

# Revisión de métodos y algoritmos para tratamiento de imagen y segmentación previo a la implementación de OCR

Review of methods and algorithms for image processing and segmentation prior to OCR implementation

Cristian David Mojica Cabeza<sup>1\*</sup>

<sup>1</sup> Corporación Centro de Desarrollo Tecnológico del Gas,  
Parque Tecnológico UIS Guatiguará, km 2 vía El Refugio, Piedecuesta, Colombia.

\* Autor de contacto. Correo electrónico: [cmojica@cdtdegas.com](mailto:cmojica@cdtdegas.com)

## Resumen

Teniendo en cuenta el auge de las tecnologías de tratamiento de imagen y la madurez de los algoritmos existentes que permiten una mejor implementación de la tecnología de reconocimiento óptico de caracteres (OCR), se hace importante realizar una revisión de estos métodos y algoritmos para contar con una cantidad considerable de opciones al momento de elegir la más adecuada. Por ende, se realiza una clasificación de las tecnologías en dos categorías: tratamiento de imágenes digitales y segmentación, exponiendo las ecuaciones asociadas a los diferentes algoritmos, así como el efecto práctico de estos sobre las imágenes tratadas. Se concluye con un gran abanico de posibilidades tecnológicas, capaces de funcionar en algunos casos de manera conjunta, permitiendo así el mejor tratamiento posible de la imagen previo a la aplicación de OCR.

**Palabras Clave:** *Segmentación, Tratamiento de imagen, OCR.*

## Abstract

Taking into account the boom in image processing technologies and the development of existing algorithms that allow a better implementation of optical character recognition (OCR), it is important to review these methods and algorithms in order to grow the number of options and choose the best one. Therefore, a classification of the technologies in two categories is made: digital image processing and segmentation, the equations associated with the algorithms are presented, as well as their practical effect on the processed images. Concluding with a wide range of possibilities that for some cases may be implemented together, allowing the best possible treatment for the image prior to the application of OCR.

**Keywords:** *Segmentation, Image processing, OCR.*

## TECNOLOGÍA

### 1. Introducción

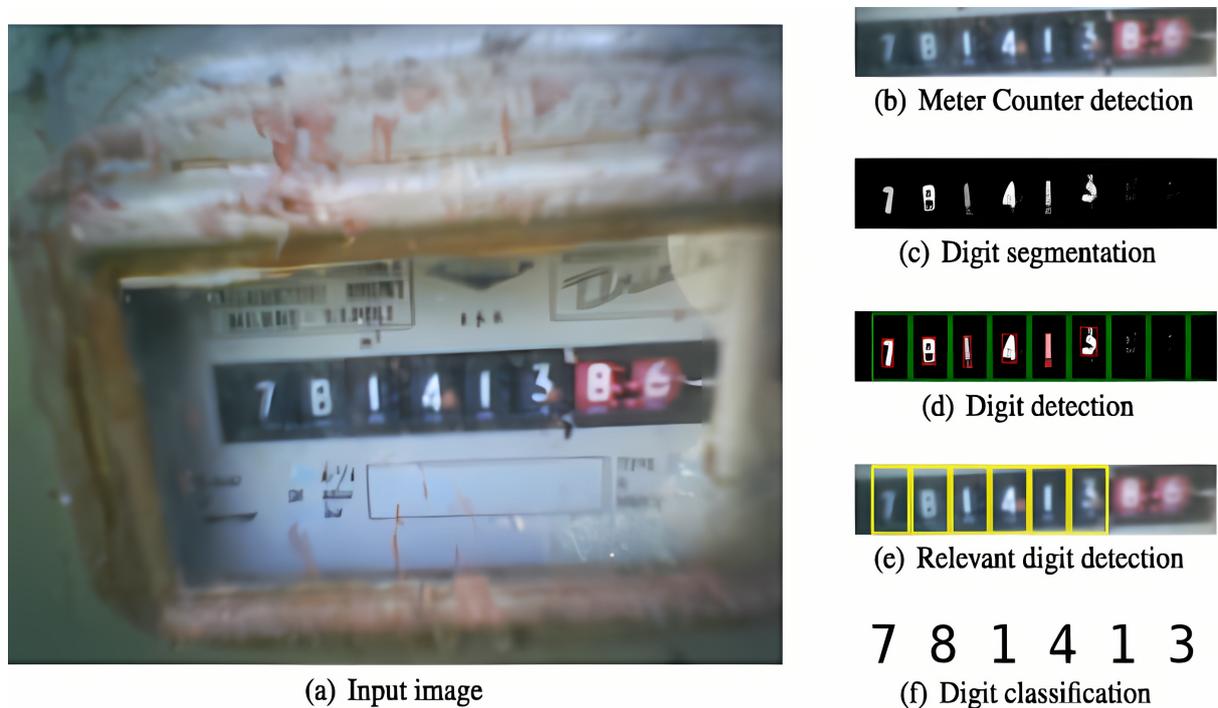
Los algoritmos de reconocimiento óptico de caracteres, OCR por sus siglas en inglés, se utilizan para mejorar procesos existentes basados en datos analógicos, que requieren ser comunicados y almacenados de manera digital. Entre los diferentes campos de aplicación de la tecnología OCR, se encuentran, el reconocimiento de texto manuscrito, el reconocimiento de matrículas y la indexación en bases de datos. Por otra parte, el uso de esta tecnología en la industria permite la digitalización de la información, particularmente en la industria del gas, en donde se requiere procesar mediciones provenientes de equipos que no cuentan con una salida digital (computador de flujo). En general, la exactitud y el buen desempeño de un sistema OCR, dependen del preprocesamiento de la imagen y del algoritmo de segmentación que se emplee para extraer la información. La cual, finalmente alimenta un algoritmo basado en inteligencia artificial capaz de realizar OCR desde una fuente de entrada analógica. Por consiguiente, se requiere refinar estos procedimientos previos para asegurar la mejor calidad de imagen posible, que permita generar los mejores resultados, es decir, reconocer correctamente los caracteres presentes en la imagen [1].

El preprocesamiento es una fase necesaria para poder implementar un algoritmo de OCR. Tiene como fin mejorar la calidad de la imagen a procesar, o llevarla a ciertos valores estándar requeridos por procedimientos posteriores, asegurando así la repetibilidad/normalización de los entrenamientos y posterior empleo de los algoritmos entrenados, haciendo más fáciles las fases subsiguientes del procesamiento de imagen para OCR, principalmente la segmen-

tación. Entre los cambios más significativos se encuentran el re-escalado, la corrección de perspectivas o inclinaciones, y diferentes tratamientos que pretenden mejorar la calidad de la imagen en cuanto a resolución [2]. También existen preprocesamientos que tienen como finalidad remover ruido o información no relevante de la imagen, y, finalmente, cambios más drásticos como pasar la imagen a escala de grises o blanco y negro [3].

Por otra parte, dentro del alcance de la tecnología OCR, el algoritmo de segmentación tiene como objetivo separar una parte de la imagen que es de interés del resto, como se puede apreciar en la Figura 1. Para esto, se vale de técnicas de diferenciación como la umbralización, proceso mediante el cual se binariza la imagen, tomando como referencia un valor umbral de intensidad, por encima del cual se considera blanco y de lo contrario negro, lo que reduce la complejidad de la imagen, pasando de escala de grises a tener solo dos valores: blanco y negro. Dicha umbralización también puede emplearse en algoritmos un poco más complejos que permiten, entre otros, encontrar los contornos que definen un objeto para poder separarlo de los demás, esto es, encontrar las coordenadas que definen la posición del objeto (entendiéndose la imagen como un plano bidimensional) y posteriormente recortar la imagen con base en esas coordenadas [4].

Los modelos que son más empleados al momento de realizar monitoreo o cálculos en tiempo real son los empíricos, debido a que en éstos las variables que normalmente no varían en el tiempo, anteriormente mencionadas (como las dimensiones del horno), se estiman por medio de constantes en un modelo que presenta como



**Figura 1.** Demostración del proceso general de OCR. Recuperada de [5].

entradas una cantidad limitada de variables y como salida la o las variables de interés para el proceso, en muchos casos, la eficiencia. Estos modelos usan como entrada para su entrenamiento los datos obtenidos de otros modelos que tienen en cuenta la mayor cantidad de datos para generar aproximaciones bastante buenas a partir de regresiones y, en la actualidad, modelos entrenados con inteligencia artificial [41].

En el presente documento se presentan los diferentes algoritmos que pueden utilizarse para el preprocesamiento de imagen, previo a la aplicación final del OCR para la extracción de información, y su clasificación se expone en la Figura 2.

## 2. Tratamiento de imágenes digitales

Para posibilitar un tratamiento digital de una

imagen, se requiere que inicialmente dicha imagen se guarde en un formato compatible (usualmente JPEG o PNG). Teniendo la imagen en los formatos PNG y JPEG se posibilita la representación de la misma como una matriz numérica de dimensiones  $M \times N$ , donde  $M$  y  $N$  son las dimensiones en píxeles de la imagen, que se pueden entender como la cuantización de intensidad o nivel de gris. Si la imagen es en escala de grises, solo existe una matriz asociada, con valores de representación de intensidad que varían de 0 a  $2n-1$ , siendo uno de los más usados  $n$  igual a 8; esto genera un rango de valores entre 0 y 255. En esta escala, 0 es el negro absoluto y 255, blanco absoluto. Dicho valor de 8 corresponde a bits de computador, que permiten codificar 256 valores distintos desde la combinación 00000000, que representa el nivel 0, hasta la combinación 11111111, que se corresponde con el nivel 255 [6].

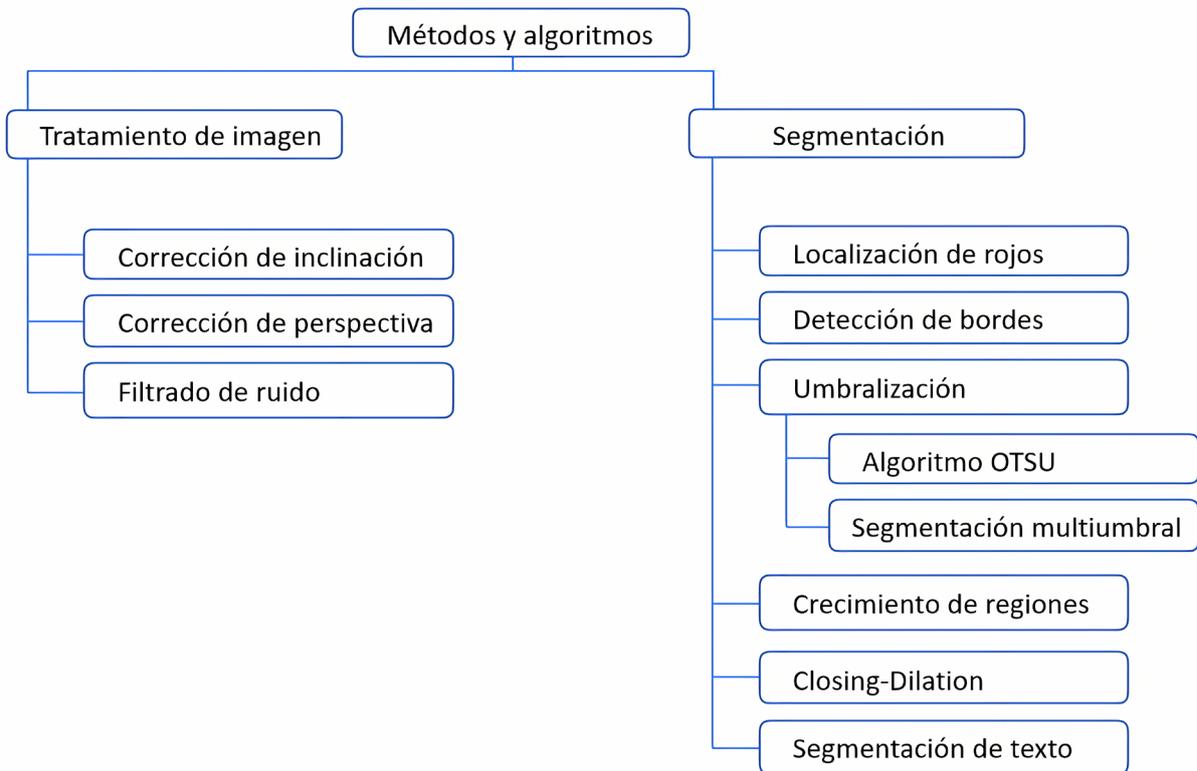
Para el caso de una imagen a color, la codificación más empleada es la RGB (siglas en inglés de Red, Green, Blue), cada posición de la matriz tiene asociados 3 valores, o también puede entenderse como una imagen codificada en 3 matrices diferentes, una para cada uno de los componentes principales de color, rojo (R), verde (G), y azul (B). Para este caso el conjunto de valores para el negro sería (0, 0, 0), mientras que el blanco sería (255, 255, 255). La combinación de distintos valores representa otros colores [6].

Otra forma de apreciar la distribución de la intensidad de una imagen es por medio de su histograma, una función discreta que representa el número de píxeles en la imagen para cada nivel

de intensidad,  $g$ . Usualmente se emplea para imágenes en escala de grises, pero también pueden analizarse por separado los histogramas de los componentes principales de color (RGB). En donde, la probabilidad de ocurrencia de un determinado nivel se define como se expresa en la Ecuación 1.

$$P(g) = \frac{N(g)}{M} \quad (1)$$

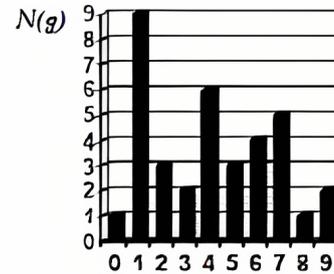
Donde  $M$  es el número de píxeles en la imagen y  $N(g)$  es el número de píxeles en el nivel de intensidad  $g$ . Como con cualquier distribución de probabilidad, todos los valores de  $P(g)$  son menores o iguales que 1 y la suma de todos los valores de  $P(g)$  es 1. Las intensidades o nive-



**Figura 2.** Clasificación de los diferentes métodos y algoritmos expuestos en el documento.

0	3	3	4	4	4
1	1	1	4	4	5
1	1	1	4	5	5
1	1	1	7	6	6
2	2	7	7	6	6
8	2	7	7	9	9

(a)



**Figura 3.** (a) Imagen con 10 niveles de gris del 0 al 9; (b) Histograma correspondiente. Adaptada de [6].

les de gris están representadas a lo largo del eje X y la frecuencia se representa en el eje Y. Por ejemplo, dada la imagen en el panel (a) de la Figura 3 en formato matriz digital basada en los pesos por pixel, con 10 niveles de gris del 0 al 9, su histograma se muestra en el panel (b) de la misma figura.

## 2.1. Corrección de inclinación

Dentro del preprocesamiento de las imágenes para una posterior aplicación de un algoritmo de OCR, uno de los principales problemas, para el caso de reconocimiento de secuencias de caracteres (dentro de los que resalta el caso de texto plano), es la inclinación de la imagen. Para esto, en [7] se propuso un filtro que corrige la inclinación de la imagen encontrando el ángulo, y de manera adicional compara matemáticamente los ángulos al final para verificar el proceso. Para tal propósito, se usa un análisis del espectro de la imagen, que se obtiene aplicando un filtro que resalta de forma brillante los puntos que muestran una misma inclinación respecto a los ejes 'x' y 'y', es decir los pixeles que forman líneas rectas. Para aumentar la fiabilidad del proceso, se divide la imagen en los cuatro

cuadrantes cartesianos, y se omiten los pixeles cercanos al centro, ya que se ha determinado empíricamente que estos siempre se resaltan con el filtro, de manera que pueden llegar a alterar los resultados, que consisten en generar una recta a partir de los puntos brillantes generados (15-20), y posteriormente promediar las 4 rectas obtenidas para estimar el ángulo promedio de inclinación [7].

Otra manera de abordar el problema de inclinación de imágenes de texto escaneadas, expuesta por [8], es buscar el borde lateral del párrafo del texto (como se puede observar en la Figura 4) o de una estructura que se pueda identificar, y de esta manera se alinea ésta con el eje 'y' para generar una nueva imagen que tiene corregida la inclinación. Para lograrlo, se pasa la imagen por un filtro de binarización (convertirla a blanco y negro) y el primer pixel negro detectado en un barrido desde la izquierda cada k líneas se guarda como coordenada en una matriz para posteriormente generar una línea recta con las diferentes coordenadas obtenidas. Se obtiene un mejor resultado cuanto menor es el valor de k, pero evidentemente se vuelve más lento el proceso [8].

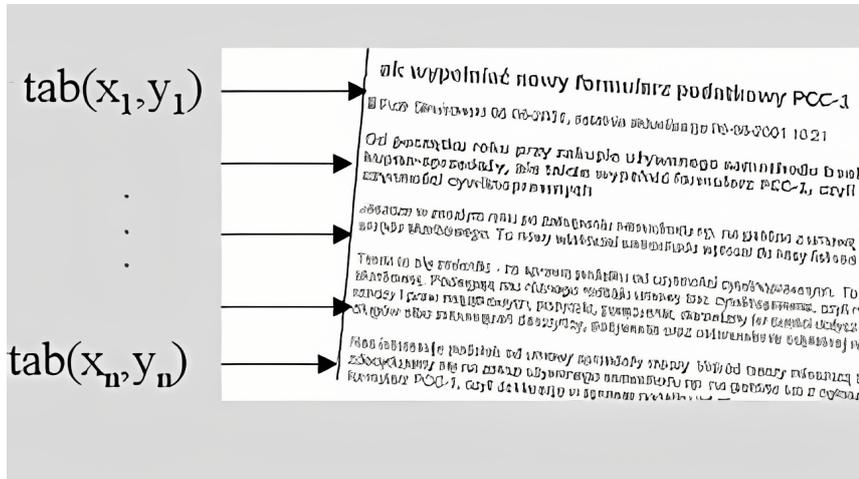


Figura 4. Identificación de la referencia de la inclinación. Recuperada de [8].

$$Tab_1 = \{(x_k, y_k) : \forall i \dots n, x_{i-1} > x_1\} \quad (2)$$

$$Tab_2 = Tab / Tab_1$$

Sin embargo, para evitar la generación de referencias erróneas se deben tener en cuenta ciertas correcciones; inicialmente se empieza por descartar/seleccionar cuales puntos son realmente representativos de la referencia que se quiere obtener, como se observa en la Figura 5.

En el caso de una rotación en sentido de las agujas del reloj, se construyen dos conjuntos de puntos que satisfagan la condición dada por la ecuación 2.

Luego se elige la que representa la línea más vertical, o sea, el más grande de los dos conjuntos. De este conjunto se extraen las coordenadas de los extremos con respecto al eje x, y una vez se tiene este dato, encontrar el ángulo de la inclinación se vuelve algo trivial como se observa en la Figura 6. Finalmente, se rota la imagen de acuerdo al ángulo de inclinación definido [8].

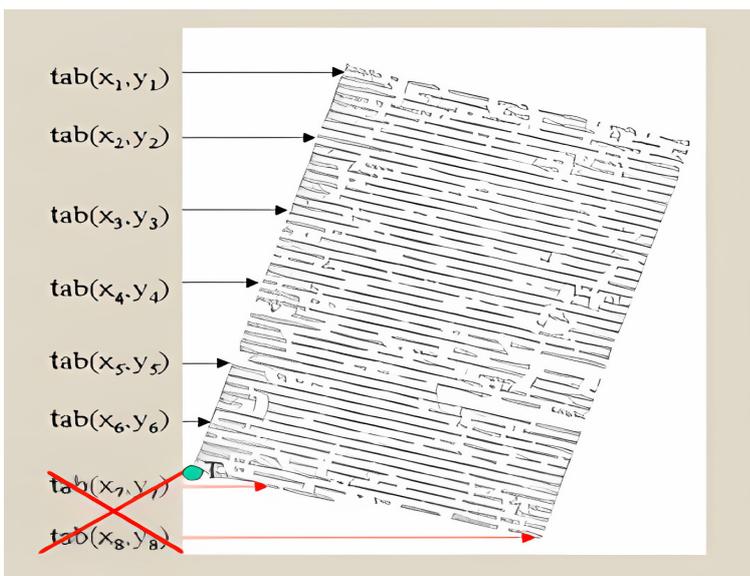
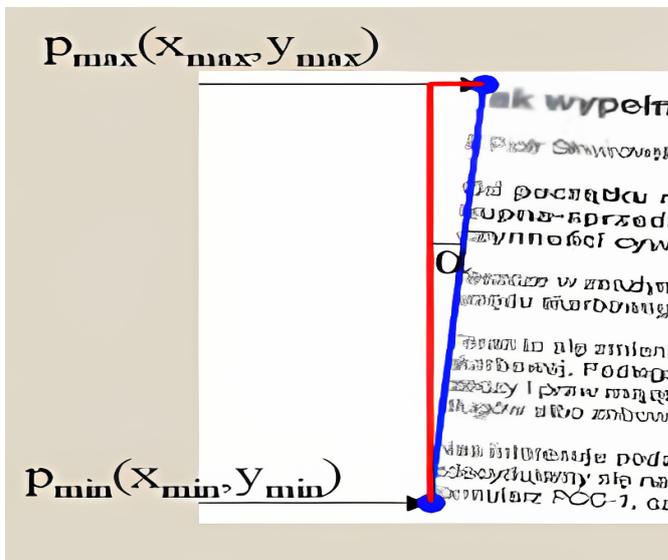


Figura 5. Selección de los puntos que pertenecen a la línea de referencia. Rotación del texto en sentido de las agujas del reloj. Recuperada de [8].



**Figura 6.** Localización de los puntos extremos de la recta. Recuperada de [8].

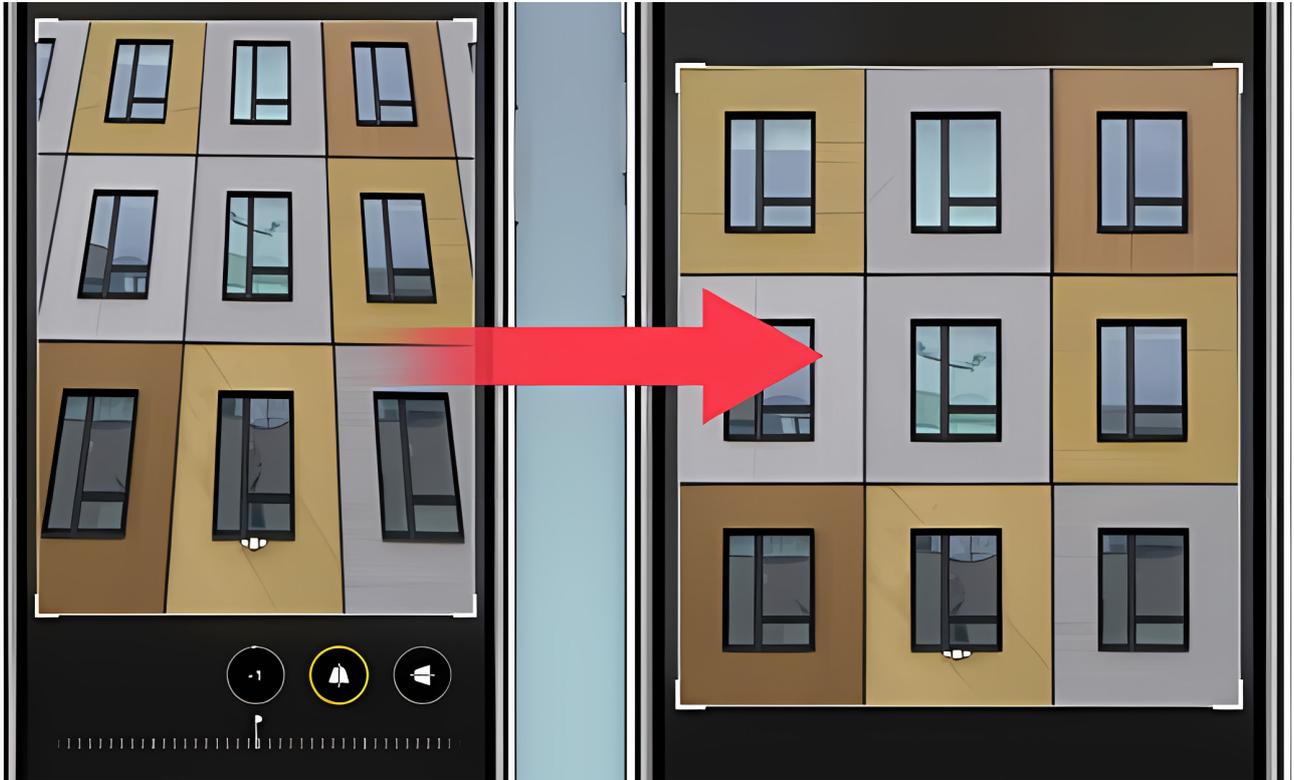
## 2.2. Corrección de perspectiva

Cuando el dispositivo de captura de la imagen no está posicionado de manera ortogonal al objetivo que se pretende capturar, se resulta obteniendo una visión más de tipo trapecoide o paralelogramo de lo que debería ser un rectángulo. Para corregir este problema, el primer paso es identificar la zona que contiene el texto, esto se puede hacer de manera manual o con el método empleado anteriormente para determinar los bordes [8]. Recientemente se ha evaluado la posibilidad de implementar la corrección de perspectiva por medio de redes neuronales que la realicen de manera automática, detectando el ángulo de inclinación o el efecto de ojo de pez en distintos tipos de imágenes -no solo de texto- para corregirlo y detectar correctamente el objeto que se requiere, como se aprecia en la Figura 7 [9].

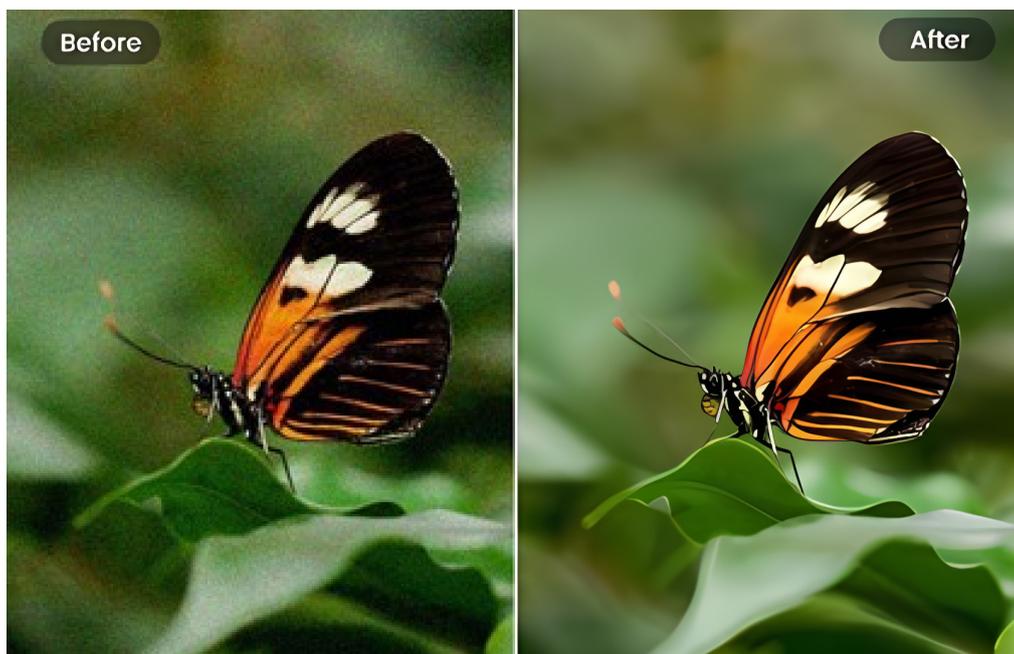
## 2.3. Filtrado de ruido

Este tipo de pre procesamiento consiste en la

aplicación de los filtros adecuados para poder obtener una imagen limpia para su posterior procesamiento. Para ello, se usan filtros de suavizado de la imagen de segmentación por umbralización y filtros morfológicos (que se emplean para resaltar o suavizar ciertas características de la imagen) [10]. El filtro de suavizado consiste en un primer limpiado de todos aquellos píxeles cuyo nivel de intensidad es muy diferentes al de sus vecinos y que causen un efecto de desenfoque que no sea propicio para la lectura y procesamiento de la imagen, como se observa en la Figura 8. A su vez, la umbralización de la imagen se encarga de intensificar el contraste de la imagen con su entorno para poder desde este punto realizar una pseudo segmentación en razón a las áreas que son de interés [11]. Actualmente existen páginas o aplicaciones que ofrecen un servicio de filtrado de ruido o denoise, dentro de una amplia gama de servicios de tratamiento de imágenes, como remover marcas de agua u objetos y voltear imágenes [12].



**Figura 7.** Corrección de perspectiva generada por una aplicación de celular (SKRWT) Fuente: Autores.



**Figura 8.** Antes y después de aplicar el filtro de ruido a una imagen capturada por cámara. Fuente: Autores.

### 3. Segmentación

Muchos de los métodos que van a ser expuestos a continuación, se pueden emplear progresivamente para constituir un procedimiento más robusto. Sin embargo, deberán tenerse en cuenta diferentes implicaciones de capacidad de cómputo o adecuamiento de los datos, por lo que se debe evaluar la pertinencia y aplicabilidad de los mismos.

Entre los métodos más empleados se encuentran la segmentación basada en detección de bordes, esquinas o cambios de distintos parámetros de la imagen para definir los contornos de las zonas de interés y segmentarlas. También se emplean redes neuronales basadas en discriminación de texturas; sin embargo, para estos casos el ruido de la imagen puede llegar a ser problemático [5].

#### 3.1. Localización de rojos

Dentro del OCR aplicado a medidores de flujo de agua, gas o electricidad existe, en algunos casos puntuales, una particularidad aprovechable,

la coloración roja sobre los últimos tres dígitos de la indicación, correspondientes a la parte decimal de la cantidad de metros cúbicos consumidos (00 000,000). Teniendo esto en mente, se han dirigido esfuerzos para realizar una segmentación basada en esta particularidad, se realiza un recorrido sobre los píxeles de la imagen obteniendo los valores RGB y se calcula el porcentaje de rojo sobre verde (PRG) y de rojo sobre azul (PRB) de acuerdo con la Ecuación 3 [5].

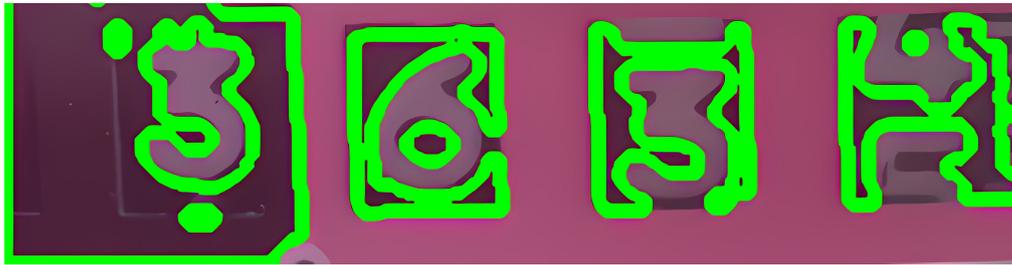
$$PRG = \frac{R}{G+\epsilon}; \quad PRB = \frac{R}{G+\epsilon} \quad (3)$$

Donde  $\epsilon = 10^{-16}$  para evitar divisiones por cero.

Entonces, una vez calculados los valores anteriores, un píxel se marca como rojo si tanto PRG como PRB son simultáneamente superiores a la unidad. En la Figura 9 se observa la imagen original y marcado en verde en la derecha los píxeles reconocidos como rojos [6].



**Figura 9.** Localización de parte roja del contador. Recuperada de [6].



**Figura 10.** Ejemplificación de resultados de detección de bordes.

### 3.2. Detección de bordes

La detección de bordes se realiza con un entrenamiento exhaustivo del algoritmo de red múltiple para la detección de objetos (MNOD, por sus siglas en inglés), en el que cada nodo se entrena en un conjunto de puntos extraídos de las imágenes de la base de datos empleando técnicas de inteligencia artificial como redes neuronales. En general, cada punto tiene la estructura (características, etiqueta), donde las características están ligadas a los parámetros que definen el punto y el valor esperado es el etiquetado correspondiente [13]. Posteriormente, se realiza un análisis de Fourier en la imagen previamente segmentada, para encontrar la fase y el periodo de la secuencia de dígitos. En este procedimiento, dada una imagen, se computa una señal obtenida al integrar localmente la imagen usando un filtro deslizante. Aplicando un análisis de Fourier a la señal, se puede obtener el componente con mayor frecuencia. Entonces, la coordenada horizontal de cada dígito, co-

rresponde o se encuentra muy cerca del máximo. Finalmente, la región se divide en regiones candidatas que contienen potencialmente un dígito. En la Figura 10 se ejemplifican los resultados de la conformación de las regiones candidatas [5]. Las aplicaciones de esta tecnología no se limitan a detección de dígitos u objetos en imágenes; en el ámbito médico, se emplea la detección de bordes para el procesamiento de imágenes en las que el ojo humano difícilmente detectaría pequeños cambios que pueden estar asociados a patologías específicas, resaltando la aplicación de segmentación sobre resultados de rayos X [14].

### 3.3. Umbralización

La umbralización (thresholding) ayuda a diferenciar claramente las zonas de interés, permitiendo así una segmentación de la imagen o de las zonas de la imagen que serán posteriormente separadas para usarse como entrada en el proceso de OCR [2].



**Figura 11.** Ejemplificación del proceso de thresholding.

El propósito de este procedimiento es diferenciar las zonas presentes en la imagen. Se pueden emplear algoritmos locales/adaptativos y globales. Para el global, se usa el mismo límite para toda la imagen, y en los locales este límite se modifica para las diferentes áreas definidas. Dicho límite funciona de la siguiente manera: se define un número asociado a la intensidad, respecto al que los valores encontrados por encima de este adquieren un valor máximo y los que se encuentren por debajo un valor mínimo, que corresponderían a negro y blanco respectivamente [2]. En la Figura 11 se aprecia el cambio de una serie de imágenes en escala de grises a su versión binarizada.

Para el caso de los algoritmos que hacen un umbralizado local, se emplea una lectura del histograma de intensidad en donde para las diferentes zonas se extrae el valor del límite que definirá la pertenencia del pixel al fondo o "frente" de la imagen teniendo en cuenta un promedio definido a partir del histograma. A dicho histograma se le puede aplicar una transformación conocida como ecualización, que pretende obtener un histograma con una distribución uniforme, es decir, que cada nivel de gris tenga el mismo número de pixeles, obteniendo como resultado una maximización del contraste de la imagen sin perder información estructural, facilitando la aplicación del umbralizado [15].

### 3.3.1. Algoritmo OTSU

Este algoritmo, nombrado en honor a un investigador japonés, trabaja con la premisa de normalizar los valores de intensidad de los pixeles y almacenar dichos valores en una matriz, para ello se hace uso del promedio y la varianza y se asignan a los pixeles valores de 0 o 1 (fondo o frente) con el objetivo de delimitar las regiones. Es decir, solo se cambia una vez la imagen. Para calcular la media y la varianza, los pixeles se dividen en dos clases, C1 con niveles de intensidad [1, ..., t] y C2 con niveles de intensidad [t+1,

..., L]. La distribución de probabilidad para las dos clases sería [16]:

$$C_1: p_1/w_1(t), \dots, p_t/w_1(t) \text{ y} \quad (4)$$

Donde,

$$C_2: p_{t+1}/w_2(t), \dots, p_L/w_2(t) \quad (5)$$

$$w_1(t) = \sum_{i=1}^t p_i \quad (6)$$

$$w_2(t) = \sum_{i=t+1}^L p_i \quad (7)$$

Entonces, la media se calcula como:

$$\mu_1 = \sum_{i=1}^t ip_i/w_1(t) \quad (8)$$

$$\mu_2 = \sum_{i=t+1}^L ip_i/w_2(t) \quad (9)$$

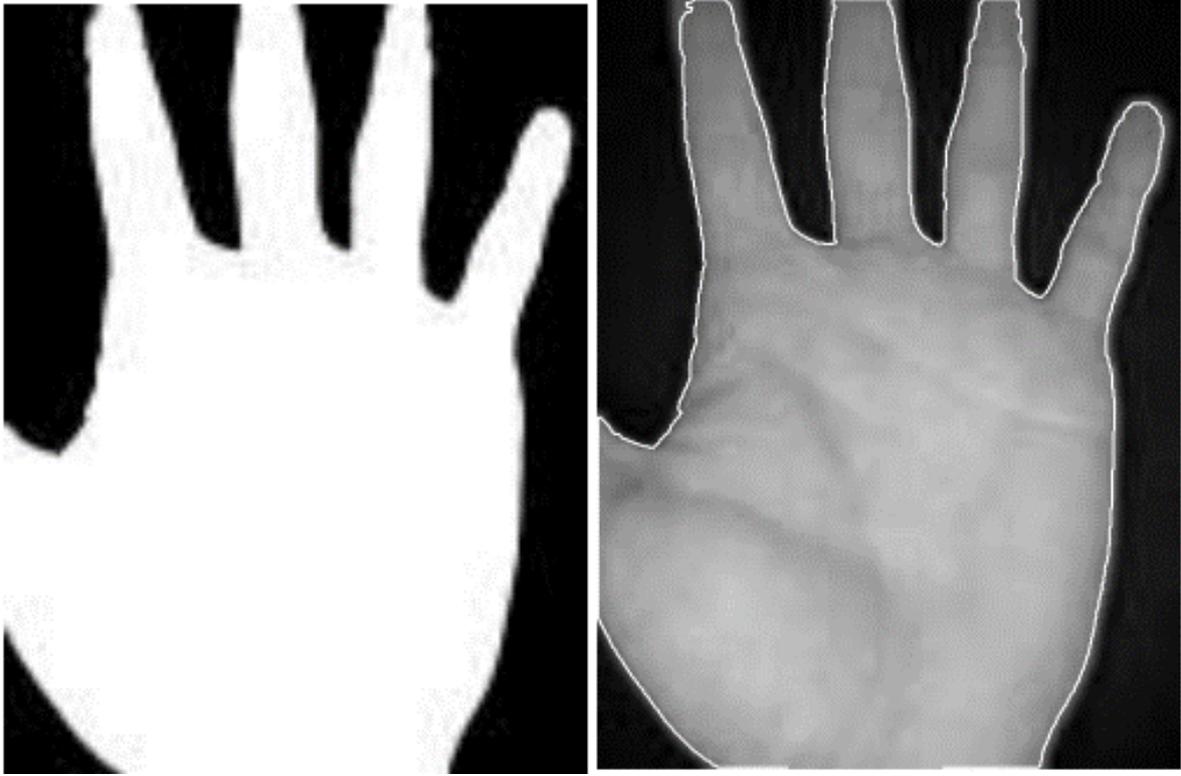
Empleando un análisis discriminativo, se define la varianza de la imagen umbralizada como:

$$\sigma_B^2 = w_1(\mu_1 - \mu_T)^2 + w_2(\mu_2 - \mu_T)^2 \quad (10)$$

Para un umbralizado bidimensional, el valor óptimo de  $t^*$  que maximiza la varianza está dado por:

$$t^* = \left\{ \sigma_B^2(t) \right\} \quad (11)$$

Para el caso de aplicaciones puntuales donde se pueda llegar a tener control sobre la iluminación y se requiera de mayor fiabilidad puede llegar a definirse el límite del umbralizado de manera manual con prueba y error [6]. En la Figura 12 se observan los resultados de binarización obtenidos por [17].



**Figura 12.** Umbralización: (a) Imagen binaria OTSU, (b) Imagen original con el borde OTSU. Adaptado de [17].

### 3.3.2. Segmentación multiumbral

En la segmentación multiumbral, se determinan los umbrales adecuados por medio del histograma de la imagen, donde se observan picos y valles que se corresponden con las múltiples regiones que constituyen la región general de interés. A cada valle del histograma se asocia un valor umbral. Todos los píxeles con un valor menor al de un umbral son asignados al segmento correspondiente, como se define en la Ecuación 12.

$$g(x, y) = \begin{cases} 0 & \text{Si } v(x, y) < t_1 \\ 1 & \text{Si } t_1 \leq v(x, y) < t_2 \\ \vdots & \vdots \\ n & \text{Si } t_n \leq v(x, y) \end{cases} \quad (12)$$

Donde cada segmento si corresponde a una región de la imagen.

Se definen entonces los valores admisibles de tonos de gris  $\lambda_1, \lambda_2, \dots, \lambda_L$ . La probabilidad discreta de cada tono de gris se puede determinar a partir de su histograma de frecuencias relativas. Para un tono de gris  $I_{ij}$  en el punto  $(i, j)$  de la imagen, la probabilidad  $P_v$  de que el punto  $(i, j)$  tenga el tono de gris  $\lambda_v$ , está dada por la Ecuación 13 [18].

$$P(I_{ij} = \lambda_v) = P_v = \frac{\text{Nro. de puntos de la imagen con tono } \lambda_v}{\text{Nro. total de puntos en la imagen}} \quad (13)$$

Con base en esta probabilidad, cada valor de umbral induce diferentes distribuciones, lo que hace posible calcular también la incertidumbre que encierra cada una de esas distribuciones de probabilidad (entropía). La cual vendría dada por la Ecuación 14:

$$\Psi(l_1, l_2, \dots, l_k) = \sum_{v=1}^{l_1} P_v + \sum_{v=l_1+1}^{l_2} P_v + \dots + \sum_{v=l_{k-1}+1}^L P_v$$

$$- \frac{\sum_{v=1}^{l_1} P_v P_v}{\sum_{v=1}^{l_1} P_v} - \frac{\sum_{v=l_1+1}^{l_2} P_v P_v}{\sum_{v=l_1+1}^{l_2} P_v} - \dots - \frac{\sum_{v=l_{k-1}+1}^L P_v P_v}{\sum_{v=l_{k-1}+1}^L P_v} \quad (14)$$

La finalidad es encontrar el conjunto  $l_1, l_2, \dots, l_k$  de umbrales que maximicen  $\Psi(l_1, l_2, \dots, l_k)$ . De esta manera, el parámetro sobre el cual se define la complejidad computacional de este método es el número k de umbrales que se buscan, como se puede ejemplificar en la Figura 13 [19].

En el peor de los casos (cuando inicio = 0 y fin = 255), el algoritmo debe generar todas las posibles combinaciones de 256 tonos de gris en grupos de k umbrales. Por lo que su complejidad se puede resumir como el número de combinaciones posibles de 256 tonos de gris en conjuntos de n elementos:

$$T(n) = \frac{256!}{n!(256-n)!} \quad (15)$$

De acuerdo a la Ecuación 15, se puede estimar la complejidad de este algoritmo, obteniendo 256 para n igual a 1, 16320 para n igual a 2, y un valor excesivamente grande de 2763520 para n igual a 3 [18].

### 3.4. Segmentación por crecimiento de regiones

Teniendo como referencia la imagen entera (R), se puede entender la segmentación como una partición de R en n subregiones R1, R2, ..., Rn de manera que:

$$\bigcup_{i=1}^n R_i = R \quad (16)$$

$$R_i \text{ es una región conexa, } i = 1, 2, \dots, n \quad (17)$$

$$R_i \cap R_j = \emptyset, \forall i, j, i \neq j \quad (18)$$

$$P(R_i) = \text{VERDADERO} \quad (19)$$

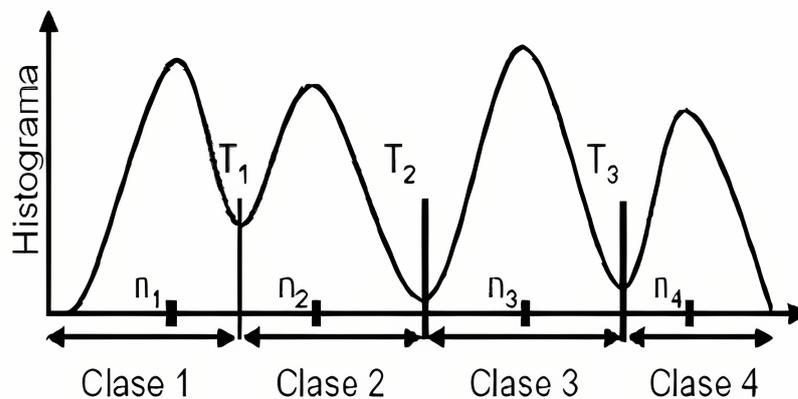


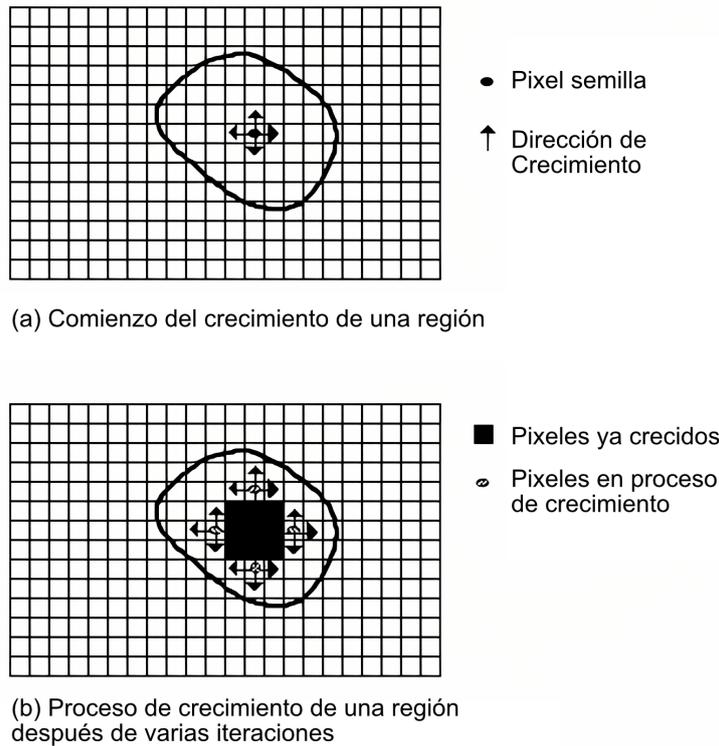
Figura 13. Definición de los umbrales de acuerdo al histograma. Adaptado de [19].

$$P(R_i \cup R_j) = \text{FALSO}, \forall i, j, i \neq j \quad (20)$$

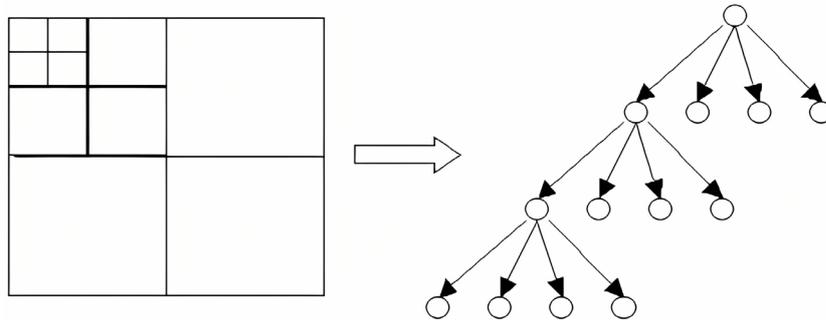
Donde  $P(R_i)$  es un predicado lógico definido sobre los puntos en el conjunto  $R_i$  y  $\theta$  denota el conjunto vacío. La primera afirmación implica que todo pixel debe pertenecer a una región. La segunda conlleva a que toda región en la imagen debe ser conexa, es decir que los pixeles de una misma región están conectados entre sí. La tercera imposibilita que un mismo pixel pertenezca a más de una sola región [20]. La conexión de los pixeles se puede generar con un crecimiento progresivo como se observa en la Figura 14 [19].

### 3.5. Segmentación separar-fusionar (closing-dilation)

Este algoritmo funciona en dos fases principales. Primero se descompone la imagen en pequeñas regiones homogéneas, esto se realiza por medio de un algoritmo de descomposición. Dicho algoritmo inicia considerando la imagen completa como una región, y evaluando el predicado P. Cada vez que una región no cumpla con el predicado, se divide en cuatro cuadrantes del mismo tamaño formando un árbol cuaternario, como se observa en la figura 15, que representa la jerarquía de regiones homogéneas. De esta manera el algoritmo se ejecuta secuencialmente sobre los nuevos nodos del árbol hasta que no queden regiones por evaluar [21].



**Figura 14.** Ejemplificación del proceso de segmentación por crecimiento de regiones.  
Adaptado de [19].



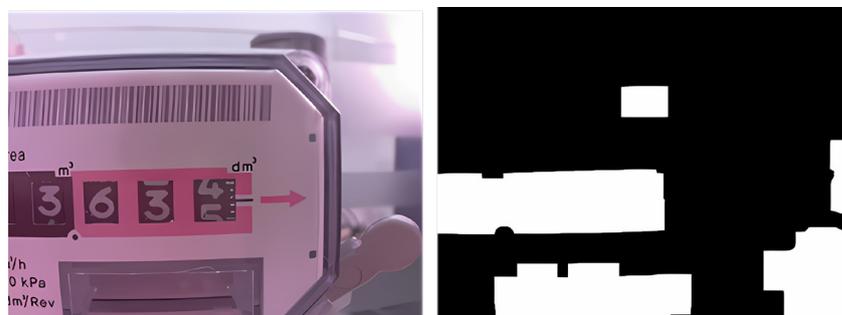
**Figura 15.** Ejemplificación de división de imagen en árbol cuaternario.

Finalmente, el proceso se complementa con una fase dos, que consiste en la fusión de las regiones homogéneas colindantes. Para este proceso se evalúa el criterio de homogeneidad para cada región junto con sus regiones adyacentes, cuando dos regiones cumplen con el criterio, se fusionan y se actualiza la información del árbol cuaternario [22]. En la Figura 16 se observa el resultado final de la definición de las regiones para una aplicación real en un medidor de gas.

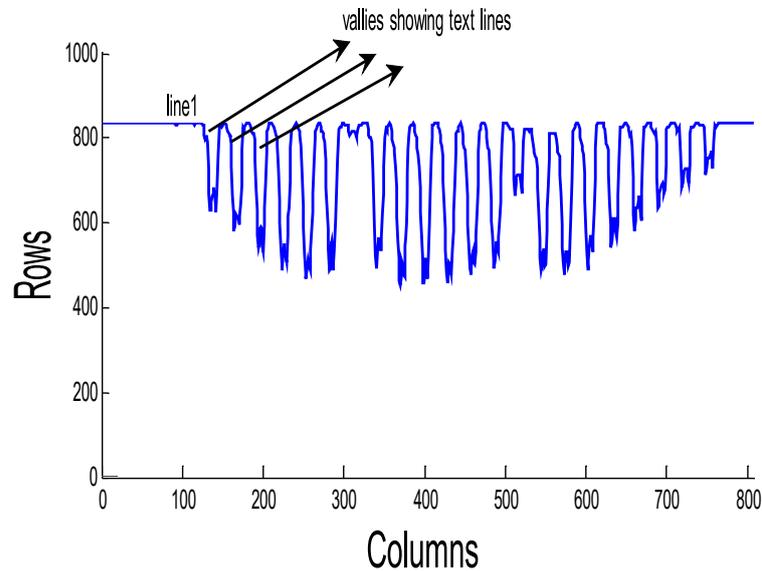
### 3.6. Segmentación de líneas de texto y palabras

Para el caso de OCR aplicado a texto impreso de documentos escaneados, se realiza segmentación de la posición del texto, que comprende separación de líneas, palabras y caracteres. Teniendo en cuenta que este tipo de texto está bastante normalizado (ASCII o UNICODE), para este tipo de aplicación del OCR la ventana de mejora está en realidad en el pretratamiento de la imagen o la cantidad de escenarios posibles que se puedan cubrir, sin importar qué condiciones [23].

Una vez se tiene segmentado el párrafo del documento escaneado, se hace uso del perfil de proyección horizontal (representación gráfica bidimensional de la posición de la cantidad de



**Figura 16.** Ejemplo real de aplicación del algoritmo closing-dilation. (a) Imagen original. (b) resultado del filtro.



**Figura 17.** Identificación de las líneas de texto con base en la proyección horizontal. Recuperado de [7].

pixeles negros), que se realiza con base en el histograma, para realizar la segmentación de las líneas, dicho perfil muestra de manera clara (si el documento tiene la inclinación previamente verificada/corregida) por medio de valles en la gráfica la ubicación de los espacios blancos entre líneas como se observa en la Figura 17 [7].

De manera similar, para realizar la segmentación de palabras en una línea de texto previamente identificada, se emplea la proyección vertical de la misma, y el espaciado entre palabras correspondería a los valles de la gráfica en este caso. Esto presenta el problema con palabras separadas, posibles manchas en el documento, en general depende del buen estado del documento escaneado y del logro de la corrección de cualquier posible inclinación que permita que las proyecciones horizontales y verticales de la imagen sean realmente descriptivas respecto al contenido de la misma [7].

#### 4. Conclusiones

Se encuentra una amplia gama de opciones para tratamiento y pretratamiento de imágenes, que van desde corrección de imperfecciones hasta una segmentación específica de objetos de interés, siendo aplicables por separado y en conjunto, constituyendo así una herramienta formidable para la alimentación de una entrada bastante procesada al algoritmo de OCR.

Dentro de las tecnologías expuestas normalmente, no todas son aplicables de manera simultánea como tratamiento previo a la implementación de un algoritmo OCR (debido a que cumplen funciones similares). Por ejemplo, para la segmentación, las tecnologías de umbralización, algoritmo OTSU y segmentación multumbral son mutuamente excluyentes ya que abordan un mismo problema desde distintos enfoques. Por otra parte, si una imagen requiere tratamiento (de ruido, posición y/o inclinación)

podrían implementarse varios algoritmos de manera consecutiva para lograr el mejor resultado posible.

Como se ha mencionado a lo largo del texto, la aplicación de las diferentes técnicas dependerá exclusivamente del caso específico que se esté tratando, ya que las necesidades de tratamiento para diferentes imágenes pueden diferir de manera drástica, siendo en algunos casos imágenes de buena calidad que no requieran filtros morfológicos o correcciones y en otros, imágenes que requieran uno o varios de estos filtros de pre tratamiento. Se recomienda entonces emplear los resultados obtenidos en el OCR posterior al pretratamiento como indicador de la necesidad de la implementación de correcciones; esto es, en la medida que el algoritmo presente buenos resultados sin un pretratamiento en específico, la pertinencia de este deberá ser evaluada y excluido del conjunto aplicado.

El tratamiento de imagen depende directamente de la calidad de la imagen a procesar, por lo que su implementación está ligada a la evaluación por experticia de usuario, y puede evaluarse la pertinencia de la misma comparando los resultados con la apariencia previa de la imagen. En contraparte, la segmentación depende de qué información se quiera extraer de la imagen, los casos específicos como localización de rojos o segmentación de texto tienen sus objetivos bastante bien definidos y son mejores para sus aplicaciones puntuales que las soluciones generales presentadas. Entre las soluciones generales presentadas, destaca el algoritmo de closing-dilation ya que engloba la tecnología de detección de bordes y de umbralización. Sin embargo, no siempre el algoritmo más complejo es el mejor, ya que en computación y sobre todo en tratamiento de imágenes, muchas veces se prefiere un algoritmo sencillo que reduzca los tiempos y costes de cómputo. En definitiva, en este campo prima la experimentación y comparación directa con el caso de estudio específico que se quiera tratar.

## Bibliografía

- [1] Agrawal, A., Jain, S., Deep, V., Sharma, P., and Mishra, S. (2021) "Meter Reading using OCR for Comfort and Cost Optimization," 2021 5th International Conference on Information Systems and Computer Networks (ISCON), 2021, pp. 1-4, doi: 10.1109/ISCON52037.2021.9702310.
- [2] Devi, H. (2006). Thresholding: A Pixel-Level image processing methodology preprocessing technique for an OCR system for the Brahmi script. *Ancient Asia*, 1.
- [3] Li, X., Li, C., Rahaman, M.M., Sun, H., Li, X., Wu, J., Yao, Y. & Grzegorzec, M. (2022). A comprehensive review of computer-aided whole-slide image analysis: from datasets to feature extraction, segmentation, classification and detection approaches. *Artif Intell Rev* 55, 4809–4878. Doi: 10.1007/s10462-021-10121-0
- [4] Ziyadinov, V., & Tereshonok, M. (2022). Noise immunity and robustness study of image recognition using a convolutional neural network. *Sensors*, 22(3), 1241. Doi: 10.3390/s22031241
- [5] Vanetti, Marco & Gallo, Ignazio & Nodari, Angelo. (2013). GAS Meter Reading from Real World Images using a Multi-net System. *Pattern Recognition Letters*. 34. Doi: 10.1016/j.patrec.2012.11.014.
- [6] Aparicio Martín de Loeches, A., & Fernández Guzmán, L. (2015). Reconocimiento óptico de caracteres en imágenes digitales de contadores de gas.
- [7] Shinde, A. A., & Chougule, D. G. (2012). Text pre-processing and text segmentation for OCR. *International Journal of Computer Science Engineering and Technology*, 2(1), 810-812.

- [8] Bieniecki, W., Grabowski, S., & Rozenberg, W. (2007). Image preprocessing for improving ocr accuracy. In *2007 international conference on perspective technologies and methods in MEMS design* (pp. 75-80). IEEE.
- [9] Zhang, X., Liu, Y., Huo, C., Xu, N., Wang, L., & Pan, C. (2022). PSNet: Perspective-sensitive convolutional network for object detection. *Neurocomputing*, 468, 384-395. Doi: 10.1016/j.neucom.2021.10.068
- [10] Ihara, S., Saito, H., Yoshinaga, M., Avala, L., & Murayama, M. (2022). Deep learning-based noise filtering toward millisecond order imaging by using scanning transmission electron microscopy. *Scientific reports*, 12(1), 1-13. Doi: 10.1038/s41598-022-17360-3
- [11] Lee, Y., Song, J., Won, Y. (2018). Improving personal information detection using OCR feature recognition rate. 1941-1952. Doi: 10.1007/s11227-018-2444-0
- [12] Li, Y., Liu, C., You, X., & Liu, J. (2022). A Single-Image Noise Estimation Algorithm Based on Pixel-Level Low-Rank Low-Texture Patch and Principal Component Analysis. *Sensors*, 22(22), 8899. Doi: 10.3390/s22228899
- [13] Yan, F., Venegas-Andraca, S. E., & Hirota, K. (2022). Toward implementing efficient image processing algorithms on quantum computers. *Soft Computing*, 1-13. Doi: 10.1007/s00500-021-06669-2
- [14] Bhaumik, R., Deb, D., & Majumdar, P. (2022). Medical Image Processing Using Image Segmentation Techniques. *AIJR Abstracts*, 7.
- [15] Abbasi, N., Khan, M. F., Khan, E., Alruzaiqi, A., & Al-Hmouz, R. (2022). Fuzzy histogram equalization of hazy images: a concept using a type-2-guided type-1 fuzzy membership function. *Granular Computing*, 1-15. Doi: 10.1007/s41066-022-00351-0
- [16] Gupta, M. R., Jacobson, N. P., & Garcia, E. K. (2007). OCR binarization and image pre-processing for searching historical documents. *Pattern Recognition*, 40(2), 389-397.
- [17] Arroyave-Giraldo, M., Restrepo-Martínez, A., & Vargas-Bonilla, F. (2011). Incidencia de la Segmentación en la Obtención de Región de Interés en Imágenes de Palma de la Mano. *TecnoLógicas*, (27), 119-138.
- [18] Abdel-Basset, M., Chang, V., & Mohamed, R. (2021). A novel equilibrium optimization algorithm for multi-thresholding image segmentation problems. *Neural Computing and Applications*, 33(17), 10685-10718.
- [19] Cuadri Vázquez, J., Fondón García, I. & Universidad de Sevilla. (2009). Segmentación de imágenes en color por intersección de histogramas de color y textura: proyecto fin de carrera. s.n.
- [20] Preetha, M. M. S. J., Suresh, L. P., & Bosco, M. J. (2012). Image segmentation using seeded region growing. In *2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET)* (pp. 576-583). IEEE.
- [21] Tian, Y., Wang, K., Li, R., & Zhao, L. (2018). A fast incremental map segmentation algorithm based on spectral clustering and quadtree. *Advances in Mechanical Engineering*, 10(2), 1687814018761296.
- [22] Yu, J., Xiao, C., Hu, T., & Gao, Y. (2022). Selective weighted multi-scale morphological filter for fault feature extraction of rolling bearings. *ISA transactions*. Doi: 10.1016/j.isatra.2022.06.003
- [23] He, S., & Schomaker, L. (2020). Fragnet: Writer identification using deep fragment networks. *IEEE Transactions on Information Forensics and Security*, 15, 3013-3022. Doi: 10.1109/TIFS.2020.2981236