

Visión Artificial

y Procesamiento de Imágenes



UNIVERSIDAD DE MÁLAGA

Adrián Racero Serrano

Juan Manuel Cardenosa Borrego

Preparado para el día 2 de junio de 2024

Índice

1. Introducción	3
I Conceptos teóricos	3
2. ¿Qué es la visión artificial?	3
3. ¿Cómo funciona la visión artificial?	4
4. La historia de la visión artificial	6
5. Aplicaciones de visión artificial	8
6. Beneficios de la visión artificial	10
7. Ejemplos de visión artificial	11
II Caso práctico - Detección de Matrículas	14
8. Contexto y herramientas utilizadas	14
9. Características de las imágenes del dataset	14
10. Imagen en escala de grises	15
11. Imagen binaria	16
12. Contornos	17
12.1. Filtrar Contornos	18
12.2. ¿Qué ocurre si hay más de un candidato?	20
13. Recortar matrícula	21
14. Procesar matrícula	22

15.Extracción del texto	23
16.Resultados y conclusiones	25
III Anexo	27
17.Repositorio de GitHub	27

1. Introducción

En este trabajo explicaremos los conceptos teóricos de la visión artificial, su funcionamiento, su historia, las aplicaciones actuales, y los beneficios que ofrece, centrándonos en la parte del procesamiento de imágenes.

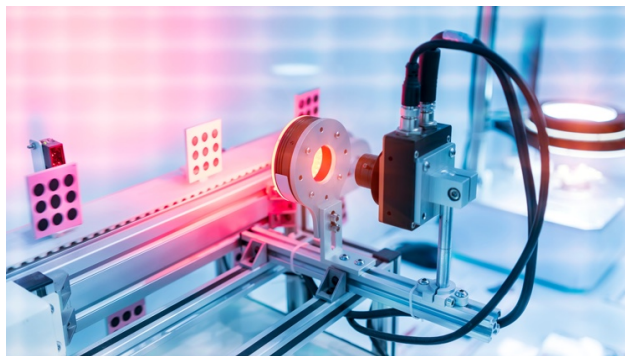
También presentaremos un caso práctico enfocado en la detección de matrículas de coches, empleando técnicas y herramientas de procesamiento de imágenes.

Parte I

Conceptos teóricos

2. ¿Qué es la visión artificial?

La visión artificial es un campo de la inteligencia artificial que se enfoca en desarrollar métodos para que los ordenadores puedan interpretar y comprender imágenes del mundo real.



Este campo combina técnicas de procesamiento de imágenes, aprendizaje automático y análisis de datos para permitir que las máquinas realicen tareas que simulen la visión humana.

En este trabajo nos centraremos esencialmente en la parte de procesamiento de imágenes.

La visión artificial es utilizada en muchos sectores, como pueden ser: la energía, los servicios públicos, la fabricación y la automoción. El mercado de la visión artificial ha alcanzado un valor de 12,88 mil millones de dólares en 2024 y se prevé que crezca hasta los 19,21 mil millones de dólares para 2029, con una tasa de crecimiento anual compuesta del 8,3 % durante el período previsto (2024-2029) [1].

3. ¿Cómo funciona la visión artificial?

La visión artificial necesita muchos datos. Se basa en ejecutar análisis de datos una y otra vez hasta que percibe diferencias y finalmente reconoce imágenes. Por ejemplo, si queremos entrenar a un ordenador para que reconozca imágenes de perros, es necesario incorporarle una gran cantidad de imágenes de perros y elementos relacionados con los perros (por ejemplo, otros animales) para que pueda aprender las diferencias y reconocer correctamente a un perro, sin confundirlo con algún otro animal.

Para conseguir esto, se utilizan dos tecnologías básicas: una especie de machine learning denominado *deep learning*, y una red neuronal convolucional (CNN, convolutional neural network).

Machine learning emplea modelos basados en algoritmos que permiten a un ordenador aprender por sí solo el contexto de los datos visuales. Si se introducen suficientes datos a través del modelo, el ordenador tomará los datos y aprenderá a diferenciar una imagen de otra. Los algoritmos permiten que la máquina sea autodidacta, en lugar de que alguien la programe para que reconozca una imagen.

Una red neuronal convolucional ayuda a un modelo de deep learning o machine learning

a desglosar las imágenes en píxeles a los que se asignan etiquetas. El uso de etiquetas permite realizar convoluciones y predicciones sobre lo que el ordenador está observando. Además, la red neuronal realiza convoluciones y comprueba la exactitud de sus predicciones en una serie de pasos hasta que las predicciones comienzan a pasar en el mundo real. Una vez conseguido esto, el ordenador puede reconocer imágenes de una manera similar a los humanos.

De forma parecida a un humano viendo una imagen a cierta distancia, una red neuronal convolucional primero distingue los bordes más marcados y las formas simples, luego rellena la información a medida que ejecuta iteraciones de sus predicciones. Una CNN se utiliza para comprender imágenes individuales. Una red neuronal recurrente (RNN, recurrent neural network) se emplea de manera similar en aplicaciones de vídeo para ayudar a los ordenadores a comprender cómo se relacionan entre sí las imágenes en una serie de fotogramas.

4. La historia de la visión artificial

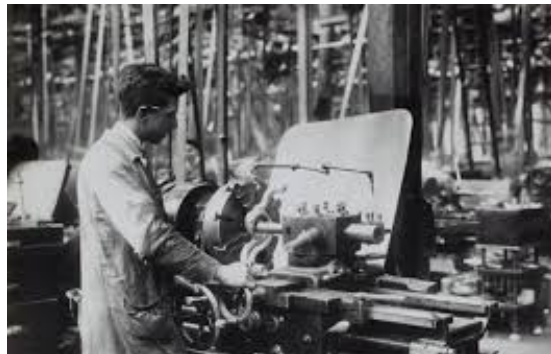
Los ingenieros y científicos de todo el mundo llevan más de 60 años intentando desarrollar formas para que las máquinas sean capaces de comprender y ver datos visuales. La experimentación dio inicio en 1959 cuando un grupo de neurofisiólogos estaban mostrando a un gato una serie de imágenes, intentando asociar una respuesta en su cerebro. Llegaron a la conclusión de que el gato respondía primero a las líneas o los bordes más marcados, lo cual significaba que el procesamiento de imágenes comienza con lo más simple, las formas, como los bordes rectos. [2]

Al mismo tiempo, se desarrolló la primera tecnología capaz de escanear imágenes, permitiendo a los ordenadores digitalizar y adquirir dichas imágenes. Otro hito importante que se alcanzó en 1963 fue cuando los ordenadores fueron capaces de transformar imágenes de dos dimensiones en formas tridimensionales. En los años 60, la IA apareció como un campo académico de estudio y también marcó el inicio de la búsqueda de la IA para resolver el problema de la visión humana.

El año 1974 fue testigo de la aparición de la tecnología de reconocimiento óptico de caracteres (más conocida como OCR), capaz de reconocer texto impreso en cualquier tipo de letra o fuente. Además, el reconocimiento inteligente de caracteres (ICR) era capaz de descifrar texto que había sido escrito a mano y digitalizarlo mediante redes neuronales. A partir de ese momento, ICR y OCR se aplican en el procesamiento de facturas y documentos, pagos móviles, traducción automática, reconocimiento de matrículas de vehículos y otros usos comunes.

Durante el año 2000, el estudio se centró en el reconocimiento de objetos, y en 2001, surgieron las primeras aplicaciones que implementaron el reconocimiento facial en tiempo real.

La normalización de cómo se anotan y etiquetan los conjuntos de datos visuales apareció poco a poco a lo largo de la década de los 2000. En 2010, se hizo público el conjunto de datos ImageNet. Este conjunto de datos contenía millones de imágenes etiquetadas en miles de clases de objetos, proporcionando una amplia base para las redes neuronales convolucionales y los modelos de deep learning que se utilizan a día de hoy. En 2012, un equipo de la Universidad de Toronto participó con una CNN en un concurso de reconocimiento de imágenes. Este modelo, llamado AlexNet, redujo el índice de errores en el reconocimiento de imágenes notablemente. Gracias a este gran avance, las tasas de error se han reducido a un porcentaje muy pequeño.



5. Aplicaciones de visión artificial

Se están realizando una gran cantidad de investigaciones en el campo de la visión artificial. Las aplicaciones del mundo real demuestran la gran importancia de la visión artificial en aplicaciones de empresa, transporte, ocio, asistencia sanitaria y tareas cotidianas. La clave para el crecimiento de estas aplicaciones radica en la gran cantidad de información visual que se genera desde cualquier dispositivo electrónico, como bien pueden ser los teléfonos móviles, cámaras de tráfico, sistemas de seguridad y cualquier otro dispositivo visualmente instrumentado. Esta cantidad de datos desempeña un papel muy significativo en distintos sectores, pero actualmente no se llegan a utilizar completamente. Gracias a toda esta información, se puede crear una base de prueba para entrenar aplicaciones de visión artificial y una plataforma de lanzamiento para que se integren en diferentes actividades humanas.

- IBM empleó la visión artificial para crear My Moments y utilizarla durante el torneo de golf Masters 2018. IBM Watson vio cientos de horas de imágenes del Masters y pudo identificar tanto las escenas como los sonidos de tomas significativas. Organizó cada uno de estos momentos clave y los presentó a los seguidores como carretes personalizados de los momentos más destacados.
- Google Translate permite a los usuarios apuntar la cámara de un teléfono inteligente a un texto escrito en otro idioma y obtener una traducción al idioma que desee el propio usuario de una manera casi inmediata.
- El desarrollo de vehículos autónomos se basa en visión artificial para dar sentido a todo estímulo visual que se pueda recibir gracias a las cámaras y otros sensores del automóvil. Es crucial identificar otros vehículos, señales de tráfico, marcas de carriles,

peatones, etc.

- IBM está utilizando la tecnología de visión artificial para aplicar la IA inteligente a edge computing y ayudar así a los fabricantes de automóviles a identificar los defectos de calidad antes de que un vehículo salga de fábrica.

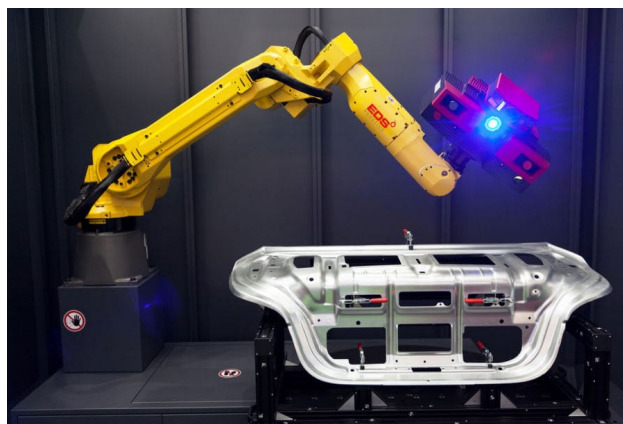


Figura 1: Robot de selección y de control de calidad

6. Beneficios de la visión artificial

El uso de la visión artificial en empresas trae consigo numerosas ventajas, ya que no solo pueden resolver cuestiones de fabricación, sino que también pueden ayudar a mejorar la productividad. Algunos son:

- **Precisión mejorada.** Por naturaleza, las decisiones humanas se alimentan de suposiciones o ideas preconcebidas sobre la verdad. En cambio, la visión artificial basa las decisiones en datos. Además, opera a nivel de píxel, algo que el cerebro humano no es capaz de procesar. Por tanto, los resultados obtenidos son mucho más precisos, logrando mejorar las tasas de detección de fallos hasta en un 90 %.
- **Mejora en productividad.** La visión artificial puede reconocer imágenes en segundos, pudiendo destinar esos recursos humanos a tareas que requieren de la mente para ejecutarse.
- **Reducción de costes.** Se puede mejorar la velocidad de fabricación y reducir la mano de obra necesaria para operar el equipo. Además, puede hacer que se reduzca la cantidad de desecho, reduciendo así la cantidad de materiales empleados y, en consecuencia, una disminución de costes.
- **Mejora la seguridad.** La visión artificial trae consigo una reducción de participación humana, por lo que fomenta la creación de un entorno más seguro en general. Los operarios de las industrias tienden a sufrir menos lesiones y accidentes cuando operan con máquinas potentes y voluminosas. Además, se consigue que los empleados no contaminen las salas limpias y se reduce su exposición a materiales y piezas peligrosas.

Por tanto, el empleo de la visión artificial en diferentes sectores permitiría una mejora en la productividad y una reducción de costes que toda empresa busca. [3]

7. Ejemplos de visión artificial

A continuación, veremos algunos ejemplos de tareas de visión artificial establecidas:

- La **clasificación de imágenes** permite ver una imagen y clasificarla (un coche, un plátano, la huella de un dedo), empleando redes neuronales convolucionales. Es más, es capaz de prever con mucha precisión que la imagen pertenece a una clase específica. Por ejemplo, una empresa de redes sociales, como puede ser Instagram, podría querer utilizarla para identificar y eliminar automáticamente imágenes censurables subidas a su plataforma por los usuarios.
- La **detección de objetos** puede emplear la clasificación de imágenes para relacionar una determinada clase de imagen, y luego detectar y tabular su apariencia en un vídeo o imagen. Para lograr la detección de objetos, se emplean técnicas de segmentación como las basadas en regiones, píxeles, modelos o bordes. Algunos ejemplos son la detección de daños en una línea de producción o la identificación de maquinaria que requiere mantenimiento.
- El **rastreo de objetos** sigue a un objeto una vez que se ha detectado. El rastreo de objetos normalmente se realiza con imágenes capturadas en secuencias o transmisiones de vídeo en tiempo real, utilizando el aprendizaje profundo o el filtro de Kalman junto a modelos de movimiento. Las cámaras de seguridad, por ejemplo, no solo tienen que clasificar y detectar personas sino que también necesitan seguirlas en movimiento para

saber si acceden a zonas en las que está prohibido el acceso. [4]

- La **recuperación de imágenes basada en contenido** utiliza la visión artificial para examinar, buscar y recuperar imágenes de grandes bases de datos, en función del contenido de las imágenes en lugar de las etiquetas de metadatos asociadas. Esta tarea puede incorporar la anotación de imágenes de forma automática, en vez del etiquetado manual convencional. Además, se pueden utilizar para sistemas de activos digitales y puede mejorar la precisión de la búsqueda y la recuperación.

Algunas de las técnicas que se pueden emplear en la visión artificial y que están basadas en el procesamiento de imágenes y vídeos son las siguientes:

- **Filtrado de imágenes:** La aplicación de filtros permite destacar o suavizar ciertas características de una imagen. El uso del filtro de suavizado gaussiano permite reducir el ruido de una imagen, como hemos visto durante la asignatura, mientras que el filtro de Canny es el más completo para la detección de bordes, muy usado para detectar contornos de objetos.
- La **transformación de imágenes** incluye operaciones como la rotación, escalado, cambio de perspectiva, etc. Gracias a estas operaciones, podemos corregir la orientación de una imagen o ajustar su tamaño según se necesite.
- **Segmentación de imágenes:** podemos dividir una imagen por regiones, esto es posible gracias a la segmentación basada en un umbral, donde se clasifican los píxeles en función de su nivel de intensidad, o mediante métodos más sofisticados como la segmentación semántica, capaz de asignar una etiqueta a cada píxeles en función de su clase.

- **Mejora de la calidad de imágenes:** es importante trabajar con imágenes y vídeos que presenten la mayor calidad posible. Para ello, se aplican técnicas como la eliminación de ruido, la mejora del contraste y la nitidez, y la corrección del balance de blancos.
- La **compresión de imágenes y vídeos** permite reducir el tamaño de los archivos notablemente sin perder demasiada calidad. Se utilizan algoritmos de compresión como JPEG para imágenes y H.264, H.265, entre otros, para vídeos.
- La **detección de texto** en vídeos e imágenes mediante el reconocimiento óptico de caracteres (OCR) resulta muy útil. Esto es posible gracias a algoritmos de detección de objetos, que pueden detectar regiones de la imagen que contienen texto para luego segmentar y reconocer cada uno de los caracteres.

Parte II

Caso práctico - Detección de Matrículas

8. Contexto y herramientas utilizadas

Para complementar este trabajo, vamos a crear un programa en `Python` que sea capaz de detectar matrículas. Para ello, realizaremos un procesamiento de imágenes de coches para tratar de identificar y dejar claro el recuadro de la matrícula; esto lo haremos con las librerías `cv2` [5] y `skimage` [6]. Luego, pasaremos esa imagen por un motor OCR, dado por la librería `pytesseract` [7].

9. Características de las imágenes del dataset

Nuestro dataset está compuesto por 10 imágenes a color cuadradas de 512x512 donde podemos visualizar una matrícula de un coche. Trabajaremos con la imagen 2 del dataset:



Figura 2: Imagen de un coche

Para cargar esta imagen en el programa, usamos la función:

```
img = cv2.imread("./dataset/coche1.png")
```

Para mostrar información de la imagen, usamos:

```
>>> print(img.shape)  
(512, 512, 3)
```

Esto nos dice que la imagen tiene 512 píxeles de ancho, 512 de alto y 3 canales (correspondientes a los colores rojo, verde y azul).



Figura 3: 3 canales de color

10. Imagen en escala de grises

Para procesar esta imagen, nos interesa que esta tenga 1 solo canal, es decir, que esté en escala de grises. Para ello, podemos usar:

```
imgBW = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Obteniendo como resultado la imagen 4:



Figura 4: Coche en escala de grises

Comprobemos los detalles de la nueva imagen:

```
>>> print(imgBW.shape)  
(512, 512)
```

Ahora tenemos 1 solo canal en la imagen.

11. Imagen binaria

El siguiente paso sería aplicarle un umbral a la imagen en escala de grises, para que, las matrículas, que son caracteres negros sobre un fondo blanco, queden marcadas y podamos detectar su contorno más fácilmente. La biblioteca `cv2` nos ofrece 2 alternativas:

```
value = 170  
imgBIN1 = cv2.threshold(imgBW, value, 255, cv2.THRESH_BINARY_INV)[1]
```

```
imgBIN2 = cv2.adaptiveThreshold(imgBW, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,  
                                cv2.THRESH_BINARY_INV, 9, 7)
```

`cv2.threshold` utiliza un umbral global fijo para toda la imagen, es adecuado para imágenes con iluminación uniforme es más rápido y es computacionalmente menos costoso.

En cambio, `cv2.adaptativeThreshold` utiliza un umbral que varía localmente según la región de la imagen, es más robusto para imágenes con variaciones de iluminación y es más lento debido al cálculo local del umbral.

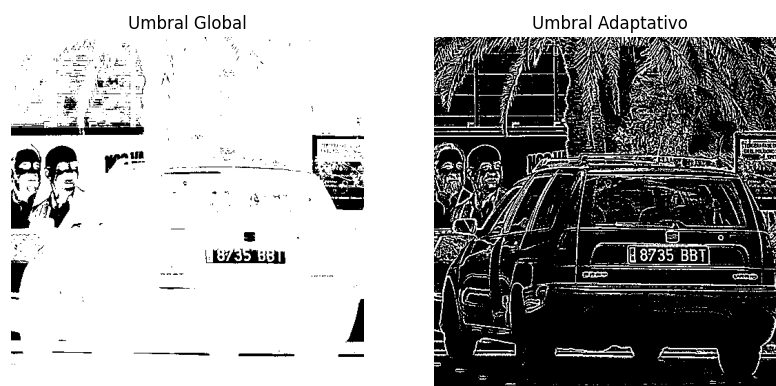


Figura 5: Umbral global y adaptativo

Como podemos observar, en ambos casos la matrícula queda bien marcada. En nuestro caso, usaremos `cv2.adaptativeThreshold`, que es con el que mejor resultados obtenemos.

12. Contornos

Ahora tenemos que encontrar los contornos de la imagen, que son curvas que unen todos los puntos continuos a lo largo de un límite, que tienen el mismo color o intensidad. Esto lo podemos realizar con:

```
contornos = cv2.findContours(imgBIN, cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)[0]
```

Al aplicar esa función a la imagen binaria, obtenemos la imagen 6:



Figura 6: Contornos del coche

12.1. Filtrar Contornos

Para filtrar los contornos primero tenemos que saber cuales son las medidas de las matrículas europeas, que son de 520mm x 120mm. A partir de ahí calculamos la relación de aspecto, que no es más que dividir 520 entre 120. También hemos añadido límites de ancho y alto para evitar que aparezcan contornos muy pequeños o muy grandes.

Entonces, iteramos sobre los contornos que hemos calculado antes y calculamos su relación de aspecto vemos si es parecida a la nuestra. También comprobamos que no sea ni inferior ni superior a los límites establecidos.

```
ratio = 520.0/120.0

min_w = 64
max_w = 256
min_h = 16
max_h = 64

candidatos = []

for c in contornos:
    x, y, w, h = cv2.boundingRect(c)
    aspect_ratio = float(w) / h

    if (np.isclose(aspect_ratio, ratio, atol=1.5) and
        (max_w > w > min_w) and
        (max_h > h > min_h)):
        candidatos.append(c)
```

En nuestro caso, obtenemos el siguiente candidato 7, que se corresponde con el contorno de la matrícula del coche:

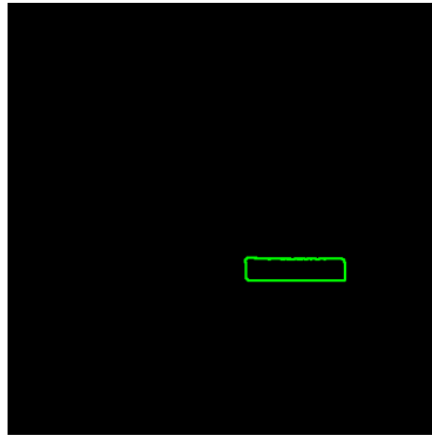


Figura 7: Contorno filtrado

12.2. ¿Qué ocurre si hay más de un candidato?

Ha habido coches donde el algoritmo nos ha detectado más de un candidato, bien porque la forma sea similar o bien porque el algoritmo no sea lo suficientemente restrictivo.

En este caso lo que se nos ha ocurrido es seleccionar el candidato más bajo, ya que las matrículas se suelen encontrar en la parte de abajo.

Esto no es lo más preciso y en otros casos puede fallar. Para mejorar esto se podría intentar perfeccionar el filtrado o dejar que la red neuronal se encargue de seleccionar el mejor candidato.

```
ys = [cv2.boundingRect(c)[1] for c in candidatos]
candidatoMenor = candidatos[np.argmax(ys)]
```

En la imagen 8 tenemos un ejemplo de este suceso:

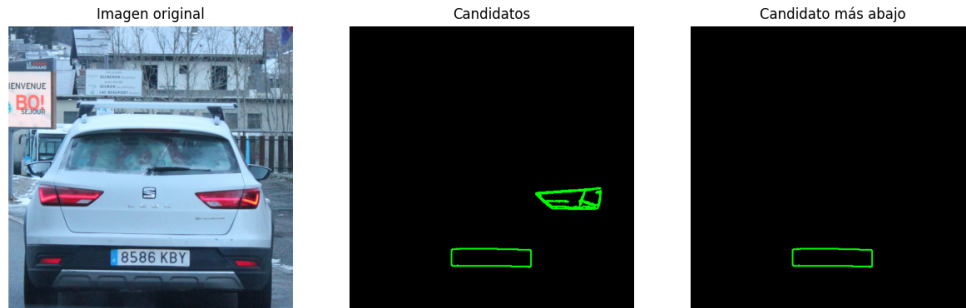


Figura 8: Varios candidatos

13. Recortar matrícula

En el siguiente paso, tenemos que recortar la parte de la matrícula con ayuda del contorno obtenido, para ello, podemos usar:

```
x, y, w, h = cv2.boundingRect(candidatoMenor)
matricula = img[y:y+h,x:x+w]
```

Al aplicar esa función obtenemos la imagen 9:



Figura 9: Matrícula

14. Procesar matrícula

Al igual que hicimos con el coche, tenemos que pasar la imagen de la matrícula a blanco y negro para posteriormente binarizarla.



Figura 10: Matrícula binarizada

El resultado de la binarización puede ser tener un poco de ruido en los bordes, así que utilizamos una función de `skimage` para eliminarlos:

```
matriculaSinBordes = skimage.segmentation.clear_border(matriculaBIN)
```

Obtenemos como resultado la imagen 11:



Figura 11: Matrícula sin ruido en los bordes

Ahora tenemos que invertir la imagen para que `pytesseract` la pueda analizar correctamente:

```
matriculaFinal = cv2.bitwise_not(img)
```

La imagen 12 es el resultado final del procesamiento para poder leer matrículas:

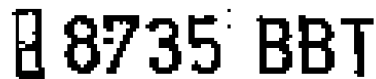


Figura 12: Matrícula invertida. Resultado final.

15. Extracción del texto

El primer paso es configurar `pytesseract` para que solo nos reconozca las letras y números que contienen la matrícula europea y no nos detecte cualquier otro símbolo. En este caso, queremos reconozca todas las letras excepto las vocales, la \tilde{N} y la Q ; y todos los números:

```
alphanumeric = "BCDFGHJKLMNPSTVWXYZ0123456789"  
options = f"-c tesseract_char_whitelist={alphanumeric} -- psm 7"  
txt = pytesseract.image_to_string(img, config=options)
```

Las matrículas a reconocer tienen un símbolo a la derecha de la unión europea, el cual `pytesseract` puede detectar como quiera. Entonces, la estrategia para devolver la cadena referente a la matrícula será recorrer la variable `txt` a la inversa, para así detectar las

3 primeras letras que aparezcan y los primeros 4 números que siguen a dichas letras. En el caso que no detectemos todos los números o todas las letras, completamos con ?.

```
txt = txt[::-1]

letras = ""

numeros = ""

counter = 0

for t in txt:

    if ((t in "BCDFGHJKLMNPSTVWXYZ") and (counter < 3)):

        letras += t

        counter += 1

    if ((t in "0123456789") and (3 <= counter < 7)):

        numeros += t

        counter += 1

letras += "?"*(3 - len(letras))

numeros += "?"*(4 - len(numeros))

matricula = numeros[::-1] + " " + letras[::-1]
```

En el caso de ejemplo, la matrícula que reconoce es: 8586 KBY, que se corresponde con la del coche de la imagen 2.

16. Resultados y conclusiones

Hemos realizado unas pruebas con un dataset de 10 imágenes de coches y estos son los resultados:

ORIGINAL	----	RESULT	-----	SIMILITUD(%)
8735 BBT	----	8735 BBT	-----	100.0%
7206 KDF	----	7206 KDF	-----	100.0%
5683 JZG	----	5533 JZC	-----	62.5%
4386 LJP	----	7385 LJP	-----	75.0%
4991 KXN	----	7991 KXN	-----	87.5%
9723 LCP	----	9723 LCP	-----	100.0%
8586 KBY	----	8586 KBY	-----	100.0%
8206 MCS	----	8206 MCS	-----	100.0%
0378 LKF	----	0373 LKF	-----	87.5%
0798 MNC	----	0798 MNC	-----	100.0%

Coches totales:	10
Numero de aciertos:	6
Numero de fallos:	4
Porcentaje simbolos erroneos:	8.75%

Los resultados son bastante buenos, ha habido 6 casos donde ha hallado la matrícula correctamente, 2 casos donde ha fallado por un símbolo y otros 2 casos donde ha fallado por más de 1 símbolos.

En todos los casos ha sido capaz de distinguir la forma de la matrícula, por lo que los fallos

se encuentran en la extracción del texto. Vemos que confunde algunos 8 con 3, el número 4 con la letra A, la G con la C,... En resumen, confunde caracteres similares. Esto puede deberse al tipo de fuente de las matrículas (con la cual **pytesseract** no está entrenado) y a que **pytesseract** no sabe el formato que tiene que reconocer y nos devuelve un string de todo lo que ve; si supiese que los 4 primeros símbolos son números, entonces no confundiría el 4 con la A.

Una posible solución sería entrenar un modelo propio que detecte perfectamente la tipografía y la estructura de las matrículas europeas o tratar de minimizar aun más el ruido de la imagen a partir de la cual se hace la extracción y partirla en 2 partes, la primera con solo números y la segunda con solo letras.

Parte III

Anexo

17. Repositorio de GitHub

Este proyecto ha sido realizado en un repositorio de GitHub, donde se puede encontrar el código fuente, todas las imágenes empleadas, los test realizados y el código Latex de este pdf. El enlace es el siguiente:

[Enlace al repositorio](#)

Referencias

- [1] Mordor Intelligence. *Mercado de la visión artificial*. 2024. URL: <https://www.mordorintelligence.com/es/industry-reports/machine-vision-systems-market>.
- [2] Rostyslav Demush. *A Brief History of Computer Vision (and Convolutional Neural Networks)*. 2019. URL: <https://hackernoon.com/a-brief-history-of-computer-vision-and-convolutional-neural-networks-8fe8aacc79f3>.
- [3] Lorena Ramírez. *Visión artificial en la industria 4.0: ¿Qué es y qué aplicaciones tiene?* 2023. URL: <https://www.iebschool.com/blog/vision-artificial-en-la-industria-4-0-que-es-big-data/>.
- [4] James Le. *The 5 Computer Vision Techniques That Will Change How You See The World*. 2018. URL: <https://heartbeat.comet.ml/the-5-computer-vision-techniques-that-will-change-how-you-see-the-world-1ee19334354b>.
- [5] andrey.senyaev. *opencv-python 4.9.0.80*. 2023. URL: <https://pypi.org/project/opencv-python>.
- [6] Emmanuelle. *scikit-image 0.23.2*. 2024. URL: <https://pypi.org/project/scikit-image>.
- [7] madmaze. *pytesseract 0.3.10*. 2022. URL: <https://pypi.org/project/pytesseract>.