

MODELO DE SEGMENTACIÓN DE IMÁGENES PARA SISTEMA DE EVALUACIÓN DE COLOR DEL PILONCILLO

Recibido: agosto 25, 2017
Aceptado: septiembre 23, 2017

H.R.Lárraga Altamirano¹
O. Espinosa Guerra²
A.M. Piedad Rubio³
P. Hernández Hernández⁴

RESUMEN

La segmentación es un proceso que subdivide la imagen en sus componentes u objetos dentro de ella, es decir, permite separar las partes de interés de las que no tienen significancia para los fines de la aplicación. El presente proyecto evalúa dos metodologías de segmentación con la finalidad de seleccionar la más apropiada, la cual es requerida por un sistema de visión artificial para evaluar el color del piloncillo; producto endulcorante derivado del procesamiento de la caña de azúcar. El objetivo es extraer únicamente la subimagen del producto (piloncillo) del resto de los datos que no son útiles para el procesamiento mediante algoritmos para detección de bordes y de umbralización. El diseño del experimento abarca tres fases, la configuración de un escenario controlado de iluminación, el desarrollo de los módulos de segmentación en el lenguaje de programación Matlab® R2015a y, por último, la fase de pruebas, con una muestra de 14 piloncillos.

PALABRAS CLAVE

Segmentación, detección de bordes, umbralización, Sobel, Otsu

ABSTRACT

Segmentation is a process that subdivides the image into its components or objects within it, ie it allows to separate the parts of interest from those that have no significance for the purposes of the application. The present project evaluates two methodologies of segmentation with the main goal of selecting the most appropriate, which is required by an artificial vision system for evaluating the color of the piloncillo; sweetening product derived from the processing of sugar cane. The objective is to extract only the subimage of the product (piloncillo) from the rest of the data that are not useful for the processing by algorithms for edge detection and thresholding. The design of the experiment involves three phases, the configuration of a controlled lighting scenario, the development of the segmentation modules in the Matlab® R2015a programming language and finally, the test phase, with a sample of 14 piloncillos.

KEY WORDS:

Segmentation, edge detection, thresholding, Sobel, Otsu

¹ Profesor de tiempo completo. Instituto Tecnológico de Ciudad Valles, hugo.larraga@tecvalles.mx

² Jefe de Ingeniería Industrial. Instituto Tecnológico de Ciudad Valles, omar.espinosa@tecvalles.mx

³ Profesora de tiempo completo. Instituto Tecnológico de Ciudad Valles, ana.piedad@tecvalles.mx

⁴ Estudiante. Instituto Tecnológico de Ciudad Valles, placidohdzhdz123@gmail.com

INTRODUCCIÓN

La segmentación es un proceso que subdivide la imagen en sus componentes u objetos dentro de ella, es decir, permite separar las partes de interés de las que no tienen significancia para los fines de la aplicación. Son varias las metodologías existentes para realizar la segmentación de una imagen, generalmente se basan en dos propiedades: la discontinuidad y la similitud. La discontinuidad consiste en detectar los cambios bruscos del nivel de gris, logrando con esto la detección de puntos aislados, líneas y bordes. Por otra parte, la similitud es la regularidad en los valores del gris, algunos algoritmos que trabajan sobre esta característica tratan la umbralización, el crecimiento de región o fusión de regiones (Palomino, N. L. S., & Concha, U. N. R., 2009).

De acuerdo al tipo de problema a resolver, es el método de segmentación el cual debe implementarse en consideración de la iluminación, el tipo de imagen, ruido lumínico entre otros factores. Por esta razón es difícil sugerir un método sobre otro, ya que cada aplicación específica tiene sus propios requerimientos, no existen métodos generales que pueden ser aplicados a cualquier variedad de imágenes. Por lo tanto, la selección del método de segmentación apropiado merece un estudio con detenimiento (Báez Rojas, J. J., Guerrero, M. L., Conde Acevedo, J., Padilla Vivanco, A., & Urcid Serrano, G., 2004).

Con base en lo anterior, el presente proyecto evalúa dos metodologías de segmentación con la finalidad de seleccionar la más apropiada, la cual es requerida por un sistema de visión artificial para evaluar el color del piloncillo; producto endulcorante derivado del procesamiento de la caña de azúcar. Para este sistema en particular, el proceso de segmentación tiene por objetivo extraer únicamente la subimagen del producto del resto de los datos que no son útiles para el procesamiento. Los métodos propuestos utilizan en su desarrollo algoritmos para detección de bordes y de umbralización, así como también la realización de operaciones morfológicas de dilatación y erosión para mejoramiento de la imagen binaria que se servirá como máscara.

En la segmentación, un borde se identifica como una discontinuidad significativa en el nivel de gris entre dos regiones adyacentes. Estas discontinuidades pueden detectarse mediante derivadas de primer y segundo orden, las cuales utilizan el operador gradiente y el laplaciano respectivamente (Figura 1). Con esto, se pretende encontrar los píxeles cuya intensidad sea mayor en magnitud que un umbral especificado, para lo cual se utiliza el operador Sobel. (Palomino, N. L. S., & Concha, U. N. R., 2009). A su vez, la umbralización (thresholding) es frecuentemente usada cuando el objeto de interés es de una intensidad contrastante con el resto de los componentes de la imagen, el resultado es una partición binaria de las intensidades de la misma. En la umbralización se trata de determinar un umbral (threshold), valor que agrupa los píxeles por intensidad (Figura 2). La segmentación entonces se logra al separar todos los píxeles con mayor intensidad al umbral en una clase (intensidad 1), y todos los otros píxeles en otra (intensidad 0) (Coto, 2003).

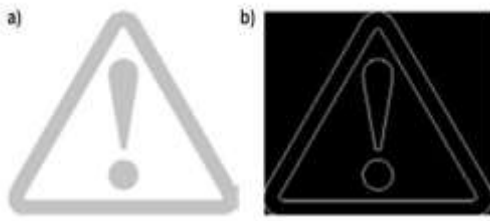


Figura 1. Resultado de la detección de bordes. Fuente (Palomino, N. L. S., & Concha, U. N. R., 2009)

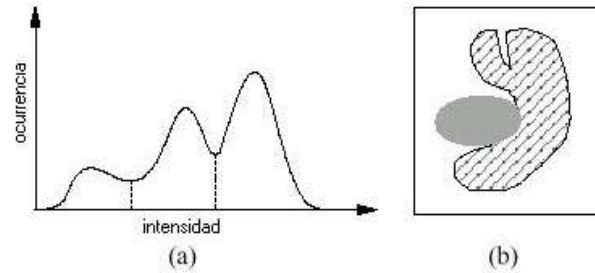


Figura 2. a) Histograma de intensidades de grises con posibles umbrales, b) imagen en escala de grises. Fuente (Coto, 2003)

METODOLOGÍA

El diseño del experimento abarcó tres fases, la primera, que no forma parte del procesamiento de la imagen pero que incide en la efectividad de la tarea de segmentar, es la **iluminación**. En efecto, este fenómeno juega un papel importante en el diseño de los sistemas de visión artificial, afecta las características de color de la imagen por lo que atenta contra la confiabilidad de los datos de entrada (Jaramillo Ortiz, Jiménez, & Ramos, 2014).

De acuerdo a la norma ISO 3664:2009 son cinco las condiciones a tomar en cuenta para la elección de un sistema de iluminación:

- Calidad del color. Para representar la luz del día teóricamente perfecta se requieren de 5.500 Kelvin.
- Intensidad de la luz. Se recomienda una iluminación de 2000 luxes con un índice de reproducción de color mayor de 80.
- Uniformidad. La intensidad de la iluminación en todos los puntos de la superficie de revisión debería ser de al menos 1200 luxes, 60% de la intensidad de luz recomendada (2000lx).
- Entorno. Debe ser neutro y mate, con reflectancia luminosa entre 10% y 60% (gris Munsell N8).
- Geometría. La fuente de luz y el observador deben posicionarse de modo que se minimice la reflexión especular (brillo) (Organización Internacional para la Estandarización (ISO), 2009).

La siguiente etapa es la implementación de los métodos de **segmentación** en sí. Al método de frontera se le denominará método “A” y a la umbralización método “B”, los flujos de proceso están esquematizados en la figura 3 y 4 respectivamente.



Figura 3. Método A Frontera



Figura 4. Método B Umbralización

Fue necesario transformar el espacio de color RGB (Red-Green-Blue) de la imagen original a escala de grises sobre la cual trabajan los algoritmos Sobel y Otsu, utilizados para la segmentación por discontinuidad (detección de bordes) y similitud (umbralización). La aplicación de ambas operaciones resultó en una imagen binaria; para el caso del método “A” se aplicó una operación morfológica de cierre con la intención de definir con mayor precisión los bordes, es decir, se dilatan (agregan) los píxeles de los bordes con la posibilidad de unirlos. Generalmente persisten algunos píxeles no deseados, o bien, hay huecos dentro del objeto, para eliminar estos detalles se requirió una operación de relleno y por último, se erosionaron los píxeles que no pertenecían al área de interés. Con la versión binaria de la imagen original, los píxeles de los objetos contenidos en ella tienen un valor de 1, de tal manera que es posible reconocer las coordenadas que delimitan a cada objeto, así como la cantidad de píxeles de cada región, útil para discriminar aquellos falsos positivos. La imagen segmentada se formó por el producto de la original y la máscara (imagen binaria), ambas correspondientes en dimensiones. (Acero, Á. R., Cano, A. M., & Builes, J. A. J., 2015).

La siguiente fase consistió en las **pruebas**, el lenguaje de programación seleccionado para el desarrollo de los módulos de segmentación fue Matlab R2015a, una suite para desarrollo de software con librerías orientadas al procesamiento de imágenes (MathWorks, 2005). El equipo de cómputo (hardware) utilizado en esta fase integró una cámara web, memoria RAM de 8 Gb y microprocesador Intel CoreI5. La muestra estuvo constituida de 5 piloncillos de prueba de laboratorio y 9 piloncillos comerciales.

RESULTADOS

El escenario de iluminación controlado se construyó bajo las recomendaciones de la norma ISO 3664:2009, aproximando el color de la luz, la distribución de la misma y el fondo para controlar la reflexión especular a las especificaciones D50. La cabina de luz consiste en una caja de madera de dimensiones 1.20 x 70 x 65, el espacio permite ajustar la distancia entre la fuente lumínica y el objeto para determinar el nivel de iluminación (luxes) además, integra una lámpara led de 4500K con una potencia de 980 lúmenes, tal como se muestra en la figura 5.



Figura 5. Cabina de iluminación, fuente original.

El software para segmentar la imagen está desarrollado en Matlab R2015a, seleccionado por su afinidad con el procesamiento de imágenes. Ambos métodos comparten similitud en algunas operaciones dentro de su proceso como se pudo apreciar en las figuras 3 y 4, es decir, a nivel de programación la diferencia entre estos radica en la aplicación de los algoritmos Sobel y Otsu, para segmentación por detección de bordes y umbralización respectivamente. A pesar de que existen otros métodos, ya sea para segmentar por discontinuidad o similitud para tomar una decisión de seleccionar uno sobre otro es conveniente considerar factores como la fácil implementación, consumo de recursos de procesamiento y la efectividad en la detección. A continuación, se explica cómo se implementaron los métodos A y B.

Para transformar el espacio de color RGB a escala de grises, se hace uso de la función *rgb2gray(img)*, la cual recibe como argumento la imagen original. Esta operación es necesaria, ya que ambos métodos actúan sobre los niveles de gris de una imagen. La definición de un umbral (*threshold*) es también un dato requerido, este valor para el método A, es un vector de dos elementos con los cuales se comparan los píxeles para determinar si son parte de un borde significativo o débil. Por su parte, el *threshold* del método B permite agrupar los píxeles como parte del fondo, o en su defecto del objeto de interés. La función *edge(imgGrises,'sobel')* opera la detección de bordes y *im2bw(imgGrises,level)* la umbralización.















Las operaciones morfológicas ayudan a mejorar la imagen binaria a partir de los defectos de distorsión, esto es, píxeles no deseados o formas incompletas. Sobel a diferencia de Otsu requiere de una operación adicional de cierre lograda por *imclose(im_edge, SE)*, actúa sobre los bordes detectados adicionando píxeles y después removiendo los excesos a través de un elemento estructural. Para ambos resultados, son rellenadas las regiones de posibles huecos (*imfill(imgSobelOtsu,'holes')*) y nuevamente se realiza un borrado de los píxeles “basura” persistentes.

Hasta este punto se han obtenido dos imágenes binarias que funcionarán como la máscara de la segmentación, en otras palabras, solo los píxeles del área de interés están en 1 (color blanco) y el resto en 0 (color negro). Sin embargo, aún existe la posibilidad de tener falsos positivos, es decir, remanentes de píxeles en color blanco confundidos con bordes para el caso de Sobel, o pasados por alto por el threshold de Otsu. Por lo tanto, la siguiente operación es encontrar las regiones de los objetos en la imagen a través de *bwboundaries(imgBinaria,'noholes')*, calcular el área de cada uno y considerar aquellos que tengan suficiente densidad de píxeles.

Como resultado de este proceso de depuración, se obtiene la identificación de las regiones (con sus coordenadas respectivas) de los piloncillos dentro de la imagen, claro está, que es la versión binaria del objeto conocida como máscara. La imagen segmentada se forma por el producto de las subimágenes extraídas de la imagen original y la máscara por medio de la función *imcrop(img,vector_coordenadas)*. Este producto involucra tres operaciones similares, por cada canal de la imagen R, G y B; para ejemplificar su implementación se muestra el cálculo del canal de rojos *imgSegmentada(:, :, 1) = imgOriginal(:, :, 1). *uint8(imgMascara);*

Las pruebas se realizaron con piloncillos fabricados en laboratorio (bajo condiciones controladas de producción) y con productos que circulan en el mercado, siendo los resultados los que se muestran en la tabla 1 y 2.

Tabla 1 Resultados segmentación Sobel

Sujeto	Imagen original	Sobel	Relleno	Erosión	Mascara	Región	Imagen segmentada
1							
2							


















































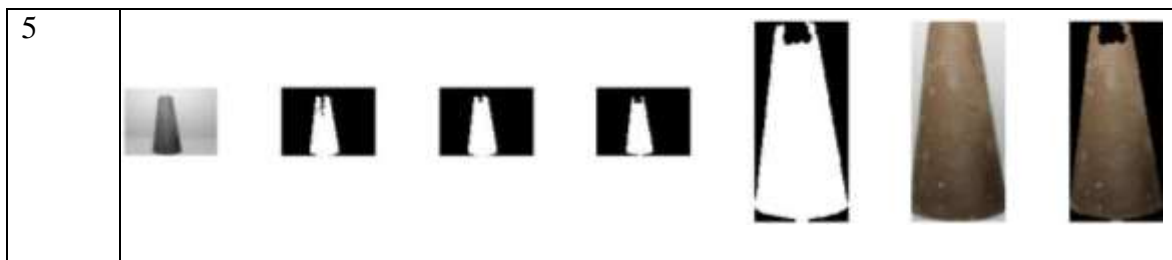
3							
4							
5							

Tabla 2 Resultados segmentación Otsu

Sujeto	Imagen original	Sobel	Relleno	Erosión	Mascara	Región	Imagen segmentada
1							
2							
3							
4							



DISCUSIÓN

Ambos métodos demuestran efectividad, sin embargo, el contorno de la imagen segmentada por umbralización es más liso, por lo tanto, más semejante al real, aunque esto no podría ser considerado como un factor decisivo en la elección del método de segmentación; se observan comportamientos en los dos casos que afectan el proceso, de mayor severidad que tan solo unos cuantos pixeles “basura” adheridos a la orilla del piloncillo.

El método “A”, detección de bordes resulta muy sensible al ruido lumínico, por ejemplo, identifica bordes falsos en sombras, fisuras diminutas o en cambios de color provocados por la iluminación, tal como lo presenta lo sujetos 3 y 5 de la tabla 1. Con la elección de un iluminante adecuado puede considerarse efectivo en la detección del borde del objeto.

El método “B”, umbralización no actúa adecuadamente cuando el color del objeto no es tan contrastante con el del fondo, en el histograma del sujeto 1 tabla 2, cuyo color es oscuro, los valles se diferencian notablemente facilitando el cálculo del umbral para separar los pixeles del fondo de los que pertenecen al área de interés, mientras que en los de color claro como el sujeto 5 tabla 2, el resultado no es el deseado (ver figura 6).

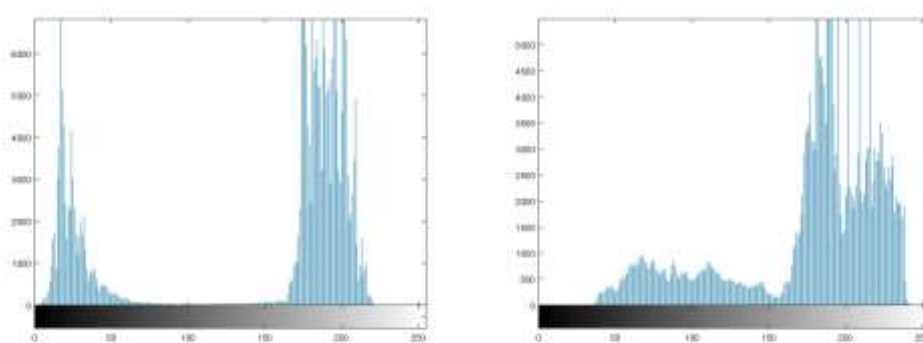




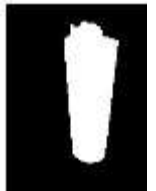




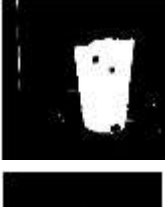







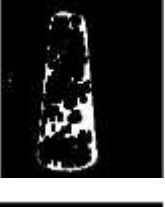
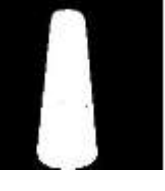

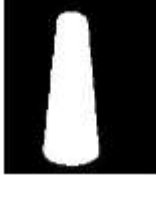



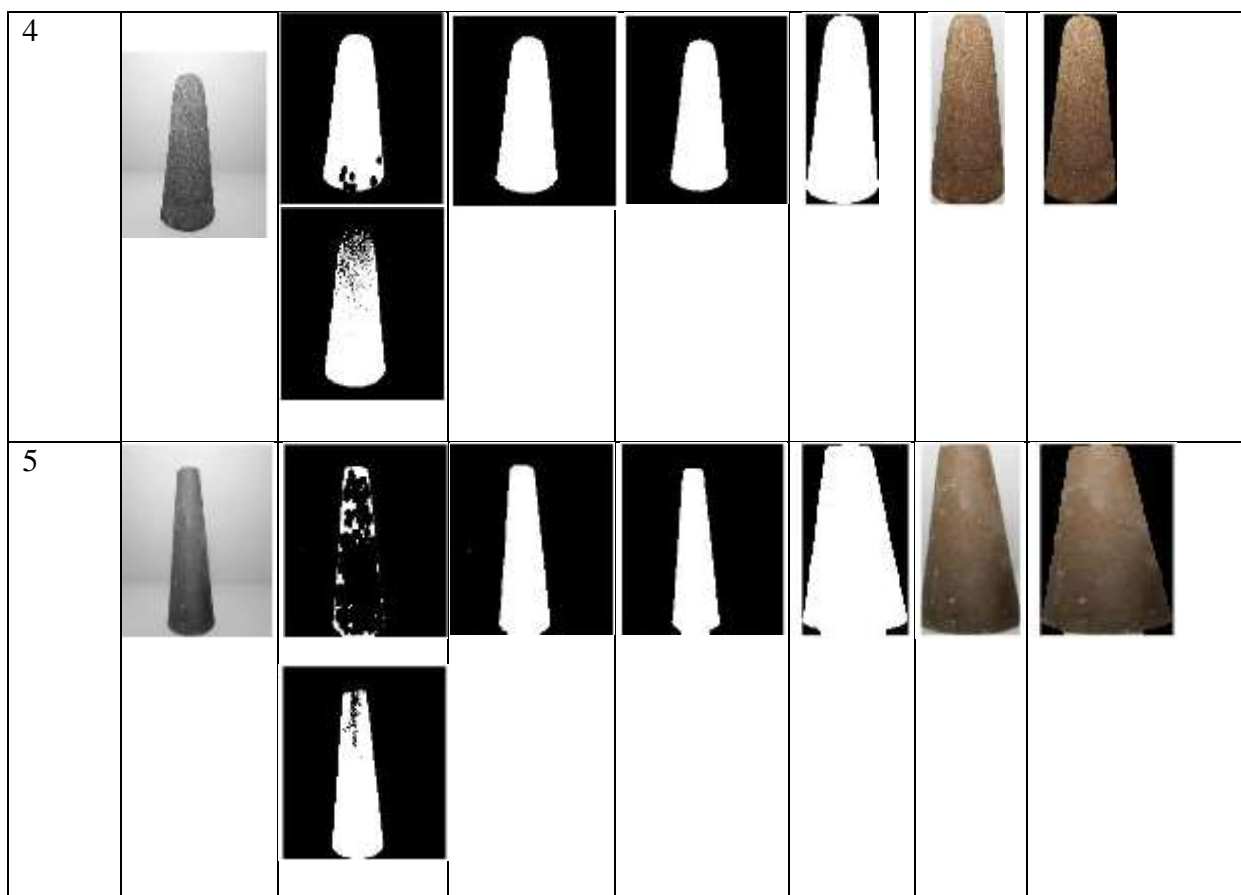


Figura 6. Histogramas piloncillo oscuro (derecha), claro (izquierda), fuente original.

Basado en estos comportamientos y en el análisis de los resultados de los métodos, se plantea la fusión de Sobel y Otsu para abarcar una mayor gama de casos en las que puede existir variabilidad en el iluminante y el color del objeto; el resultado de esta adición se observa en la tabla 3.

Tabla 3 Resultados segmentación Sobel + Otsu

Sujeto	Imagen original	Sobel + Otsu	Relleno	Erosión	Mascara	Región	Imagen segmentada
1		 					
2		 					
3		 					



CONCLUSIONES

La industria piloncillera de la región, la cual representa una economía en crecimiento potencial, recurre aún a prácticas artesanales para la manufactura del piloncillo, carecen de métodos estandarizados de producción y por lo tanto de un nulo control de calidad. Es en este punto, donde los sistemas de visión artificial han encontrado cabida en la industria alimenticia, mediante la inspección de elementos tales como la forma, tamaño, textura, color, brillo, uniformidad, entre otros; considerados como distintivos de calidad en los mercados nacionales y globales.

La segmentación es una de las etapas de mayor importancia en el desarrollo de sistemas inteligentes que involucran procesamiento de imágenes. No hay un método único para el vasto número de aplicaciones que pueden construirse, por lo tanto, es necesario diseños personalizados que aporten solución al problema en específico. Para el caso del piloncillo, la adición de los algoritmos de segmentación por detección de bordes y umbralización resultó efectiva, pues permitió procesar la variedad de colores que el producto presenta. Los piloncillos claros son efectivamente segmentados con Sobel, mientras que para los de color de mayor contraste con el fondo, Otsu actúa eficientemente.

Estas dos técnicas no son las únicas que pueden ser utilizadas para segmentar el piloncillo dentro de una imagen RGB, de hecho, se recomienda que se exploren otros modelos de

segmentación que actúen con la misma o mayor efectividad en entornos de iluminación no controlada. También, en trabajos futuros se recomienda considerar la velocidad de operación de estos métodos, factor relacionado con el consumo de procesador derivado de la ejecución de complejas operaciones.

Con estos proyectos se pretende contribuir a la innovación de la industria piloncillera del país con hincapié en el desarrollo económico de la región de la Huasteca Potosina, a través de soluciones tecnológicas que mejoren los procesos agroindustriales, particularmente en el control de calidad del piloncillo.

BIBLIOGRAFÍA

- Acero, Á. R., Cano, A. M., & Builes, J. A. J. (2015). Sistema de clasificación por visión artificial de mangos tipo Tommy. *Revista UIS Ingenierías*, 14(1).
- Báez Rojas, J. J., Guerrero, M. L., Conde Acevedo, J., Padilla Vivanco, A., & Urcid Serrano, G. (2004). Segmentación de imágenes de color. *Revista mexicana de física*, 50(6), 579-587.
- Coto, E. (2003). Métodos de segmentación de imágenes médicas. Universidad Central de Venezuela: *Lecturas en Ciencias de la Computación*, 1, 9-15.
- Jaramillo Ortiz, A., Jiménez, R., & Ramos, O. L. (2014). Inspección de calidad para un sistema de producción industrial basado en el procesamiento de imágenes. *Tecnura*, 18(41), 76-90.
- Organización Internacional para la Estandarización (ISO). (2009). ISO 3664:2009(E) Graphic technology and photography - Viewing conditions. Switzerland: ISO.
- Palomino, N. L. S., & Concha, U. N. R. (2009). Técnicas de segmentación en procesamiento digital de imágenes. *Revista de investigación de Sistemas e Informática*, 6(2), 9-16.
- Palomino, N. L. S., & Concha, U. N. R. (2009). Técnicas de segmentación en procesamiento digital de imágenes. *Revista de investigación de Sistemas e Informática*, 6(2), 9-16.