ISSN 1870-4069

Sistema de procesamiento de IDs basado en modelos de aprendizaje profundo

Raúl Aguilar-Figueroa, Rolando Borja-Brito,
Carlos Daniel Virgilio-González

Biometría Aplicada,
Departamento de Innovación,
México

{raguilar, rborja, cvirgilio}@biometriaaplicada.com

Resumen. Una de las fases del onboarding digital, consiste en autenticar la identidad del cliente a partir de la captura de una fotografía de un documento de identidad (ID). Esta imagen del ID suele adquirirse en entornos donde factores como la orientación o la presencia de fondos, complican la correcta ejecución de tareas posteriores en el flujo del onboarding. Con la finalidad de superar estos inconvenientes, en el presente artículo se introduce un sistema para el procesamiento de IDs, el cual localiza, segmenta y alinea el ID a través de módulos construidos completamente en función del enfoque del aprendizaje profundo. Para la tarea de localización, se utiliza el algoritmo de detección YOLOv4; la segmentación del ID se realiza con la arquitectura U-Net, y la alineación del ID se logra con el sistema RotNet. Buscando incrementar el tamaño de la base de datos de IDs, se aplica una metodología de aumentación de datos que genera imágenes de IDs sintéticas. Los resultados obtenidos al evaluar los modelos de localización, segmentación y alineación con base en credenciales de elector de los Estados Unidos Mexicanos, resaltan su robustez ante las variaciones agresivas que suelen tener las imágenes de entrada, pues el modelo de localización alcanza un rendimiento de 0.9634 con la métrica *IoU*, por 0.9820 del modelo de segmentación de acuerdo a la métrica *coeficiente de dice*, mientras que los ángulos inferidos por el modelo de alineación tienen una discrepancia promedio de apenas 1.5920 con respecto a los ángulos verdaderos.

Palabras clave: IDs, onboarding digital, aprendizaje profundo, YOLOv4, U-Net, RotNet.

ID Processing System based on Deep Learning Models

Abstract. One of the phases of digital onboarding, consists of authenticating the identity of the client by capturing a photograph of an identity document (ID). This image of the ID is usually acquired in environments where factors such as orientation or the presence of backgrounds, complicate the correct execution of subsequent tasks in the onboarding flow. In order to overcome these

drawbacks, this article introduces a system for the processing of IDs, which locates, segments and aligns the ID through modules built entirely based on the deep learning approach. For the localization task, the YOLOv4 detection algorithm is used; the segmentation of the ID is done with the U-Net architecture, and the alignment of the ID is achieved with the RotNet system. Seeking to increase the size of the ID database, a data augmentation methodology capable of generating synthetic ID images is applied. The results obtained when evaluating the localization, segmentation and alignment models based on voter credentials from the United Mexican States, highlight their robustness in the face of the aggressive variations that input images usually have, since the localization model achieves a peformance of 0.9634 with the *IoU* metric, by 0.9820 of the segmentation model according to the *dice coefficient* metric, while the angles inferred by the alignment model have an average discrepancy of only 1.5920 with respect to true angles.

Keywords: IDs, digital onboarding, deep learning, YOLOv4, U-Net, RotNet.

1. Introducción

En la actualidad, debido a factores como la masificación de los dispositivos conectados a internet y la crisis sanitaria provocada por el COVID-19, la implementación de sistemas de onboarding digital se ha convertido en una necesidad prioritaria en el sector financiero y en la industria en general, ya que para incorporar o dar de alta a nuevos clientes, el procedimiento tradicional que implica que el cliente se traslade físicamente a la institución y lleve a cabo largos trámites presenciales, es reemplazado por un proceso completamente remoto y automatizado, el cual genera una reducción en los costos financieros, una mejor experiencia del usuario y la eliminación de los formularios de papel, entre otros beneficios [10, 2, 8]. Una de las fases del onboarding digital, consiste en autenticar la identidad del cliente a partir de la captura de una fotografía de un documento de identidad (ID) [16].

En este escenario, se vuelve indispensable contar con sistemas de análisis de documentos que permitan procesar la imagen del ID con el objetivo de realizar tareas posteriores en el flujo del onboarding, como la extracción de información de interés, la verificación de la pertenencia del documento a una categoría específica o la identificación de documentos falsificados [1, 7].

Estos sistemas de procesamiento de IDs, deben ser capaces de operar en entornos no controlados, ya que es probable que las fotografías de los IDs ingresados por los usuarios, contengan variaciones ocasionadas por cambios en la perspectiva, la iluminación, la orientación y la traslación.

Asimismo, es muy común que las fotografías de IDs presenten distintos fondos, siendo este un aspecto trascendental que dificulta la localización de la región correspondiente al ID, y que es causado por elementos adversos presentes en la captura realizada, tales como el bajo contraste o la similitud de las tonalidades de los colores entre el fondo y el ID [3].

Con la intención de superar los problemas previamente mencionados, en este artículo se presenta un sistema para el procesamiento y análisis de IDs que permite localizar, segmentar y alinear el ID, utilizando para cada una de estas etapas, módulos expresamente construidos con base en un área del aprendizaje máquina conocida como aprendizaje profundo.

De acuerdo a nuestro conocimiento, el sistema propuesto se destaca como el primer esfuerzo en el estado del arte del procesamiento de IDs, en el que sus módulos funcionan íntegramente con técnicas de aprendizaje profundo, y que, además, asume que los IDs han sido capturados en ambientes sin restricciones, en donde el grado de orientación del ID, su ubicación en la imagen, los fondos, la perspectiva y la iluminación, no son homogéneos y pueden cambiar dependiendo de la fotografía que capture un usuario en particular.

En la etapa de localización del ID, se utiliza el algoritmo de detección YOLOv4 [5]; para la fase de segmentación, se emplea la arquitectura U-Net [19], y en la fase de alineación, se hace uso del sistema RotNet [20]. También es importante señalar que, por primera vez en el estado del arte, en este artículo se utilizan YOLOv4 y RotNet para el procesamiento de IDs.

Otra de las contribuciones esenciales de este trabajo, es que durante el entrenamiento de los modelos de segmentación y orientación, se generan imágenes sintéticas especialmente diseñadas para asemejarse a las fotografías de los IDs capturados por los usuarios, lo cual incrementa drásticamente el tamaño de la base de datos y ayuda a disminuir el fenómeno principal que aqueja al rendimiento de los modelos de aprendizaje profundo: el sobreajuste.

El resto del artículo se organiza de la siguiente manera. En la Sección 2, se ofrece un resumen de los trabajos del estado del arte referentes al procesamiento de IDs. La teoría preliminar relacionada con el enfoque del aprendizaje profundo y los algoritmos de este tipo que forman parte del sistema de procesamiento de IDs propuesto, se presentan en la Sección 3.

La descripción del flujo de procesamiento del sistema se expone en la Sección 4. La información respecto a los resultados experimentales, se encuentra en la Sección 5. Finalmente, en la Sección 6 se brindan las conclusiones obtenidas de esta investigación, así como el trabajo futuro a realizar.

2. Trabajos relacionados

En el estado del arte, la implementación de métodos de aprendizaje profundo en los sistemas de análisis y procesamiento de IDs, es un ámbito de documentación escasa, y la mayoría de los trabajos de investigación, se concentran en abordar los inconvenientes en la captura de IDs mediante técnicas tradicionales de procesamiento de imágenes.

Fang et al. [11], propusieron un sistema de identificación de IDs de China que determina el ángulo de inclinación de los IDs por medio de la Transformada de Hough, aunque no se especifica si a través de este enfoque, la corrección en la alineación se realiza sólo si los IDs tienen un ángulo de orientación en un rango determinado o si el ángulo de inclinación es despreciable.

En el trabajo de Xu et al. [21], se presentó un proceso de reconocimiento para ambos lados de los IDs de China usados en los experimentos, el cual incluye la alineación del ID considerando ángulos de inclinación menores a 45°. Para alinear correctamente el ID, se emplea la Transformada de Hough, seguida de una transformación afín.

Posteriormente, la etapa de localización de la región del ID se lleva a cabo a través de la detección de ciertos elementos del ID mediante el algoritmo de AdaBoost, el cual es alimentado por características de tipo Haar. Vale la pena destacar que, en los dos trabajos referenciados con antelación, no se ofrece una descripción cuantitativa de los resultados experimentales alcanzados en la detección y alineación de los IDs.

En [3], Attivissimo et al. presentaron un prototipo que, entre otras operaciones, localiza IDs italianos con base en un algoritmo que ajusta iterativamente los vértices de un rectángulo de referencia hasta que estos se acoplan con los vértices del ID, el cual debe estar posicionado en la parte central de la imagen.

Los resultados experimentales indican que los vértices de los IDs fueron detectados con un tasa de precisión de 68.57 %. Castelblanco et al. [7], propusieron un sistema de análisis de IDs de Colombia que incluye una etapa de segmentación de IDs con rotaciones en el rango de [0°, 359°] y que se realiza por medio de la arquitectura U-Net.

Después, el ID es recortado y alineado usando una regresión lineal y una transformación en la perspectiva. El rendimiento del modelo de segmentación presentó una exactitud de 98.41 % y un valor en el índice de Jaccard de 0.98. Por otra parte, el método de recorte alcanzó una exactitud de 88.54 %.

3. Teoría preliminar

En esta Sección, se establece un contexto teórico pertinente para la comprensión del funcionamiento del sistema propuesto. Se describen los principios básicos del aprendizaje profundo y de un algoritmo de aprendizaje profundo denominado red neuronal convolucional (CNN, del inglés Convolutional Neural Network), así como de los modelos YOLOv4, UNet y RotNet, los cuales fueron construidos a partir de CNNs.

3.1. Aprendizaje profundo

Los algoritmos que operan bajo el paradigma del aprendizaje profundo, brindan la posibilidad de extraer características directamente de los datos crudos de entrada, sin la necesidad de aplicar técnicas de preprocesamiento, de tal forma que, mediante el enfoque del aprendizaje de representaciones y a través de arquitecturas compuestas por múltiples capas que determinan la profundidad del modelo, los métodos de aprendizaje profundo aprenden de forma automatizada capas sucesivas de representaciones cada vez más sofisticadas y significativas de los datos crudos de entrada [9].

3.2. Redes neuronales convolucionales

Una CNN es una arquitectura de red neuronal profunda que procesa los datos de entrada a través de una estrategia basada en el aprendizaje de patrones locales por medio

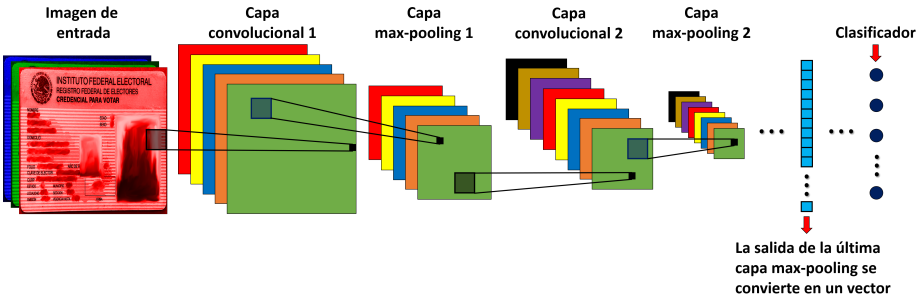

Fig. 1. Ejemplo de una arquitectura clásica de CNN.

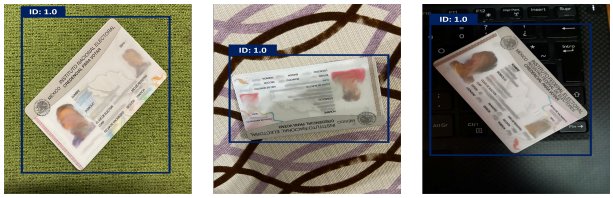

Fig. 2. YOLOv4 localiza y clasifica el objeto de interés contenido en la imagen. En estos ejemplos, se observa que los IDs localizados y delimitados mediante bounding boxes, han sido categorizados en la clase ID, con una precisión del 1.0 (100 %).

de la operación matemática de convolución. Las entradas que son ingresadas a una CNN, se estructuran en múltiples arreglos, como es el caso de las imágenes a color o de las secuencias de lenguaje [14].

Apegándose a los principios de conexiones locales, pesos compartidos, pooling y el uso de una gran cantidad de capas, las CNN's ofrecen robustez ante las variaciones y distorsiones de los datos de entrada, además de una metodología jerárquica espacial para el aprendizaje de patrones [13, 14, 9]. En la Figura 1, se muestra una arquitectura clásica de CNN.

3.3. YOLOv4

You Only Look Once (YOLO) es un sistema de detección de objetos en imágenes propuesto por Redmon et al. [17], el cual actúa como un método de regresión que opera a través de una sola CNN, prediciendo las coordenadas asociadas a la posición de objetos específicos dentro de una imagen.

Estas coordenadas inferidas por YOLO, suelen representarse por medio de cajas delimitadoras, comúnmente conocidas como *bounding boxes*. De forma simultánea a la localización de objetos, YOLO también efectúa la tarea de categorizarlos en sus clases correspondientes (Ver Figura 2). En este trabajo, se hace uso de la cuarta versión de YOLO, desarrollada por Bochkovskiy et al. [5], y denominada YOLOv4.

Fig. 3. Ante una imagen de entrada, U-Net produce su mapa de segmentación correspondiente.

3.4. U-Net

U-Net es una arquitectura de CNN que fue diseñada por Ronneberger et al. [19], para ser aplicada en problemas biomédicos relacionados con la segmentación semántica de células y estructuras neuronales.

Por segmentación semántica, se entiende aquella tarea dedicada a la clasificación de los píxeles de una imagen en ciertas categorías. Por ejemplo, en el caso de la segmentación de fotografías de IDs, los píxeles pueden asignarse ya sea a la clase ID o a la clase fondo.

La estructura de la arquitectura U-Net, que es una CNN completa que carece de capas densamente conectadas, se compone de una ruta de contracción y una ruta de expansión.

La ruta de contracción se conforma por una red neuronal convolucional típica, mientras que la ruta de expansión se constituye por un conjunto de capas convolucionales sucesivas donde se aplica la operación de convolución ascendente, que genera como salida un mapa de segmentación con la misma resolución de la imagen de entrada (Ver Figura 3).

3.5. RotNet

Desarrollado por Daniel Sáez [20], RotNet es un sistema que predice el ángulo de orientación de una imagen por medio de CNNs. En contraste con otras propuestas basadas en CNNs que abordan esta tarea desde la óptica de un problema de regresión [12, 4], Sáez enmarca la tarea de predicción del ángulo de inclinación dentro del ámbito de los algoritmos de clasificación, de tal manera que, ante una imagen de entrada que presente cierto ángulo de rotación, RotNet la clasifica en alguna de las 360 clases que representan cada uno de los posibles ángulos de orientación de dicha imagen.

Como parte de la configuración de RotNet, uno de sus aspectos medulares consiste en la aplicación de la estrategia de transferencia de aprendizaje mediante el modelo ResNet50, incrementando así la eficiencia y la eficacia del proceso de entrenamiento.

Otra de los propiedades fundamentales en el funcionamiento de RotNet, es que durante la etapa de entrenamiento, se generan imágenes sintéticas con rotaciones artificiales en el rango de $[0, 359] \in Z$, aumentando en gran medida la base de datos y refinando la capacidad de generalización del modelo ante las imágenes de prueba, lo que reduce ostensiblemente el riesgo de que el modelo se ajuste en exceso al conjunto de entrenamiento.

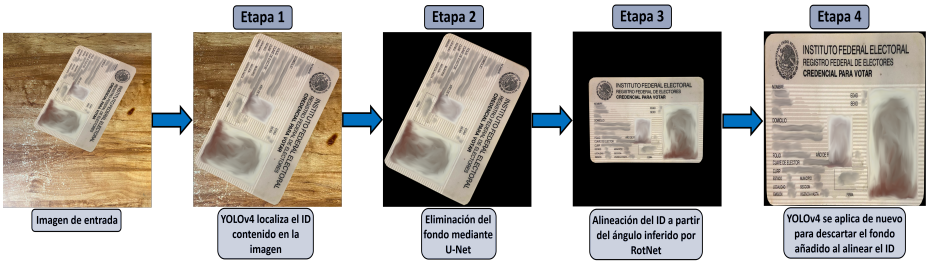

Fig. 4. Flujo de procesamiento del sistema propuesto.

4. Sistema de procesamiento de IDs

A continuación, se brinda una descripción pormenorizada de las etapas de operación del sistema de procesamiento de IDs propuesto. Este sistema funciona de acuerdo a las predicciones que se obtienen al desplegar los modelos de localización (YOLOv4), segmentación (U-Net) y orientación (RotNet). En la Figura 4, se expone el flujo de procesamiento del sistema.

4.1. Etapa 1: Localización del ID

Una vez que el sistema recibe la fotografía del ID capturada por el usuario, el primer proceso que se ejecuta es el de la localización de la región correspondiente al ID a través de YOLOv4. Dado que YOLOv4 proporciona como salida las coordenadas que permiten ubicar el ID, estas coordenadas son usadas para recortar el ID de la fotografía original mediante la función crop de la librería Pillow, de Python (Ver Sección 5 para más detalles técnicos de la implementación del sistema), lo que ocasiona que gran parte del fondo presente en la fotografía sea removido, facilitando así el entrenamiento del modelo de segmentación en la Etapa 2.

4.2. Etapa 2: Segmentación del fondo

En esta fase del sistema, U-Net recibe como entrada la imagen del ID recortado, y genera como salida su mapa de segmentación correspondiente (Ver Subsección 3.4). Luego, mediante una multiplicación a nivel de píxeles, el mapa de segmentación se aplica a la imagen del ID recortado, produciéndose así una imagen resultante de alto contraste entre el fondo y el ID, ya que la porción del fondo original que se venía acarreando desde la Etapa 1, es reemplazada por píxeles que representan el color negro, permitiendo que el ID quede resaltado.

4.3. Etapa 3: Alineación del ID

Después de que la imagen del ID recortado queda libre de fondo, el sistema procede a predecir el ángulo de inclinación del ID por medio de RotNet. Posteriormente, este ángulo es usado para alinear el ID con base en la función rotate_bound de la librería imutils, de Python.

Al alinear el ID, no sólo se cambia el ángulo de orientación del ID, sino de la imagen completa, lo que provoca que los bordes de esta imagen sean rellenados automáticamente con píxeles de color negro (Ver Figura 4).

4.4. Etapa 4: Segunda localización del ID

En la etapa final del sistema, se vuelve a localizar el ID debido a los bordes que se añaden en la etapa previa. Por lo tanto, YOLOv4 se emplea de nuevo para posicionar el ID. La ocurrencia de esta etapa tiene una condición: el ID de entrada debe tener cierta rotación, de lo contrario, basta con que se desplieguen las primeras tres etapas del sistema de procesamiento de IDs.

5. Resultados experimentales

En las Subsecciones siguientes, se detallan las características de la configuración experimental dispuesta para entrenar, validar y evaluar los modelos de localización, segmentación y alineación que componen el sistema de procesamiento de IDs propuesto, así como un análisis del funcionamiento integral del mismo.

En lo que se refiere a la implementación del sistema, esta se llevó a cabo en la plataforma de Google Colaboratory, la cual permite programar en el navegador a través del lenguaje de Python, además de que da acceso a GPUs genéricos para el entrenamiento de modelos de aprendizaje profundo.

Bajo el entorno de Google Colaboratory, los modelos de aprendizaje profundo que forman parte del sistema propuesto, se construyeron mediante Keras, que es una API de alto nivel de Tensorflow.

5.1. Bases de datos

Las imágenes de IDs usadas en este trabajo, corresponden a fotografías frontales de credenciales de elector de los Estados Unidos Mexicanos. En total, se recolectaron 2,000 imágenes con el consentimiento de los participantes, quienes realizaron las capturas con sus smartphones, siguiendo un protocolo con una única restricción: el ID debía aparecer lo más alineado posible.

Otras condiciones de captura, tales como la resolución, la iluminación, la perspectiva o los fondos, podían tener cualquier variación. Además de la adquisición de IDs, también se procedió a recolectar una base de datos de imágenes de fondos que contienen diversas texturas, objetos y fluctuaciones en la iluminación.

Las imágenes de fondos se recopilaron manualmente y también se obtuvieron de otras fuentes [6, 15], lo que permitió construir una base de datos de 26,534 fondos.

Los IDs alineados de las 2,000 imágenes recopiladas, fueron recortados manualmente, y a cada uno de ellos se les agregó un canal alfa, con la intención de que fuera posible generar imágenes sintéticas a partir de ellos, siguiendo un procedimiento que denominamos metodología de aumentación de datos: el ID recortado y con canal alfa puede rotarse libremente en el rango de $[0, 359] \in Z$, para después pegarlo en cualquier posición de un fondo, donde la resolución de este fondo se

ajusta a las dimensiones de la entrada especificada por el modelo de aprendizaje profundo en cuestión, de manera tal que, al pegar el ID, este se adapta al tamaño del fondo redimensionado.

Finalmente, la imagen puede ser sometida a alteraciones en el brillo, aunque en el caso en que el ID se coloque sobre un fondo negro, como ocurre con las imágenes de entrada que requiere RotNet (Ver Subsección 3.5), el cambio en el brillo se aplica únicamente en el ID. Usando esta metodología de aumentación de datos, es factible incrementar fácilmente el tamaño de la base de datos, ya que se puede producir una amplia gama de variaciones por rotación, traslación, iluminación y el elevado número de imágenes de fondos con que se cuenta.

El conjunto de 2,000 imágenes de IDs recortados se particionó en tres conjuntos de datos: entrenamiento, validación y prueba. En el conjunto de entrenamiento, se colocaron 1,200 imágenes, mientras que, tanto al conjunto de validación como al de prueba, se les asignaron 400 imágenes. Las imágenes de los IDs que conforman estos conjuntos, se seleccionaron de forma aleatoria.

A continuación se muestra cómo, en función de estos tres conjuntos de imágenes, a los que denotaremos como las versiones originales, se construyeron las bases de datos para el entrenamiento, validación y prueba de los modelos de aprendizaje profundo que forman parte del sistema de procesamiento de IDs.

5.2. Modelo de localización

Para entrenar de forma adecuada a YOLOv4, fue preciso construir nuevos conjuntos de entrenamiento, validación y prueba, a partir de sus versiones originales y empleando la metodología de aumentación de datos diseñada. De esta forma, se generó un conjunto de entrenamiento con 10,000 imágenes sintéticas, mientras que los conjuntos de validación y prueba, se constituyeron con 2,000 imágenes sintéticas cada uno. Estas imágenes se redimensionaron a un tamaño de 416×416.

Debido a que originalmente los IDs fueron recortados, es sencillo generar las etiquetas para el entrenamiento, validación y evaluación de YOLOv4, las cuales consisten en definir la clase y las coordenadas de la ubicación del ID. En el problema que se está abordando, sólo se tiene la clase ID, y como el ID recortado (y luego rotado) se pega sobre algún fondo, usando para ello coordenadas aleatorias, estas mismas coordenadas son las que se especifican en la etiqueta del ID en cuestión.

El entrenamiento de YOLOv4 se realizó de acuerdo a las especificaciones por default establecidas por su creador [18]. Así pues, el entrenamiento se desarrolló durante 6,000 épocas, utilizando como métricas de evaluación del rendimiento a las funciones *IoU (intersection over union)* y *mAP (mean average precision)*. El modelo con el mejor rendimiento con respecto al conjunto de validación, se alcanzó en la época 5,000, obteniendo un valor de *IoU* de 0.9721 y un 100 % en *mAP*.

Al evaluar este modelo en el conjunto de prueba, se alcanzó un valor de *IoU* de 0.9634 y un 100 % en *mAP*.

5.3. Modelo de segmentación

La arquitectura U-Net se entrenó con base en un enfoque de generación de imágenes sintéticas durante la propia etapa de entrenamiento, incrementando en gran medida el conjunto de entrenamiento y haciendo que el modelo no se sobreajustara, pues ninguna de las imágenes sintéticas era igual entre sí, ya que se diseñaron siguiendo la metodología de aumentación de datos propuesta.

Las imágenes sintéticas producidas, simulan a las imágenes que se obtienen de la Etapa 1 del sistema de procesamiento de IDs (Ver Subsección 4.1): el ID aparece localizado en el centro de la imagen y una gran porción del fondo es eliminada.

Asimismo, de forma adjunta a cada imagen sintética, se genera su etiqueta, que en este caso corresponde a la imagen de un mapa de segmentación (Ver Subsección 3.4).

En este sentido, el entrenamiento de U-Net se inició con los 1,200 IDs recortados que conforman el conjunto de entrenamiento original, y en función de estos IDs, se fueron generando imágenes sintéticas durante el entrenamiento, tal como se explicó previamente.

Por consiguiente, debido a que se requirieron 100 épocas para entrenar el modelo de segmentación, en cada una de estas épocas, los pesos del modelo se ajustaron con 1,200 imágenes de IDs sintéticas y distintas unas de otras, por lo que, al final de la época 100, el modelo terminó entrenándose con 120,000 imágenes completamente diferentes entre sí.

En relación a los conjuntos de validación y prueba, ambos se construyeron a partir de sus versiones originales, conformándose cada conjunto con 2,000 imágenes de IDs junto con sus respectivos mapas de segmentación, los cuales también se generaron sintéticamente, procurando que simularan las propiedades de las imágenes resultantes de la Etapa 1 del sistema de procesamiento de IDs. Tanto las imágenes sintéticas de IDs como sus respectivos mapas de segmentación, se adecuaron a una resolución de 512 $\times$ 512.

Para el entrenamiento de U-Net, como optimizador se usó la función Adam con una tasa de aprendizaje de 0.0001 y como función de pérdida, se empleó binary_crossentropy.

Para darle seguimiento al rendimiento del modelo, se utilizaron las métricas *accuracy* y el *coeficiente de Dice*, donde la primera aporta certeza acerca de la categorización correcta de cada píxel y la segunda proporciona información sobre la precisión en el traslape entre el mapa de segmentación objetivo y el que es inferido por el modelo.

El mejor desempeño del modelo en relación al conjunto de validación, se obtuvo en la época 94, con un *accuracy* de 98.30 % y un 0.9889 de *coeficiente de Dice* (Ver Figura 5). Después de evaluar este modelo en el conjunto de prueba, el *accuracy* obtenido fue de 97.85 %, mientras que el valor del *coeficiente de Dice* llegó a 0.9820.

5.4. Modelo de alineación

De forma similar a la estrategia aplicada para entrenar U-Net y usando nuestra metodología de aumentación de datos, para el modelo de alineación basado en el sistema RotNet también se generaron imágenes sintéticas durante el proceso de

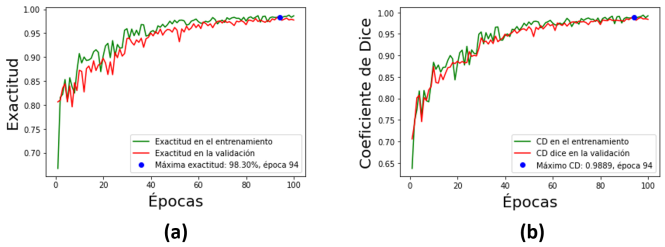

Fig. 5. (a) Gráfica de la exactitud en el entrenamiento y la validación. (b) Gráfica del coeficiente de Dice en el entrenamiento y la validación.

entrenamiento, las cuales simulan las características de las imágenes derivadas de la Etapa 2 del sistema de procesamiento de IDs (Ver Subsección 4.2), donde el ID se ubica en el centro de la imagen y el fondo que aparece es de color negro.

Al rotar la imagen de entrenamiento sintética, el ángulo de rotación se guarda como la etiqueta de dicha imagen rotada.

Por su parte, los conjuntos de validación y prueba se diseñaron partiendo de sus versiones originales, constituyéndose cada conjunto con 2,000 imágenes de IDs generadas sintéticamente junto con sus etiquetas asociadas, donde estas imágenes sintéticas tienen una estructura similar a las de las imágenes que se obtienen de la Etapa 2 del sistema de procesamiento de IDs. La resolución de las imágenes sintéticas generadas para este modelo, se ajustó a un tamaño de 224×224.

Es importante precisar que, aunque de forma predeterminada RotNet puede rotar las imágenes de entrada durante la fase de entrenamiento, no es capaz de pegar una imagen sobre otra ni de cambiar el brillo de las imágenes, por lo que fue necesario modificar su configuración original para hacer que la generación de imágenes sintéticas se adaptara a nuestra metodología de aumentación de datos, con lo que se incrementaron las capacidades iniciales del funcionamiento del sistema.

El entrenamiento del modelo de alineación se desarrolló durante 100 épocas, empleándose la función SGD (Stochastic Gradient Descent) como optimizador, con una tasa de aprendizaje de 0.01 y un momentum de 0.9.

La función categorical_crossentropy se usó para determinar la pérdida del modelo, y la función angle_error, que es una métrica exclusiva de RotNet, sirvió para monitorear la discrepancia entre los ángulos esperados y los ángulos inferidos por el modelo.

Es en la época 62 donde se obtiene el mejor rendimiento del modelo con respecto al conjunto de validación, alcanzándose un valor de angle_error de 1.5600 (Ver Figura 6).

Una vez que este modelo se evaluó en el conjunto de prueba, el angle_error que se obtuvo fue de 1.5920.

5.5. Despliegue y evaluación del sistema de procesamiento de IDs

Luego de haber entrenado los modelos de localización, segmentación y alineación, estos se usaron para poner en operación el sistema de procesamiento de IDs, tomando como entradas a las imágenes y etiquetas que conforman el conjunto de prueba del modelo de localización (Ver Subsección 5.2).

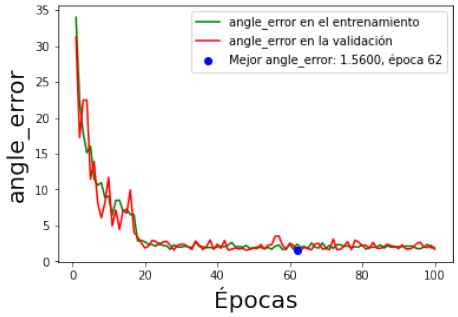

Fig. 6. Gráfica del angle_error en el entrenamiento y la validación.

En consecuencia, estas imágenes fueron procesadas a través de las 4 etapas del sistema propuesto. Como resultado de la etapa 1, el rendimiento obtenido es exactamente el mismo al que se detalla en la Subsección 5.2.

Por su parte, las imágenes recortadas derivadas de la etapa 1, fueron pasadas al modelo de segmentación de la etapa 2, donde se alcanzó un accuracy de 97.27 % y un *coeficiente de dice* de 0.9786.

Las imágenes segmentadas que provienen de la etapa 2, se sometieron al procesamiento del modelo de alineación de la etapa 3, teniendo un desempeño de 1.6015 de acuerdo a la métrica angle_error.

En la etapa 4, se ingresaron las imágenes de los IDs alineados que resultaron de la etapa 3, las cuales tienen bordes de relleno añadidos al ser rotadas, por lo que es necesario ubicar de nuevo el ID, y en esta segunda aplicación del modelo de localización, se obtiene un valor de *IoU* de 0.9659 y un 100 % en *mAP*.

Estos resultados experimentales, indican que el sistema de procesamiento de IDs propuesto en el presente artículo, ofrece un rendimiento elevado ante las variaciones agresivas que comúnmente se presentan en las fotografías de IDs que se ingresan a los sistemas de onboarding digital.

Esta cualidad del sistema propuesto, lo hace apto para ser desplegado en producción, ya que se mejora notablemente la experiencia del usuario, quien no tiene que preocuparse por alinear correctamente la imagen del ID de entrada, colocar el ID en un fondo con textura simple o realizar la captura en un ambiente con iluminación controlada.

6. Conclusiones y trabajo futuro

En este artículo, se introdujo un sistema de procesamiento de IDs basado en modelos de aprendizaje profundo que realiza tres operaciones fundamentales: localización del ID, segmentación del fondo y alineación del ID.

Para el modelo de localización, se empleó el sistema YOLOv4; la tarea de segmentación se realizó a través la arquitectura U-Net, y la corrección en la alineación del ID, se alcanzó gracias al uso del sistema RotNet.

El entrenamiento de los modelos se vio beneficiado por la adopción de una metodología de aumentación de datos que permitió incrementar sustancialmente el número de imágenes disponibles, y, en el caso particular de los modelos de segmentación y alineación, este incremento se llevó a cabo durante la etapa de entrenamiento.

Al evaluar los modelos, los resultados mostraron la viabilidad de desplegar nuestro sistema en un proceso de onboarding digital, ya que este exhibe una considerable robustez ante la mayoría de las variaciones presentes en las fotografías de IDs ingresadas por los usuarios.

Por otro lado, contemplando la posibilidad de que la fotografía del ID posea distorsiones severas en la perspectiva (p. ej. distorsiones afines o proyectivas), se pretende agregar posteriormente un módulo para la aplicación de una matriz de transformación sobre el ID, de tal forma que la imagen resultante facilite el procesamiento de la fase de alineación de nuestro sistema.

La adición de este módulo, sería asimismo benéfica para las etapas subsiguientes del onboarding digital, específicamente, para la tarea de extracción de texto, que es un problema cuyo tratamiento también se tiene planeado como trabajo futuro.

Referencias

1. Arlazarov, V. V., Bulatov, K., Chernov, T., Arlazarov, V. L.: Midv-500: A dataset for identity document analysis and recognition on mobile devices in video stream. Computer Optics, vol. 43, no. 5, pp. 818–824 (2019) doi: 10.18287/2412-6179-2019-43-5-818-824
2. ASOFOM: Digital Onboarding: Definition, characteristics and how it works (2020), https://asofom.mx/2020/12/01/onboarding-digital-solucion-practica-y-omnipresente/
3. Attivissimo, F., Giaquinto, N., Scarpetta, M., Spadavecchia, M.: An automatic reader of identity documents. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 3525–3530 (2019) doi: 10.1109/smc.2019.8914438
4. Baltruschat, I. M., Saalbach, A., Heinrich, M. P., Nickisch, H., Jockel, S.: Orientation regression in hand radiographs: A transfer learning approach (2018) doi: 10.1117/12.2291620
5. Bochkovskiy, A., Wang, C. Y., Liao, H. Y. M.: Yolov4: Optimal speed and accuracy of object detection, (2020) doi: 10.48550/arXiv.2004.10934
6. Burghouts, G. J., Geusebroek, J. M.: Amsterdam Library of Textures (ALOT) (2021), https://aloi.science.uva.nl/public_alot/
7. Castelblanco, A., Solano, J., Lopez, C., Rivera, E., Tengana, L., Ochoa, M.: Machine learning techniques for identity document verification in uncontrolled environments: A case study. In: Mexican Conference on Pattern Recognition. pp. 271–281 (2020) doi: 10.1007/978-3-030-49076-8_26
8. Cath, J.: COVID-19 and the rush to Digital Onboarding. Fintrail (2020), https://www.fintrail.co.uk/news/2020/10/28/covid-19-and-the-rush-to-digital-onboarding
9. Chollet, F.: Deep Learning with Python. Manning Publications Co (2017)
10. Electronic Identification: Digital Onboarding: definition, characteristics and how it works (2021), https://www.electronicid.eu/es/blog/post/digital-onboarding-process-financial-sector/en
11. Fang, X., Fu, X., Xu, X.: ID card identification system based on image recognition. In: 12th IEEE Conference on Industrial Electronics and Applications. pp. 1488–1492 (2017) doi: 10.1109/ICIEA.2017.8283074

12. Fischer, P., Dosovitskiy, A., Brox, T.: Image orientation estimation with convolutional networks. In: German Conference on Pattern Recognition. vol. 9358, pp. 368–378 (10 2015) doi: 10.1007/978-3-319-24947-6_30
13. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE. vol. 86, pp. 2278–2324 (1998) doi: 10.1109/5.726791
14. Lecun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature, vol. 521, no. 7553, pp. 436–444 (2015) doi: 10.1038/nature14539
15. Morales, F.: Complete end-to-end training. Keras (2019), https://keras-ocr.readthedocs.io/en/stable/examples/end_to_end_training.html
16. Narayan, A.: Customer onboarding in a COVID-19 world. Refinitiv (2020), https://www.refinitiv.com/perspectives/financial-crime/customer-onboarding-in-a-locked-down-world/
17. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 779–788 (2016) doi: 10.1109/CVPR.2016.91
18. Redmon, J.: Darknet: Open source neural networks in C (2016), http://pjreddie.com/darknet/
19. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation, (2015) doi: 10.1007/978-3-319-24574-4_28
20. Sáez, D.: Rotnet. guthub (2017), https://github.com/d4nst/RotNet\#readme
21. Xu, J., Wu, X.: A system to localize and recognize texts in oriented id card images. In: IEEE International Conference on Progress in Informatics and Computing (PIC). pp. 149–153 (2018) doi: 10.1109/PIC.2018.8706303