, con *Python* 3.7, *Torch* 1.13.1 y *Torch Vision* 0.14.1. En todos los entrenamientos se utiliza una función pérdida de entropía cruzada (*CrossEntropyLoss*) y un optimizador *Adam* con un ratio de aprendizaje de 0.001.

$MobileNetV2_{seq}$ y $Sequencer2D_{seq}$ han sido entrenados durante 14 épocas, mientras que $Sequencer2D_{ran}$ durante 30 y $MobileNetV2_{ran}$ durante 21. El criterio para detener el entrenamiento es que los valores de pérdida y de exactitud del entrenamiento y los valores de test no mejorasen durante 5 épocas. En estas cifras se puede observar cómo al exponer a $Sequencer2D_{ran}$ a datos no secuenciales tarda mucho más en aprender de ellos (más del doble que su análogo entrenado secuencialmente).

IV-D. Resultados

Los cuatro modelos resultantes se evaluan con una parte de cada base de datos destinada a la evaluación. En este caso, utilizamos los documentos de Eslovaquia y los pasaportes de Serbia, destinando un total de 14365 imágenes. Cabe destacar que, pese a que tanto las imágenes de entrenamiento como las de evaluación proceden de la misma base de datos (MIDV-2020), los documentos utilizados para cada fase (entrenamiento y evaluación) son de diferentes países, y por ende presentan diferente aspecto visual. Así, estaríamos evitando obtener buenos resultados en la evaluación debido a un sobreajuste (*overfit*) del modelo.

A continuación, presentamos las métricas que hemos utilizado para evaluar los resultados: el APCER (*Attack Presentation Classification Error Rate*), el BPCER (*Bona-Fide Presentation Classification Error Rate*) y el ACER (*Average Classification Error Rate*).

El APCER representa la tasa de error de clasificación de presentaciones fraudulentas, es decir, la proporción de ataques que se clasifican incorrectamente como documentos reales (fraudes). Por otra parte, el BPCER representa la tasa de error de clasificación de presentaciones legítimas, es decir, la proporción de intentos de autenticación genuinos que se clasifican erróneamente como ataques (fallos). Para finalizar, el ACER es el promedio entre el APCER y el BPCER. Estas métricas se consideran mejores a menores valores.

Así, cada una de estas métricas se pueden calcular con las

$BBDD/Modelo$	$Sequencer2D_{seq}$	$MobileNetV2_{seq}$	$Sequencer2D_{ran}$	$MobileNetV2_{ran}$
$BBDD_{seq}$	**BPCER: 0.002** **APCER: 0.003** **ACER: 0.002** **ACC: 0.996**	BPCER: 0.006 APCER: 0.004 ACER: 0.005 ACC: 0.994	BPCER: 0.338 APCER: 0.462 ACER: 0.400 ACC: 0.599	BPCER: 0.125 APCER: 0.235 ACER: 0.180 ACC: 0.819
$BBDD_{ran}$	BPCER: 0.002 APCER: 0.979 ACER: 0.491 ACC: 0.508	BPCER: 0.006 APCER: 0.973 ACER: 0.490 ACC: 0.509	BPCER: 0.338 APCER: 0.024 ACER: 0.181 ACC: 0.818	BPCER: 0.125 APCER: 0.011 ACER: 0.068 ACC: 0.931

Tabla II: Resultados de evaluación.

$BBDD/Modelo$	$Sequencer2D_{seq}$	$MobileNetV2_{seq}$	$Sequencer2D_{ran}$	$MobileNetV2_{ran}$
Mobbeel	Recall: 0.859 F1: 0.795 ACC: 0.900	**Recall: 0.989** **F1: 0.947** **ACC: 0.902**	Recall: 0.501 F1: 0.472 ACC: 0.643	Recall: 0.469 F1: 0.306 ACC: 0.328

Tabla III: Resultados de evaluación de transferencia.

siguientes fórmulas:

$$\text{APCER} = \frac{\text{Número de fraudes exitosos}}{\text{Número total de ataques}} \tag{9}$$

$$\text{BPCER} = \frac{\text{Número de fallos}}{\text{Número total de documentos reales}} \tag{10}$$

$$\text{ACER} = \frac{\text{APCER} + \text{BPCER}}{2} \tag{11}$$

Asimismo, también utilizamos como métrica la exactitud del modelo (*Accuracy*, ACC) refleja la proporción de predicciones correctas que realiza un modelo de clasificación sobre el total de predicciones, utilizando la siguiente fórmula:

$$\text{ACC} = \frac{\text{Número de predicciones correctas}}{\text{Número total de predicciones}} \tag{12}$$

Los resultados obtenidos en la evaluación se pueden ver en la Tabla II. El modelo que obtiene mejores resultados en todas las métricas es $Sequencer2D_{seq}$ al ser entrenado y evaluado con $BBDD_{seq}$, seguido de $MobileNetV2_{seq}$. El siguiente modelo con mejores métricas es $MobileNetV2_{ran}$ entrenado y evaluado con $BBDD_{ran}$. Las métricas de este modelo son peores que las de ambos modelos secuenciales sobre $BBDD_{seq}$.

Estos resultados parecen indicar que el uso de datos secuenciales, tanto en el entrenamiento como en la evaluación, puede tener un impacto beneficioso a la hora de mejorar la precisión de la detección de recaptura de pantallas en documentos.

V. Transferencia

En esta sección incluimos otra evaluación, en este caso con datos provenientes de un escenario de uso real de la tecnología de verificación de identidad de Mobbeel. Para ello, utilizando nuestro servicio de verificación de documentos, nuestro equipo ha registrado diferentes intentos de verificación, utilizando sus documentos de identidad de forma legítima y realizando recapturas de pantalla de los mismos, utilizando diferentes dispositivos (móviles, tabletas, monitores) y bajo diferentes condiciones lumínicas. Esta base de datos, $BBDD_{pri}$, está formada por 2223 imágenes, conteniendo 1963 imágenes reales de documentos y 260 ataques.

En este caso procedemos a evaluar los cuatro modelos presentados en la sección anterior utilizando $BBDD_{pri}$. Los resultados detallados de esta evaluación pueden consultarse en la Tabla III. En esta evaluación usamos como métricas, además de ACC, el *recall* y la puntuación F1. El *recall*, también conocido como tasa de verdaderos positivos, mide la capacidad de un modelo para identificar correctamente todas las instancias positivas en un conjunto de datos. Por otra parte, la puntuación F1 se corresponde con la media armónica de la precisión y el *recall*, y proporciona una medida del equilibrio entre la capacidad de un modelo para evitar falsos positivos y falsos negativos. Preferimos usar estas métricas frente a las de la evaluación anterior ya que son menos sensibles a clases desbalanceadas (como ocurre en este caso con 1963 imágenes de documentos reales y 260 ataques) en comparación con el APCER, BPCER y ACER. Las siguientes fórmulas detallan cómo calcular el *recall* y la puntuación F1:

$$\text{Precisión} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Positivos}} \tag{13}$$

$$\textit{Recall} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Negativos}} \tag{14}$$

$$\text{F1} = 2 \times \frac{\text{Precisión} \times \textit{Recall}}{\text{Precisión} + \textit{Recall}} \tag{15}$$

De nuevo, los modelos secuenciales son los que obtienen mejores resultados, siendo en este caso $MobileNetV2_{seq}$ el que queda en primer lugar. Esto se puede deber a que los datos extraídos de nuestro servicio de verificación no se corresponden con datos secuenciales y por lo tanto $Sequencer2D_{seq}$ no puede aprovechar la dimensión temporal de los datos.

VI. Conclusiones

En este artículo presentamos una propuesta para la detección de capturas de pantalla en entornos de datos restringidos teniendo en cuenta la dimensión temporal de los datos.

A continuación, resumimos las contribuciones principales de este artículo:

1. Nuestra primera aportación consiste en el desarrollo un flujo de trabajo (ver Sección III) que permite crear bases de datos de documentos identidad que contengan muestras reales y de ataques de recaptura de pantalla en entornos de datos restringidos. Utilizando herramientas de generación de documentos sintéticos, especímenes o bases de datos disponibles públicamente, es posible generar una base de datos que puede ser utilizada para entrenar modelos de clasificación sin violar la privacidad utilizando documentos de identidad reales.
2. A continuación, hemos introducido el concepto del RLI (ver Subsección III-C), un índice que sirve para medir cómo de probable es que una imagen sea una recaptura de pantalla en función de cuán presente esté en ella el efecto Moiré. Este índice puede ser utilizado como factor de decisión trivial a la hora de evaluar si una imagen es una recaptura o para medir la calidad de los artefactos a la hora de generar recapturas sintéticas, tal y como hemos hecho nosotros para seleccionar las texturas a transferir.
3. Asimismo, a la hora de enfocar la creación de la base de datos y de su evaluación, nos hemos centrado en estudiar el impacto que podría tener el utilizar datos con dimensión temporal. Así, hemos creado una base de datos que tiene en cuenta qué *frames* pertenecen a un mismo vídeo. En cada grupo de *frames*, aplicamos texturas de pantalla que imitan cómo se moverían los artefactos de pantalla en un vídeo (ver Subsección III-B). De cara a la evaluación, hemos utilizado una arquitectura, *Sequencer2D*, que tiene en cuenta si los datos son secuenciales a la hora de clasificarlos. Los resultados obtenidos en la Sección IV muestran el potencial de esta técnica, obteniendo mejores resultados al usar *Sequencer2D* entrenado y evaluado con una base de datos secuencial.
4. Por último, hemos repetido esta evaluación en un entorno real, demostrando cómo el estudio presentado en este artículo se transfiere a un caso de uso real. Hemos utilizado como base de datos de evaluación documentos y ataques pertenecientes a nuestro equipo técnico utilizando la tecnología de Mobbeel (ver Sección V). En este caso, pese a que los datos evaluados no cuentan con dimensión temporal (son fotos de documentos y sus respectivas recapturas de pantalla), de nuevo el modelo que obtiene mejores resultados ha sido entrenado con datos secuenciales, en este caso una *MobileNetV2*.

Creemos que las aportaciones que hemos realizado en este campo son prometedoras y en un futuro nos gustaría aplicar nuestro flujo de trabajo en conexión con una herramienta de generación de documentos sintéticos. Así, podríamos generar una base de datos de mayor tamaño, pudiendo estudiar cómo afecta el uso de datos de entrenamiento con dimensión en temporal a la hora de evaluar otras bases de datos.

También nos gustaría estudiar el uso de este tipo de datos con diversas arquitecturas más allá de *Sequencer2D*. Sería interesante ver qué arquitecturas ofrecen mejores resultados para documentos de determinado tipo, diferentes condiciones lumínicas, dispositivos de captura, etc. Además, planeamos extender este estudio a otros tipos de ataques de documentos, como podrían ser los escaneos, impresiones en papel, impresiones de tarjetas en alta calidad, etc.

Por último, queremos realizar un estudio que nos permita mejorar y adaptar las contribuciones de este artículo a nuestra tecnología de verificación. Creemos que entrenar modelos de clasificación con datos con dimensión temporal es una aproximación prometedora para poder lidiar con las complejidades de un entorno de verificación de documentos real, mejorando la precisión y las prestaciones generales del sistema de detección de fraude.

Referencias

[1] S. C. Valverde, P. J. Cuadros-Solas, and F. R. Fernández, "Digitalización financiera y covid-19: Evidencia empírica," *Papeles de Economía Española*, vol. 170, p. 143–156, 2021.

[2] BOE, "Boe-a-2021-7966 orden etd/465/2021, de 6 de mayo, por la que se regulan los métodos de identificación remota por vídeo para la expedición de certificados electrónicos cualificados," 2021, retrieved from https://www.boe.es/buscar/doc.php?id=BOE-A-2021-7966.

[3] R. E. . of the European Parliament, of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data, on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), 2016, retrieved from https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679.

[4] ISO/IEC, "Iso/iec 30107-1:2023," 2023, retrieved from https://www.iso.org/standard/83828.html.

[5] R. Markham, J. M. Espin, M. Nieto-Hidalgo, and J. E. Tapia, "Open-set: Id card presentation attack detection using neural transfer style," 2023, retrieved from https://arxiv.org/abs/2312.13993.

[6] D. V. Polevoy, I. V. Sigareva, D. M. Ershova, V. V. Arlazarov, D. P. Nikolaev, Z. Ming, M. M. Luqman, and J.-C. Burie, "Document liveness challenge dataset (dlc-2021)," *Journal of Imaging*, vol. 8, no. 6, p. 181, 2022.

[7] C. Chen, S. Zhang, F. Lan, and J. Huang, "Domain generalization for document authentication against practical recapturing attacks," 2021, arXiv preprint arXiv:2101.01404, Retrieved from https://arxiv.org/abs/2101.01404.

[8] D. Benalcazar, J. E. Tapia, S. Gonzalez, and C. Busch, "Synthetic id card image generation for improving presentation attack detection," *IEEE Transactions on Information Forensics and Security*, vol. 18, p. 1814–1824, 2023.

[9] M. J. Chong and D. Forsyth, "Effectively unbiased fid and inception score and where to find them," 2020, arXiv preprint arXiv:1911.07023.

[10] K. Bulatov, E. V. Emelianova, D. V. Tropin, N. S. Skoryukina, Y. S. Chernyshova, Z. Ming, J.-C. Burie, and M. M. Luqman, "Midv-2020: a comprehensive benchmark dataset for identity document analysis," 2022, retrieved from http://l3i-share.univ-lr.fr/MIDV2020/midv2020.html.

[11] Y. Zhang and L.-Y. Shen, "Detection of moiré pattern in high-resolution images," *SIViP*, vol. 18, p. 561–568, 2024.

[12] D. I. Q. Testing, "Nyquist frequency, aliasing, and color moiré — documentation v23.2," imatest.com, Retrieved from https://www.imatest.com/docs/nyquist-aliasing/.

[13] F. Joucken, F. Frising, and R. Sporken, "Fourier transform analysis of stm images of multilayer graphene moiré patterns," 2014, arXiv preprint arXiv:1409.3105.

[14] Y. Tatsunami and M. Taki, "Sequencer: Deep LSTM for image classification," 2022.

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