Andrea Velázquez de Dios
Pablo Ignacio Ochoa Sordo
Juan Carlos Pérez Becerra
Mateo Ruiz Navarro
Michael Stiven Caracas Núñez

Solución al reconocimiento de caracteres para empresa Rappi
Visión computacional

2021-09-18

Abstract

Hoy en día, muchas empresas se enfrentan al reto de asegurar la identidad de sus clientes; nuestro socio formador, Rappi, es una de estas empresas. En el presente documento se pretende ofrecer una solución a Rappi, para procesar e interpretar imágenes de identificaciones de clientes para un proceso de KYC (Know Your Customer), que es relevante para autentificar la identidad del cliente, y así evitar cosas como el robo de identidad, lavado de dinero o el fraude. La automatización de este proceso se logra por medio de la extracción de información a través de modelos de visión computacional y el uso de redes neuronales convolucionales, para el reconocimiento de las etiquetas dentro de las identificaciones.

1 Introducción

Rappi es una empresa que brinda al consumidor una mejor calidad de vida en cuanto al proceso de comprar algún tipo de producto, ya que tiene la capacidad de acercar al consumidor cualquier tipo de producto o servicio en un mismo lugar sin tener que desplazarse. Fundada en Colombia en el año 2015, esta empresa multinacional es una de las más reconocidas en la región latinoamericana. Hoy en día Rappi ofrece al cliente diferentes servicios como: mercado, restaurantes, cajero (Rappicash), los denominados rappifavores, farmacia, entre otros.[1]

1.1 Descripción de la Problemática

A pesar de que a lo largo del tiempo Rappi ha logrado un muy grande alcance y éxito, es indispensable seguir buscando alternativas para mantenerse al día con lo que busca su clientela sin dejar atrás el hecho de que debe existir seguridad y certeza en las identidades de los clientes. Una gran herramienta para lograr esto es el método de KYC (Know Your Customer) o, en español, conoce a tu cliente, que consiste en que los profesionales se esfuercen por verificar la identidad, idoneidad y riesgos involucrados en el mantenimiento de una relación comercial con estos.

Para el entendimiento del proceso KYC y resolución de la problemática, se utilizarán métodos de visión computacional y Deep Learning para desarrollar y entrenar dos modelos de redes neuronales convolucionales (CNN) que logren el reconocimiento de caracteres a partir de fotografías de identificaciones oficiales de México y Colombia, esto con el objetivo de extraer la información necesaria de las identificaciones de clientes y así lograr la automatización del proceso de autenticación del cliente de manera segura y precisa.

2 Metodología utilizada

2.1 Datos

Primeramente se partió desde la recolección de identificaciones obteniendo 13 identificaciones mexicanas (INE) y 48 colombianas (cédulas de ciudadanía), cargando estas imágenes en un cuaderno de programación de Python (Versión 3.6.9) y con ayuda de las librerías *augly* y *magic*, que sirven para la edición de imágenes, se transformaron las imágenes originales añadiéndoles saturación o cambiando la gama de colores, entre otras variaciones. En total se obtuvieron 468 identificaciones mexicanas. Este paso es clave de realizar antes del entrenamiento y aplicación en los modelos, ya que se deben tomar en cuenta los posibles sesgos asociados al reconocimiento de las identificaciones. Esto se debe a que el cliente no siempre tomará la imagen como es deseado y esto puede alterar la precisión del modelo, así que estos escenarios deben ser prevenidos. Se decidió hacer dos modelos diferentes: uno para identificaciones mexicanas y otro para las colombianas.

La creación de nuevas imágenes con sesgo agregado se hizo con el uso de la librería *augly*, la cual permite agregar brillo a una imagen, rotación, bordes, entre otras modificaciones. Con esta herramienta se hizo una base de datos más amplia, que permitirá construir modelos más robustos. Con estas bases de datos se entrenarán los modelos para que sean capaces de actuar en diferentes escenarios.

Para poder llevar a cabo el etiquetado de las secciones las cuales nos interesa extraer, lo que se hizo fue señalar manualmente el área de cada imagen, la cual representa aquello que estamos buscando, utilizando la herramienta de ”ybat-YOLO BBox Annotation Tool”. Esta herramienta permite cargar una imagen, seleccionar un área específica, y asignarle una etiqueta.

Luego de crear una lista con el dataset de imágenes de las identificaciones, las dividimos en dos listas, una de entrenamiento con la cuál se entrenará a la CNN y una de validación para evaluar los resultados de las predicciones generadas por la CNN. La lista de entrenamiento corresponde al 70% de las imágenes del dataset, mientras que la lista de validación corresponde al 30%.

Figure 1: Aplicación de la herramienta ybat en la asignación de etiquetas a las áreas específicas de la identificación.

2.2 Modelo

Recordemos que el modelo que se busca es un modelo de aprendizaje supervisado, por lo que usan datos etiquetados, en este caso, seleccionados con la herramienta de Ybat. Esta herramienta regresa la posición del centro de esa etiqueta, así como el ancho y alto de la caja, relativo al tamaño de la imagen. Conocer de manera numérica la posición, así como el tamaño de las etiquetas en las diferentes imágenes, se usará para entrenar al modelo y que éste logre identificar dónde se encuentra una determinada etiqueta cuando se visualiza una imagen de una identificación, de manera automática. Las cases elegidas para las INEs fueron 5, la cuales son: ”curp, ’domicilio’, ’fecha_de_nacimiento’, ’nombre’, ’sexo’ y para la cédula fueron 10: ’apellidos’, ’estatura’, ’expedicion’, ’fecha_de_nacimiento’, ’foto’, ’lugar_de_nacimiento’, ’nombres’, ’numero’, ’rh’, ’sexo’.

Además, antes de la construcción se realizó un paso intermedio donde, con el fin de realizar, ahorrar tiempo de procesamiento por imagen y eliminar información que pueda ser redúndate, se decidió ajustar las imágenes a cuadrados de 416x416 pixeles. Teniendo ya una base de datos completa se construyó un primer modelo utilizando Redes Neuronales Convolucionales. Se realizó entonces los modelos con el algoritmo *YOLO* (You Only Look Once) mediante transferencia de aprendizaje, para este entrenamiento se empleó la *YOlov5*. Posteriormente, se define un *batch size* de 80, que es un meta-parámetro que permite seleccionar muestras pequeñas del conjunto completo de datos. De esta manera que no se entrena el modelo usando todo el dataset de entrenamiento, sino que se hace a través de subsecciones de este dataset, también llamados ”*batches*”, por medio de iteraciones llamadas ”*épocas*. Esto tiene un positivo impacto en términos computacionales a la hora de entrenar el modelo,para ambos modelos se tomó un total de 100 épocas.

La interpretación de caracteres en las imágenes se llevó a cabo con el uso de la herramienta *pytesseract* (Versión 0.3.8), que es una herramienta especializada en el reconocimiento óptico de caracteres. Esta herramienta regresa el texto incluido en una imagen dada. Lo que se hizo para poder segmentar la información y acomodarla por etiquetas es crear una función que corte el área específica de la imagen que contiene la información de

la etiqueta que se busca, la interprete con *pytesseract* y posteriormente se almacene esa información en un diccionario.

3 Resultados

En la Figura 2 se puede observar varias de las imágenes de verificación luego de pasarlas por el modelo de INEs. Aquí se puede observar la caja para las clases y sobre esta el nombre de la clase pronosticada y la probabilidad de predicción de las clases. En la figura 3 se observa la curva de F1, arrojada por *yolov5* esta contiene el valor del score vs la confianza. Las mismas imágenes y gráficas para el modelo de la CC se pueden observan en las Figuras 4 y 5.

Figure 2: Predicción del modelo para INE

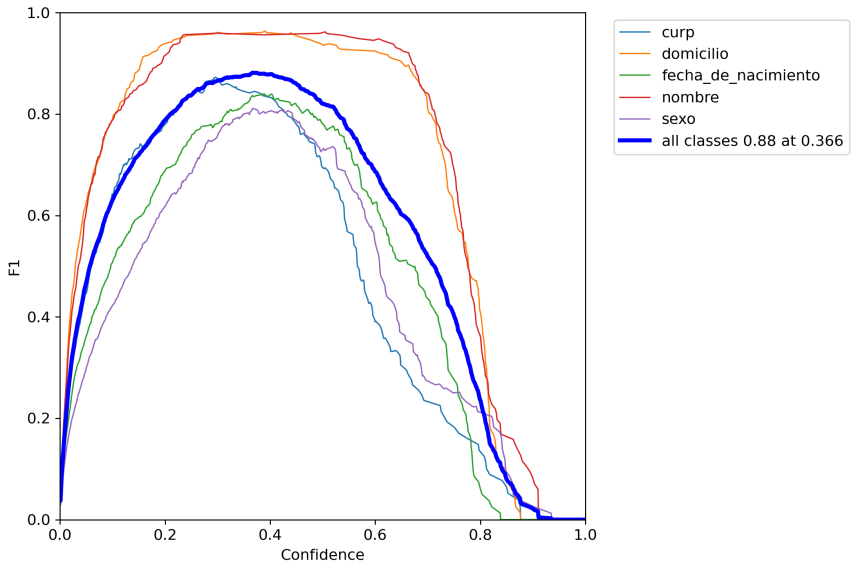

Figure 3: Curva del score F1 del modelo INE

Figure 4: Predicción del modelo para CC

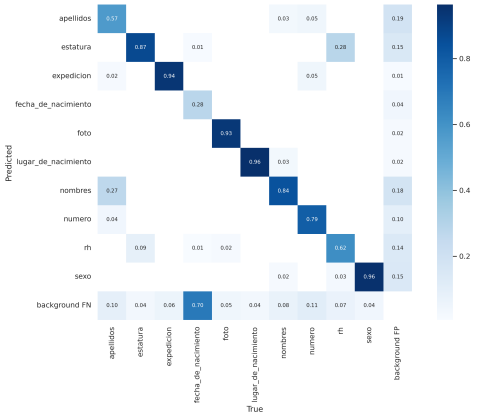

Figure 6: Matriz de confusión arrojada por el algoritmo *yolov5* para el modelo de cédulas de ciudadanía (CC)

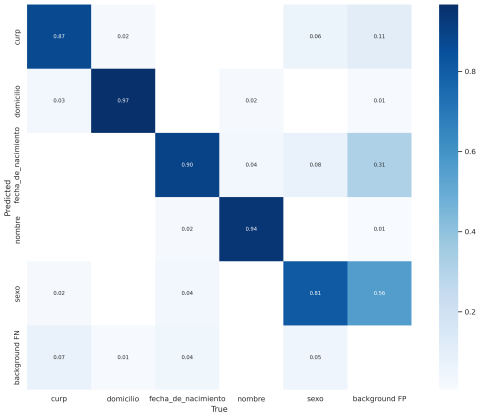

Figure 7: Matriz de confusión arrojada por el algoritmo *yolov5* para el modelo de INE

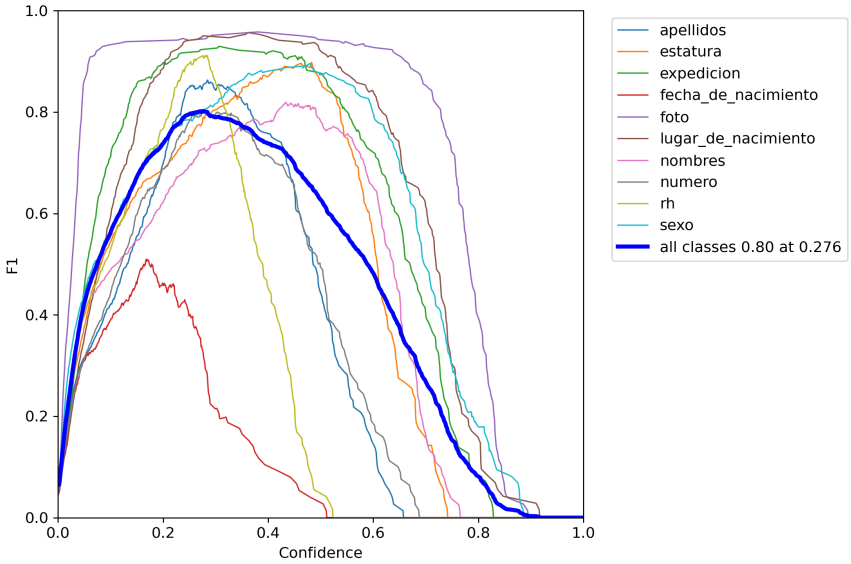

Figure 5: Curva del score F1 del modelo CC

como se observa en la Figura 5 el modelo tiene problemas identificando la fecha de nacimiento del usuario, que con frecuencia la confunde con el fondo de la imagen, como se observa en la matriz de confusión presentada en la Figura 6, para el modelo de INE; pasa lo mismo con la variable 'sexo' en la matriz de confucion del modelo de CC presenada en la figura 7 .En el gráfico de barras de las instancias en las que se encontraron las clases, presentado en la Figura 8, también se presenta en Figura 9 las instancias de cada clase para el modelo de INE, donde se observa que la categoría que meno se identificó fue la de 'domicilio'.

Como se observa en ambas gráficas de F1 score en las Figuras 3 y 5, el modelo de INE tuvo mejor rendimiento que el modelo de CC, esto considerando que un mayor porcentaje de sus variables sobrepasaron el límite de 0.85 en este indicador. Se observó que la interpretación de las clases en las fotografías rotadas, era aceptable, pero su interpretación de texto era deficiente. Cuando se intentaba interpretar el texto en las identificaciones con una cierta rotación, la herramienta de *pytesseract* no era capaz de llevar a cabo la tarea de manera precisa. Por lo tanto, se optó por utilizar únicamente las fotos originales, así como fotos con variaciones en el brillo.Las fotos con alto brillo, pierden precisión a la hora de ser interpretadas, sin embargo, esta precisión se pierde únicamente cuando se trata de niveles muy altos.

Se creó una función llamada "ine", que aceptaba como parámetro una imagen y una lista con los valores de las posiciones relativas de las etiquetas. Esta función automatiza la tarea del recorte de imágenes, así como el almacenamiento de la información en variables. Todos los resultados se pueden encontrar en el repositorio del proyecto[1].

[1] https://github.com/juan-p-b/Reto_Rappi

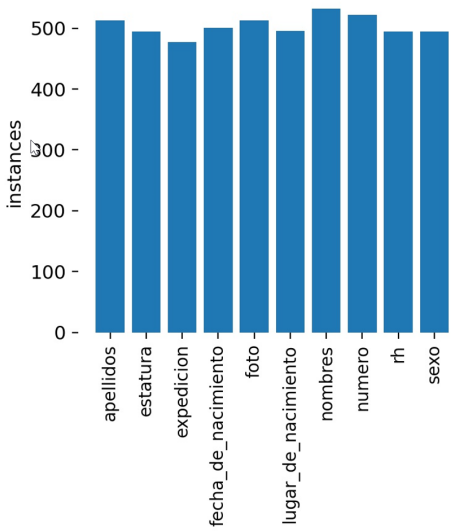

Figure 8: Gráfico de barras de las instancias encontradas por clase para arrojada modelo de cédulas de ciudadanía (CC) por el algoritmo *yolov5*.

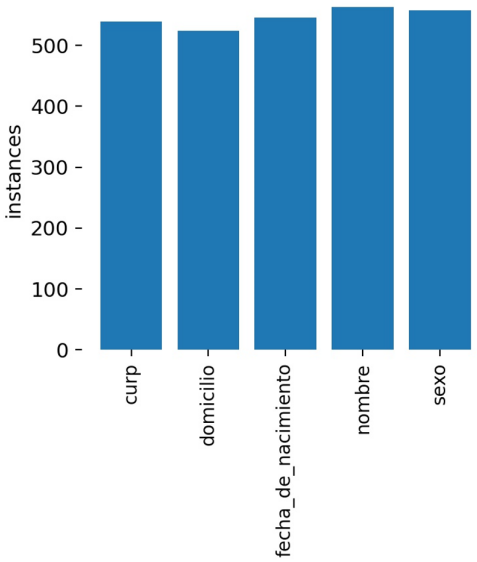

Figure 9: Gráfico de barras de instancias por clase para arrojada modelo de INE por el algoritmo *yolov5*.

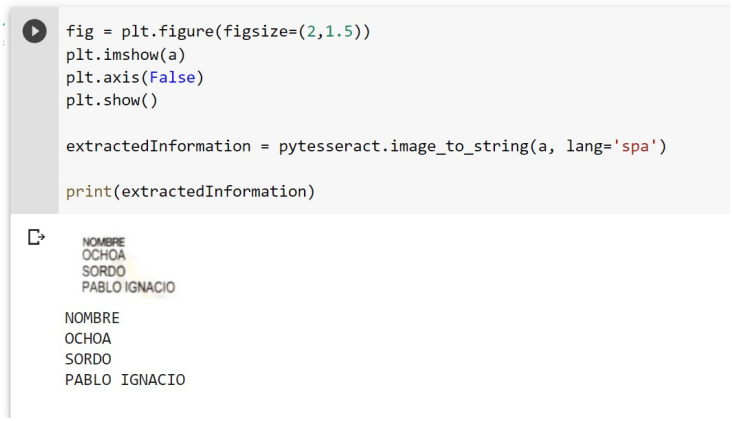

Figure 10: Interpretación del tesseract en la etiqueta *Nombre* para imagen con brillo de 1.4.

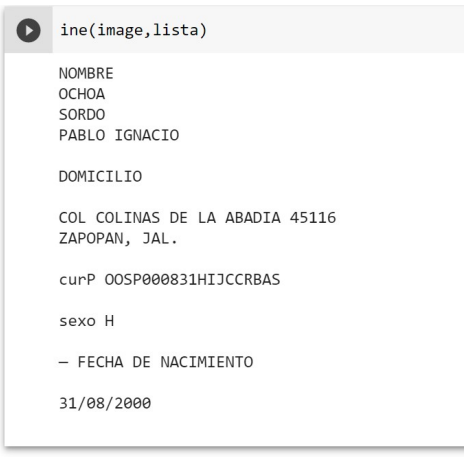

Figure 11: Interpretación del tesseract en todas las etiquetas de una INE.

4 Conclusiones

La visión computacional es una herramienta bastante útil para el hombre en lo que concierne a el aprendizaje de máquina, en la sección 3 se pudo comprobar exitosamente que con la ayuda de esta herramienta es posible,

no solo la detección de caracteres, sino la posibilidad de entrenar un modelo para que con la ayuda de redes neuronales lea y acumule datos en esta misma detección.

Se espera que los hallazgos encontrados en el presente documento sean relevantes para la empresa Rappi en su objetivo de conocer al cliente y en la credibilidad de la información que este proporciona para validar su identidad, para futuras investigaciones o aplicaciones del modelo se recomienda altamente aumentar el número de entradas iniciales, en este caso identificaciones esto para que el modelo tenga un mejor entrenamiento y sea más robusto y de igual manera ser lo más preciso posible con el cliente sobre la manera que se debe tomar la imagen de la identificación para que el modelo pueda reconocer claramente los caracteres.

References

[1] “Qué es Rappi y cómo funciona: conoce cómo mejoramos tu calidad de vida”, Blog de Rappi, 2016. Disponible: https://blog.rappi.com/que-es-rappi/