

Capítulo 2. Técnicas de procesamiento digital de imágenes y reconocimiento de patrones.

2.1 Revisión sistema reconocimiento caracteres

Un sistema de reconocimiento típicamente esta conformado por los siguientes partes [8] [9]:

Adquisición: Se refiere al proceso de convertir un documento a una representación apta para ser procesada por la computadora. Tal adquisición y conversión puede ser realizada por un escáner, cámara fotográfica o de video, etc. y el resultado puede variar dependiendo del proceso de digitalización usado y el método de codificación.

Binarización: Convierte la imagen recibida en una imagen binaria separando así el fondo de los objetos a analizar.

Análisis de documento: Trata de analizar la estructura de un documento y entender la información contenida en sus componentes.

Segmentación: Se refiera a delimitar las regiones que nos interesan. En el caso del reconocimiento de caracteres se refiere a las líneas y caracteres individuales.

Normalización: Trata de ajustar la forma, tamaño y posición del caracter esto con el objetivo de minimizar la variación entre imágenes de la misma clase.

Extracción de características: Permite conocer las características como lo son el tamaño, perímetro, área, etc. Así como características topográficas como lo es la orientación de segmentos.

Reconocimiento: Convierte la imagen binaria en una representación electrónica. Misma que permite realizar operaciones de validación y análisis.

Procesado: Se refiere a mejorar el proceso de reconocimiento mediante el uso de información contextual como lo es análisis a nivel lingüístico.

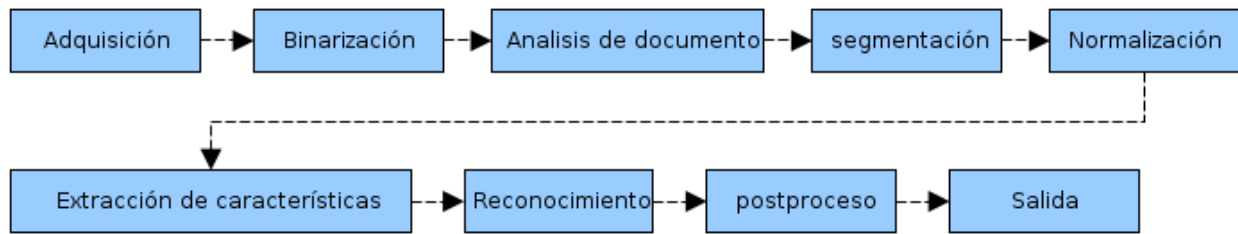


Figura 2.1: Sistema óptico de reconocimiento de caracteres

2.2 Binarización

Los métodos de binarización se pueden dividir en dos tipos: Globales y locales [8].

Los métodos globales tratan de encontrar un umbral el cual aplicar a toda la imagen entre estos se encuentran los métodos por umbral y Otsu. Los métodos locales obtiene el umbral para cada pixel en la imagen usando para ello los valores de sus vecinos, entre los métodos de esta categoría entran el método de Niblack y Saovola.

Los métodos locales generalmente producen un mejor resultado al binarizar la imagen incluso en situaciones donde la iluminación en el documento es variable. Sin embargo las limitaciones de procesamiento y memoria lo hacen difícil de implementar en dispositivos móviles. Si bien se pueden utilizar alternativas como los son las imágenes integrales [10] para obtener los resultados de la binarización local y una rápida ejecución esto requiere memoria, además de que los grandes valores que se puede necesitar la imagen pueden provocar un desbordamiento de búfer.

Por las razones mencionadas arriba se optó por un método global que está que se define como [11]:

$$g(x, y) = \begin{cases} 1 & \text{si } f(x, y) > T \\ 0 & \text{si } f(x, y) \leq T \end{cases}$$

Para definir el tamaño del umbral se recurre al algoritmo iterativo:

1. Se obtiene un estimado inicial del umbral T
2. Se calcula la media de los valores debajo (m_1) y sobre (m_2) del umbral T .
3. Se calcula el nuevo umbral: $T = \frac{1}{2}(m_1 + m_2)$
4. Se repite el paso dos hasta que el nuevo umbral sea igual al umbral anterior.

2.3 Segmentación

En el caso de la segmentación se evaluaron dos métodos Descomposición de árbol X-Y y Etiquetado de componentes conexos. Una breve explicación de su funcionamiento, sus características y por qué se eligió uno se muestra a continuación.

2.3.1 Descomposición de árbol X-Y

La descomposición de árbol X-Y [8] es un algoritmo de bajo nivel que realiza la proyección horizontal y vertical de una página para separar los caracteres de un documento. Primero se obtiene la proyección horizontal de la página para obtener los renglones del documento (Figura 2.2.). Una vez obtenidos los renglones se realiza la proyección vertical, se

analiza la proyección para obtener los caracteres del renglón. Estas proyecciones pueden variar dependiendo de la complejidad del documento.

La descomposición es un método comúnmente usado, sin embargo tiene las siguientes desventajas:

1. Susceptible a fallar en la presencia si existen líneas que cubran o rodeen a los caracteres.
2. Puede fallar en la presencia de ruido en la imagen.
3. Requiere que la imagen se encuentre alineada, por lo cual precisa de aplicar corrección de inclinación sobre la imagen.

La efectividad de este algoritmo puede no ser la adecuada para ser aplicado a dispositivos móviles debido a que la posibilidad de que las imágenes obtenidas con la cámara de un teléfono celular contengan ruido es alta, la inclinación de la fotografía tomada raramente estará correctamente alineada y que estimar tamaño del carácter es difícil dado que el histograma no sea lo suficientemente claro debido a las variaciones de luz.

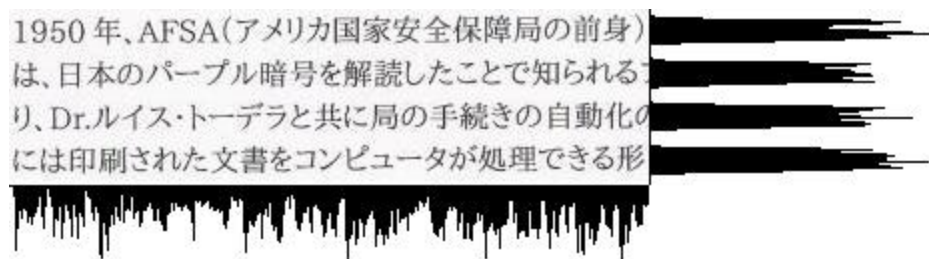


Figura 2.2: Proyecciones vertical y horizontal

2.3.2 Etiquetado de componentes conexas

Es una técnica que asigna una etiqueta a cada componente conexo para obtener al final una región [12]. Básicamente el algoritmo recorre una imagen binaria de izquierda a derecha y de arriba hacia abajo en busca de píxeles con valor “1” que se encuentren conectados entre si en un rango que puede ser N8, N4 o ND y les asigna una etiqueta construyendo de esta manera regiones.

Esta técnica, a diferencia de la descomposición de árbol X-Y, no es tan susceptible a tener errores si los caracteres están rodeados o de líneas que cubran o rodeen los caracteres. Además no requiere que la imagen se encuentre alineada con respecto al eje para ser efectiva lo cual es de gran utilidad debido a que no siempre se pueden tener correctamente alienado el dispositivo móvil con la imagen como seria con un escáner de sobremesa o plano.

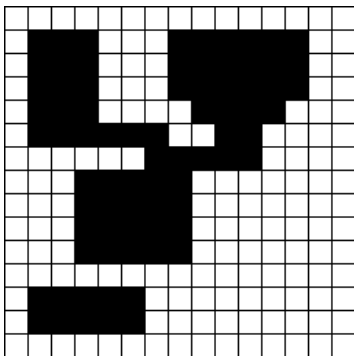


Figura 1.3: Imagen Binaria

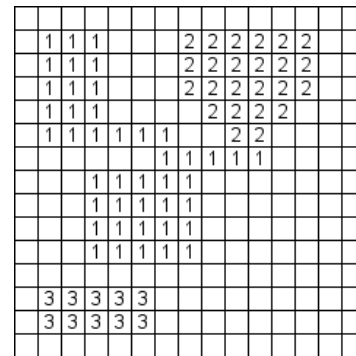


Figura 1.4: Etiquetado terminado

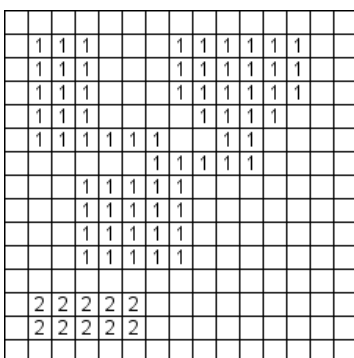


Figura 1.5: Etiquetado después de unificar

2.4 Normalización

Una vez binarizada la imagen se realiza la normalización del caracter. Para esto se centra la imagen en un plano cuyas dimensiones son iguales a $m \times m$ donde m es igual al tamaño del lado más largo del caracter a normalizar. Después esta imagen se escala a un tamaño de 128×128 pixeles y se le aplicara un filtro para eliminar el ruido que pueda quedar en la imagen para terminar con un suavizado.

2.5 Extracción de características

Para la fase de extracción de características se propone utilizar *Local Strokedirection features* (LSD) [8]. En esta técnica se procesa por cada pixel "1" los pixeles repetidos en cada una de las cuatro direcciones y después es normalizado. Después se divide la imagen en secciones de tamaño $n \times n$ y se procesan las direcciones de cada área como el promedio de los pixeles en el área que en este caso serian ocho.

2.6 Clasificación

2.6.1 K-vecino más cercano

K-vecino más cercano es un algoritmo de clasificación supervisada, esto es que se parte de un conjunto de muestras que son etiquetadas dentro de una clase a la que pertenecen; siendo estas clases asociadas después con un conjunto de muestras de entrenamiento. En este algoritmo clasifica los objetos de acuerdo a sus atributos y a las muestras de entrenamiento [13].

Dado un conjunto de entrenamiento $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, que contiene clases conocidas, para decidir si una muestra desconocida x pertenece a una clase se decide mediante a la probabilidad condicional $p(y|x)$.

En su versión más básica ($K = 1$) se busca determinar el vecino más cercano al cual afecta x :

$$1_NN = \operatorname{argmin} \|x - x_i\|$$

Para determinar la cercanía a de alguna instancia se utiliza por lo general distancias euclidianas, aunque se pueden utilizar otras.

En el caso de k -vecinos más cercano dado un dato x_q a clasificar y sean $x_1 \dots x_k$ los vecinos más cercanos a x_q usando, en el caso de clasificaciones discretas:

$$f(x_q) = \operatorname{argmax}_v \in V \sum_{i=1}^k \partial(V, f(x_i))$$

Y en el caso de continuas:

$$f(x_q) = \frac{\sum_{i=1}^k f(x_i)}{k}$$