Verificación de información en credenciales escolares mediante técnicas de visión artificial.

Munguia Poblano Erwin, Olarte Astudillo Rodrigo, Ortiz Cruz Luis Gerardo, Albortante Morato Cecilia, Serrano Talamantes José Félix

Escuela Superior de Cómputo IPN México CDMX

Tel. 57-29-6000 ext. 52000 y 52021. E-mail: emunguiap1700@alumno.ipn.mx, rolartea1900@alumno.ipn.mx, lortizc2000@alumno.ipn.mx

Resumen. **– En este trabajo terminal se desarrolla y mide el rendimiento de una herramienta para la verificación de información en las credenciales estudiantiles de la Escuela Superior de Cómputo mediante visión artificial. La herramienta propuesta realiza la verificación a través de un programa que corrobora los datos de la credencial con los registrados en la Dirección de Administración Escolar, accesibles públicamente mediante el código QR de la credencial.**

Palabras clave. **– Credencial escolar, Inteligencia artificial, Procesamiento de texto, Visión artificial.**

I. Introducción

Este trabajo consiste en un proyecto científico en cómputo, orientado a la aplicación de la ingeniería en inteligencia artificial. El objeto de estudio es la verificación de información de las credenciales no inteligentes. Su desarrollo busca investigar alternativas a las etapas de extracción y comparación de información dentro del proceso.

Algunas tecnologías o mecanismos que sirven para la verificación de información en credenciales hacen uso de dispositivos externos para extraer información de la credencial. Estos mecanismos requieren de la presencia de chips o procesadores embebidos que formen parte de las credenciales. El uso de estas tecnologías les otorga el nombre de tarjetas inteligentes [1].

Sin embargo, no todas las instituciones o personas utilizan este tipo de tarjetas. Por ejemplo, dentro de la Escuela Superior de Cómputo (ESCOM) la credencial institucional consiste en una tarjeta plástica de forma rectangular de 8.5 y 5.5 centímetros por lado, que presenta información del alumno como sus datos personales básicos, compuestos por su nombre completo y CURP (Clave Única de Registro de Población); datos académicos, como su programa de estudios, unidad académica y número de boleta; una fotografía del alumno, logotipos institucionales y algunas medidas de seguridad, como un código QR y un código de barras.

Esta credencial es un ejemplo de una tarjeta no inteligente, que otorga a los alumnos un medio de identificación como integrantes del Instituto Politécnico Nacional (IPN) y les posibilita el acceso a las instalaciones tanto de la ESCOM como de otras unidades académicas, así como acceso a servicios culturales, deportivos o de transporte [2].

Por otro lado, para corroborar la veracidad de los datos presentes en las tarjetas no inteligentes se recurre a procesos no automatizados de verificación de la información visible. Este tipo de procesos dependen de las velocidades de reacción y pensamiento humano para la captura o comparación de información.

Una herramienta capaz de efectuar tales procesos sin recurrir a dispositivos físicos adicionales, cómo los de las tarjetas inteligentes, posibilitaría un aumento en el grado de automatización del proceso de verificación de información en este tipo de tarjetas.

Puesto que, desde una perspectiva general, el problema de verificación de información en la credencial se puede modelar como un proceso de comparación de la información proveniente de dos fuentes, la credencial escolar y la información correcta correspondiente, para la solución de este problema, se plantea la posibilidad del desarrollo de una herramienta con inteligencia artificial, ya que, el uso de técnicas como la visión artificial no requiere de información adicional a la que ya está presente en la credencial.

A partir del objetivo general de desarrollar una herramienta para la verificación de información en las credenciales estudiantiles de la Escuela Superior de Cómputo, mediante mecanismos de visión artificial para extraer la información, se establecen los siguientes objetivos específicos:

- Obtener los caracteres presentes en la credencial escolar mediante mecanismos de visión artificial.
- Consultar y extraer la información oficial de la credencial de la página web perteneciente a la Dirección de Administración Escolar a través de raspado web.
- Comprobar la coincidencia aproximada de la información adquirida de la credencial y la consultada, recurriendo al cálculo de una medida de similitud entre cadenas de caracteres.
- Proporcionar los resultados de la verificación de información obtenidos por la herramienta.

Adicionalmente, se establecen las siguientes hipótesis:

Si los algoritmos de inteligencia artificial son capaces de extraer el texto visible de la credencial con un rendimiento de exactitud a nivel de letras superior al 70%, entonces se puede construir una herramienta de verificación de información sin recurrir a tarjetas inteligentes para la extracción de información.

$$H_0: \text{Media de exactitud a nivel de letras} = 0.7 \quad (1)$$
$$H_a: \text{Media de exactitud a nivel de letras} > 0.7 \quad (2)$$

Si la herramienta desarrollada puede realizar la verificación, con un rendimiento promedio en tiempo menor a 30 segundos, entonces se puede utilizar para verificar información de credenciales escolares en actividades académicas.

$$H_0: \text{Media de tiempo de la herramienta} = 30 \quad (3)$$
$$H_a: \text{Media de tiempo de la herramienta} < 30 \quad (4)$$

Por último, si el procedimiento para la extracción de la información visible logra una exactitud promedio a nivel de letras similar entre credenciales verticales y horizontales, entonces puede ser empleado para la verificación de información en credenciales que presentan diversos formatos de disposición de sus elementos.

$$H_0: \text{Media de exactitud a nivel de letras}_{\text{credenciales horizontales}} = \text{Media de exactitud a nivel de letras}_{\text{credenciales verticales}} \quad (5)$$
$$H_a: \text{Media de exactitud a nivel de letras}_{\text{credenciales horizontales}} \neq \text{Media de exactitud a nivel de letras}_{\text{credenciales verticales}} \quad (6)$$

Se plantea entonces el uso de reconocimiento óptico de caracteres (OCR), raspado web y la comparación aproximada de cadenas para alcanzar los objetivos, mientras que, para la evaluación de las hipótesis, se recurre al desarrollo de una herramienta que implemente el proceso propuesto.

Utilizando la herramienta y un conjunto de datos etiquetado de imágenes de las credenciales objetivo, se realiza una medición de su rendimiento, así como la evaluación de los efectos de la distribución de los datos a extraer dentro de la credencial.

El proceso propuesto consiste en la captura de imágenes para la extracción del texto que se encuentra en la credencial y la adquisición de los datos legítimos por medio del sitio web de la Dirección de Administración escolar (DAE) accesibles a través de la decodificación del código QR presente en la parte posterior de la credencial.

En tanto que la comparación se realiza con ayuda de un algoritmo que proporciona un valor comparativo que permite realizar un contraste entre evaluaciones.

II. METODOLOGÍA

La visión artificial o visión por computadora es un campo de la inteligencia artificial que tiene como objetivo que las computadoras emulen las capacidades de la visión humana incluyendo el aprendizaje, la inferencia y la toma de decisiones a partir de entradas visuales [3].

Dentro de este campo uno de los problemas más estudiados ha sido el reconocimiento de caracteres o también conocido como reconocimiento óptico de caracteres (OCR). El OCR tiene el objetivo de convertir el texto escrito o impreso en texto que sea editable por medio de una computadora [4].

Como parte de los primeros intentos en resolver este problema se utilizaron procedimientos, que empleaban diversos mecanismos de procesamiento digital de imágenes y reconocimiento de patrones para realizar una segmentación y posterior clasificación de los caracteres presentes en la imagen. Si bien estos procedimientos obtuvieron

buenos resultados dependían de una compleja especialización para ser aplicables al mayor número de casos posible.

Por esa razón, se suele recurrir al aprendizaje profundo en métodos más recientes. El aprendizaje profundo un tipo de aprendizaje de máquina, donde se emplean redes neuronales para aprender de forma automática la representación necesaria que debe ser aplicada a los datos sin procesar para que sean útiles en modelos de detección o clasificación [5].

En este trabajo se consideró como problema principal del procedimiento propuesto, la extracción de caracteres, es por ello que se contemplan tres implementaciones con las cuales resolver este problema, dos de ellas utilizando bibliotecas externas del lenguaje de programación Python y una implementación propia utilizando modelos preentrenados y no preentrenados.

Las tres implementaciones utilizadas son:

- EasyOCR
- Pytesseract
- Implementación propia

La primera de ellas utiliza la biblioteca EasyOCR, esta biblioteca realiza un proceso de seis etapas para la extracción del texto:

1) Preprocesamiento: Durante esta primera etapa se realiza una normalización a cada componente de color de la imagen, así como un escalado [6].
2) Detección: Se utiliza una red neuronal convolucional autocodificadora CRAFT (Character-Region Awareness For Text detection) para la obtención de dos mapas de características. Estos mapas indican la probabilidad que posee cada pixel de pertenecer a una región de la imagen que contenga texto, así como la probabilidad de que corresponda a la región de unión entre caracteres [7].
3) Procesamiento intermedio: Se utilizan los mapas de características para delimitar y segmentar las regiones de texto de la imagen [8].
4) Detección: Las regiones segmentadas son procesadas en una red neuronal recurrente convolucional. Esta red aplica filtros convolucionales a la imagen con el objetivo de obtener una secuencia de vectores de características. Las secuencias resultantes son procesadas por las capas recurrentes de la red, de tipo LSTM (Long Short Term Memory) bidireccional, para realizar las predicciones de probabilidad sobre los caracteres, correspondientes a cada vector en la secuencia [9].
5) Decodificación: Se utiliza el método de etiquetado CTC (Connectionist Temporal Classification) y un algoritmo voraz para realizar la decodificación de la secuencia en palabras [10].
6) Posprocesamiento: Se realiza la concatenación de los resultados de cada recuadro de texto en la imagen [11].

La segunda implementación realiza un preprocesamiento a la imagen en el que se realiza una conversión a escala de grises utilizando los coeficientes NTSC.

A la imagen resultante se le aplica un umbralizado adaptativo gaussiano: utilizando un filtro gaussiano de 11 por 11. Se toma como umbral el valor promedio ponderado de los pixeles en el vecindario menos un valor constante de nueve.

La imagen binaria obtenida del proceso anterior es invertida y dilatada morfológicamente con un filtro de máximo de 3 por 3 en una iteración.

Posteriormente como parte del proceso de la biblioteca Pytesseract se realiza un análisis de la disposición del texto, es decir se emplean las componentes conexas de la imagen binaria para agrupar contornos y organizarlos en líneas de texto [12].

Luego se utiliza una red recurrente LSTM para intentar decodificar las palabras en las líneas de texto [13]. Para ello se utiliza una ventana de extracción de características que es desplazada sobre la línea de texto a procesar [14].

En caso de que este procedimiento no obtenga resultados aceptables, los segmentos de la imagen agrupados en líneas de texto son separados en palabras y caracteres utilizando distintos algoritmos [12]. Después, la clasificación se realiza con dos pases usando clasificadores con técnicas de reconocimiento de patrones [12].

Ambos pases utilizan distintos clasificadores para mejorar la detección de palabras, y el segundo de ellos utiliza como entrada la salida del primero [12].

La tercera implementación comienza de igual forma con un preprocesamiento de la imagen. Este preprocesamiento consiste en una segmentación por color que comienza por transformar la imagen en un mapa en el que cada pixel representa el valor de la norma euclidiana del vector asociado a su color en la imagen original.

Cada uno de estos valores es transformado utilizando la raíz cubica del cuadrado de su valor y el resultado es reescalado entre los valores de cero y 255 para realizar un umbralizado adaptativo gaussiano con un filtro de 11 por 11 tomando como umbral el valor promedio ponderado de los pixeles en el vecindario menos un valor constante de dos.

La imagen binaria resultante es procesada por la red neuronal convolucional por regiones EAST (Efficient and Accurate Scene Text Detector) para obtener las regiones candidatas a poseer texto en la imagen [15].

Las regiones obtenidas son procesadas por medio de una supresión de no máximos (NMS) para seleccionar los recuadros más probables para cada resultado. Posteriormente se realiza su segmentación y asignación de líneas de texto.

Una vez ordenadas, las regiones segmentadas son divididas en caracteres utilizando las componentes conexas que la integran y un umbral respecto a la razón entre su altura y la del recuadro de la región.

Cada carácter segmentado es enviado a un clasificador con la siguiente arquitectura:

- Imagen de entrada 28×28
- 32 filtros 3×3 con bordes constantes para conservar el tamaño, función de activación ReLU y submuestreo máximo de 2×2
- 64 filtros 3×3 con bordes constantes para conservar el tamaño, función de activación ReLU y submuestreo máximo de 2×2
- 128 filtros 3×3 con bordes constantes para conservar el tamaño, función de activación ReLU y submuestreo máximo de 2×2
- Aplanado a un vector de 1152 elementos
- Transformación lineal de 64 características con función de activación ReLU
- Transformación lineal de 128 características con función de activación ReLU
- Transformación lineal de 36 características

El clasificador realiza la predicción del carácter en la imagen y el resultado es concatenado al texto de resultado.

Por otro lado, la extracción de los datos válidos se realiza a través de raspado web mediante las bibliotecas Requests y Beautiful Soup para realizar la petición de la página web y extraer los datos del documento HTML. La URL del sitio es decodificada previamente utilizando una imagen del código QR de la parte posterior de la credencial y las bibliotecas OpenCV y QReader.

Mientras que la comparación de cadenas utiliza una de las variantes de la distancia de Levenshtein [16] y el cálculo de la medida de exactitud final utiliza la siguiente formula:

$$\text{Exactitud}_{\text{letras}} = \frac{\sum_{i=1}^{N_{\text{elementos}}}[|\text{elemento}_i|] - \sum_{i=1}^{N_{\text{elementos}}}[d_{\text{Levenshtein}}(\text{elemento}_i, \text{texto})]}{\sum_{i=1}^{N_{\text{elementos}}}[|\text{elemento}_i|]} \quad (7)$$

III. Resultados

Para la evaluación de las implementaciones, así como de la herramienta final se recolecto un conjunto de datos de 81 imágenes de credenciales con 46 con elementos en disposición horizontal y 35 en disposición vertical.

Para su posterior división en conjuntos de entrenamiento y prueba se utilizaron el 80% en el conjunto de entrenamiento y 20% en el conjunto de prueba de forma estratificada.

Los resultados de la exactitud a nivel de letras son los siguientes:

Tabla 1
Resultados de la evaluación de la herramienta en los conjuntos de datos

	Conjunto de entrenamiento	Conjunto de prueba
EasyOCR	Exactitud: 79% Tiempo: 16.92 s Longitud: 404.89 caracteres	Exactitud: 84% Tiempo: 18.24 s Longitud: 417.29 caracteres

Pytesseract	Exactitud: 46% Tiempo: 0.808 s Longitud: 415.56 caracteres	Exactitud: 49% Tiempo: 0.86 s Longitud: 422.76 caracteres
Implementación propia	Exactitud: 43% Tiempo: 7.16 s Longitud: 229.03 caracteres	Exactitud: 41% Tiempo: 7.10 s Longitud: 229.64 caracteres

Con base en ellos la implementación seleccionada para integrarse a la herramienta fue la correspondiente a la biblioteca EasyOCR puesto que presento los mejores resultados en exactitud.

Adicionalmente para cada hipótesis se asumieron a los conjuntos de datos recolectados como muestras independientes e idénticamente distribuidas con medias y varianzas finitas y definidas lo suficientemente grandes para calcular los estadísticos de prueba. Se utilizó una distribución t como distribución nula de los estadísticos de prueba.

Los resultados de hipótesis principal fueron los siguientes:

TABLA 2
RESULTADOS DEL ESTADÍSTICO DE PRUEBA EN LA HIPÓTESIS PRINCIPAL

Muestra	Valor del estadístico	Grados de libertad	Valor de p
Conjunto de datos recolectado	3.9797	80	7.5419 × 10 ^ -05
Conjunto de entrenamiento	3.0243	63	0.0018
Conjunto de prueba	3.1486	16	0.0031

Tomando una significancia $p < 0.05$ se concluye que existe evidencia suficiente para rechazar la hipótesis nula principal.

En cuanto a las hipótesis secundarias la primera de ellas obtuvo los siguientes resultados:

TABLA 3
RESULTADOS DEL ESTADÍSTICO DE PRUEBA EN LA PRIMERA HIPÓTESIS SECUNDARIA

Muestra	Valor del estadístico	Grados de libertad	Valor de p
Conjunto de datos recolectado	-25.9633	80	4.9930 × 10 ^ -41
Conjunto de entrenamiento	-21.5135	63	4.9992 × 10 ^ -31
Conjunto de prueba	-24.6816	16	1.8280 × 10 ^ -14

Al igual que en el caso anterior, utilizando una significancia $p < 0.05$ se rechaza la primera hipótesis nula secundaria.

Por otra parte, la segunda hipótesis secundaria obtuvo:

TABLA 4
RESULTADOS DEL ESTADÍSTICO DE PRUEBA EN LA SEGUNDA HIPÓTESIS SECUNDARIA

Muestra	Valor del estadístico	Grados de libertad	Valor de p
Conjunto de datos recolectado	0.3105	71.9851	0.7571
Conjunto de entrenamiento	0.5901	57.0021	0.5575
Conjunto de prueba	-0.7081	14.2896	0.4903

En este caso utilizando una significancia $p < 0.025$ no se tiene evidencia suficiente para rechazar la segunda hipótesis secundaria.

IV. CONCLUSIONES

Este trabajo terminal se realizó en dos etapas, la primera etapa del proyecto tuvo como principales resultados la definición de los modelos, lenguajes, herramientas o procedimientos que serían sometidos a la experimentación y medición dentro de la segunda fase, así como, la esquematización requerida como parte de la planeación necesaria para el desarrollo de la herramienta.

En la segunda etapa del proyecto se realizó la implementación, codificación y documentación técnica de cada componente de la herramienta.

Finalizado este proceso se comenzó con su integración, así como con las pruebas utilizando los datos recolectados con lo que se pudo determinar que tanto para la hipótesis principal y la primera hipótesis secundaria se obtuvo suficiente evidencia para rechazar la hipótesis nula.

En conclusión, este trabajo terminal permitió exploración y practica de cada una de las fases de un proyecto de aprendizaje de máquina, también permitió profundizar en aplicaciones de la inteligencia artificial. En última instancia los resultados aquí obtenidos permitieron aproximar en rendimiento a algunas medidas de las herramientas preexistentes consideradas y sobre todo proporcionaron una explicación indirecta del gran éxito que el aprendizaje de máquina y las redes

neuronales han tenido durante los últimos años para resolver problemas complejos.

Reconocimientos

Los Autores agradecen a la Escuela Superior de Cómputo del Instituto Politécnico Nacional por el apoyo recibido y las facilidades otorgadas para el desarrollo del presente trabajo terminal.

Referencias

[1] K. M. Shelfer y J. D. Procaccino, «Smart card evolution,» *Communications of the ACM,* p. 83–88, 2002.

[2] Dirección de Administración Escolar, «Reposición de Credencial Institucional,» [En línea]. Available: https://www.ipn.mx/dae/tramites/credencial-institucional.html.

[3] B. R. Hunt, «Digital Image Processing,» *In Advances in Electronics and Electron Physics,* vol. 60, p. 161–221, 1983.

[4] M. A. Awel y A. I. Abidi, «Review on optical character recognition,» *International Research Journal of Engineering and Technology (IRJET),* pp. 3666-3669, 2019.

[5] Y. LeCun, Y. Bengio y G. Hinton, «Deep learning,» *Nature,* vol. 521, nº 7553, p. 436–444, 2015.

[6] JaidedAI, «easyocr/detection.py,» GitHub, 22 agosto 2022. [En línea]. Available: https://github.com/JaidedAI/EasyOCR/blob/master/easyocr/detection.py.

[7] Y. Baek, B. Lee, D. Han, S. Yun y H. Lee, «Character Region Awareness for Text Detection,» *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 9357-9366, 2019.

[8] Jaided AI, «EasyOCR,» GitHub, 4 septiembre 2023. [En línea]. Available: https://github.com/JaidedAI/EasyOCR.

[9] B. Shi, X. Bai y C. Yao, «An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition,» *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 39, nº 11, pp. 2298-2304, 2017.

[10] A. Hannun, «Sequence Modeling with CTC,» *Distill,* 2017.

[11] Jaided AI, «easyocr/easyocr.py,» GitHub, 28 junio 2023. [En línea]. Available: https://github.com/JaidedAI/EasyOCR/blob/master/easyocr/easyocr.py.

[12] R. Smith, D. Antonova y D.-S. Lee, «Adapting the Tesseract Open Source OCR Engine for Multilingual OCR,» *MOCR '09: Proceedings of the International Workshop on Multilingual OCR,* 2009.

[13] R. Smith, «Architecture and Data Structures,» 2016. [En línea]. Available: https://github.com/tesseract-ocr/docs/blob/main/das_tutorial2016/2ArchitectureAndDataStructures.pdf.

[14] A. Ul-Hasan, «Generic Text Recognition using Long Short-Term Memory Networks,» enero 2016.

[15] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He y J. Liang, «EAST: An Efficient and Accurate Scene Text Detector,» *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 2642-2651, 2017.

[16] H. Hyyrö, «Practical Methods for Approximate String Matching,» 2003. [En línea]. Available: https://trepo.tuni.fi/bitstream/handle/10024/67325/951-44-5840-0.pdf.