
Una aplicación del Algoritmo de Detección de Bordes Canny en Procesos Alimentarios

An application of the Canny Edge Detection Algorithm in Food Processes

Sheyla Nivia Barreda-del-Arroyo¹, Zulema Lilian Mamani-Huacani², Jorge Javier Mendoza-Montoya³, Omar Castillo-Alarcón⁴

Universidad Nacional del Altiplano, Universidad Andina Nestor Cáceres Velásquez,
Juliaca, Perú

Resumen

Una potencialidad del Perú es la industria alimentaria, esto por la calidad y diversidad de sus insumos, se propone la detección de bordes mediante Canny para la estimación del tamaño de productos en una faja transportadora, esto para garantizar un tamaño mínimo del producto. Una aplicación evalúa continuamente imágenes de una cámara y al estímulo de un sensor de varilla o similar hace el cálculo de área. Se concluye que Canny, con una debida configuración puede realizar el trabajo adecuadamente.

Abstract

One potential of Peru is the food industry, this due to the quality and diversity of its products, it is proposed to detect edges using Canny to estimate the size of products in a conveyor belt, this to guarantee a minimum size of the product. An application continually evaluates images from a camera and, upon stimulation from a rod sensor or similar, makes the area calculation. It is concluded that Canny, with proper configuration, can do the job properly.

Palabras Clave: Canny, Visión Artificial, Detección de Bordes, Industria Alimentaria.

Keywords: Canny, Machine Vision, Edge Detection, Food Industry.

I. INTRODUCCIÓN

Las naranjas son un producto emblemático de la selva de la región Puno, se produce en abundancia debido a su clima, una faja transportadora desplaza las naranjas en un proceso, se hace uso del algoritmo de detección de bordes Canny.

Usando una cámara, se implementó un mecanismo de detección de bordes usando el algoritmo Canny para detectar el contorno de una naranja y mediante este se calculó el área de una naranja en píxeles, a continuación se calculó un área patrón de cm^2 por píxeles, de esta manera el área encerrada en el contorno se podía calcular mediante relación entre píxeles y cm^2 , finalmente se guardó el resultado en una base de datos local.

¹ Sheyla Nivia Barreda-del-Arroyo, Universidad Nacional del Altiplano, UNAP, sheybarreda@gmail.com.

² Zulema Lilian Mamani Huacani, Universidad Nacional del Altiplano, UNAP, zulney@gmail.com.

³ Jorge Javier Mendoza-Montoya, Universidad Andina Néstor Cáceres Velásquez, UANCV, javier.mmo@gmail.com.

⁴ Omar Castillo-Alarcón, Universidad Nacional Mayor de San Marcos, UNMSM, gnu.omar@gmail.com.

La aplicación desarrollada permite analizar naranjas de distintos tamaños y verificar si cumplen con tamaños mínimos o algún tipo de requerimiento similar.

El presente trabajo está organizado de la siguiente manera, la sección 2 presenta una descripción general del algoritmo Canny y el método usado para calcular el tamaño de una naranja usando solo la detección de borde; luego, el conjunto de datos y la metodología se describe en la sección 3 así como el proceso experimental, el diseño y los resultados se presentan en la sección 4, y finalmente conclusiones y recomendaciones de trabajo futuro se discuten en la sección 5.

II. MATERIALES Y MÉTODOS

El Algoritmo de Canny fue descrito por John F. Canny en 1986 (Canny, 1986), usa un algoritmo de múltiples etapas para detectar una amplia gama de bordes en imágenes, aunque se puede aplicar a imágenes a color, estas deben convertirse a escala de grises para ser tratadas mediante Canny.



Figura 1, Un ejemplo del proceso de detección de bordes mediante el algoritmo Canny, la imagen original es convertida a escala de grises, la misma que es filtrada con un filtro de desenfoque blur gaussiano y finalmente el algoritmo Canny es aplicado. Fuente, Elaboración propia.

Las etapas del Algoritmo Canny son (*Figura 1*):

- Convertir la Imagen Original a una escala de grises, creando la imagen *gris*.
- Un filtro Gaussiano *blur* es aplicado a la imagen *gris* para eliminar el ruido, este proceso hace que la imagen pierda definición, pero es necesario para mejorar la detección de bordes.
- Se aplica el Algoritmo Canny, estableciendo en el proceso valores de histéresis (umbral) tanto superior como inferior, el resultado es una imagen que muestra los bordes detectados.

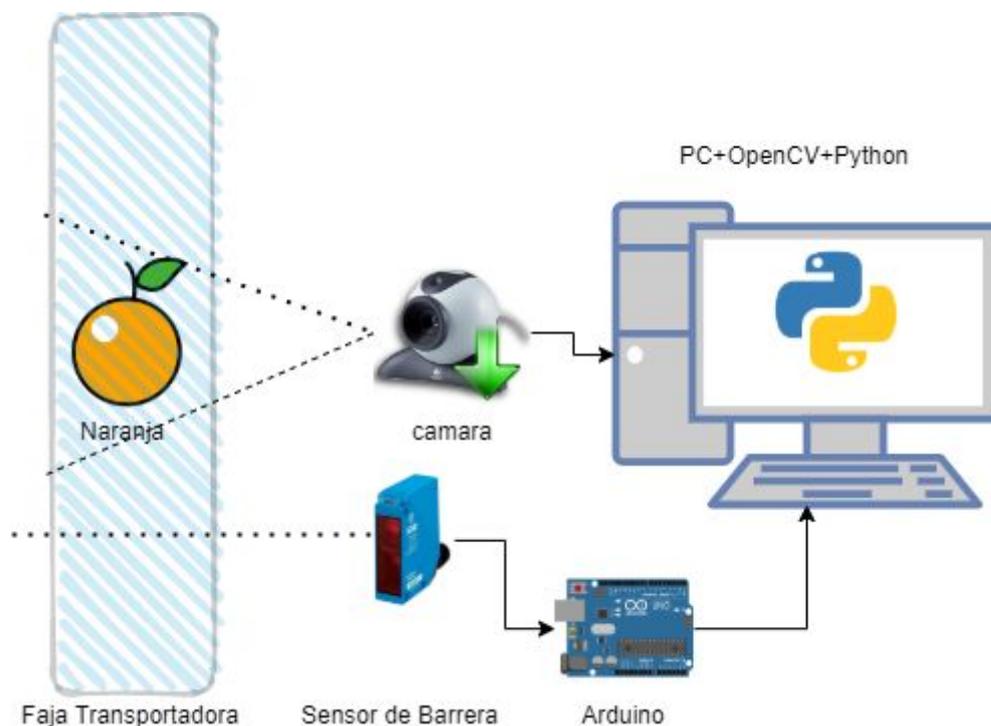


Figura 2, escenario del estudio, una faja transportando naranjas, un sensor de barrera unida a una tarjeta arduino(tm) y una cámara, todo esto procesado en una aplicación escrita en Python, Fuente, Elaboración propia.

Describimos los pasos previos realizados en las imágenes capturadas; primero la captura de imágenes mediante una cámara; luego la conversión a una escala de grises ya que Canny solo acepta imágenes en gris; seguidamente la aplicación de un filtro gaussiano para eliminar el ruido de la imagen, ya que este afecta la eficiencia del algoritmo Canny, esto trae como consecuencia una pérdida en la “nitidez” de la imagen, pero un incremento en la eficiencia de la detección de bordes; finalmente se aplica el algoritmo Canny con valores adecuados de histéresis, es decir límites superior e inferior a la imagen tratada y se obtiene los bordes detectados (*Figura 2*) (Himani Singh Rana, 2018),

La Figura 3 muestra la segmentación de la captura en un área de interés donde se encuentra la naranja a analizar, este proceso se ejecuta para aligerar el proceso de la imagen, ya que no es necesario procesar los 640x480 píxeles de la imagen, solo la parte donde se encuentra la naranja, ya que procesar toda la matriz de la imagen es un gasto insulso de recursos.

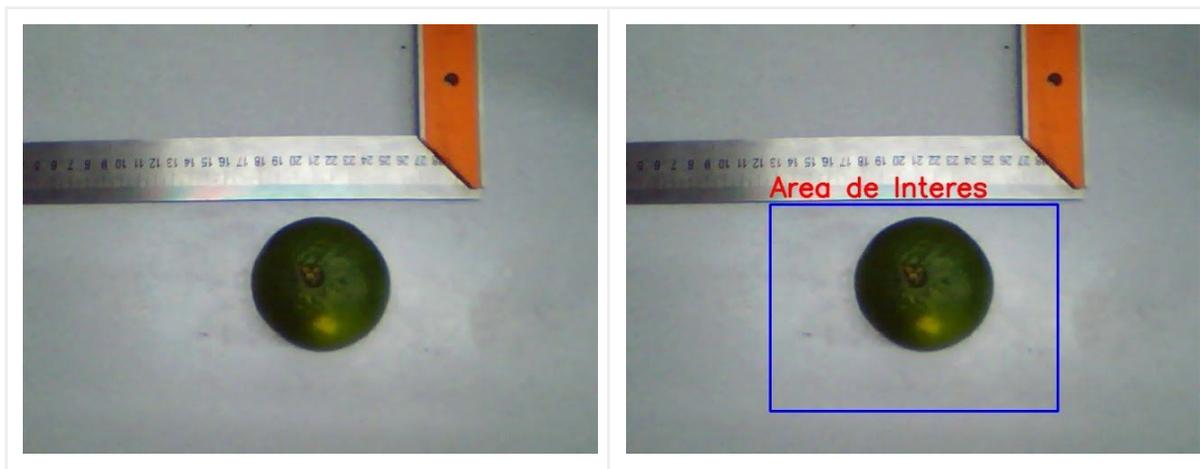


Figura 3, Obtención de la Imagen Original Fuente, Elaboración propia.

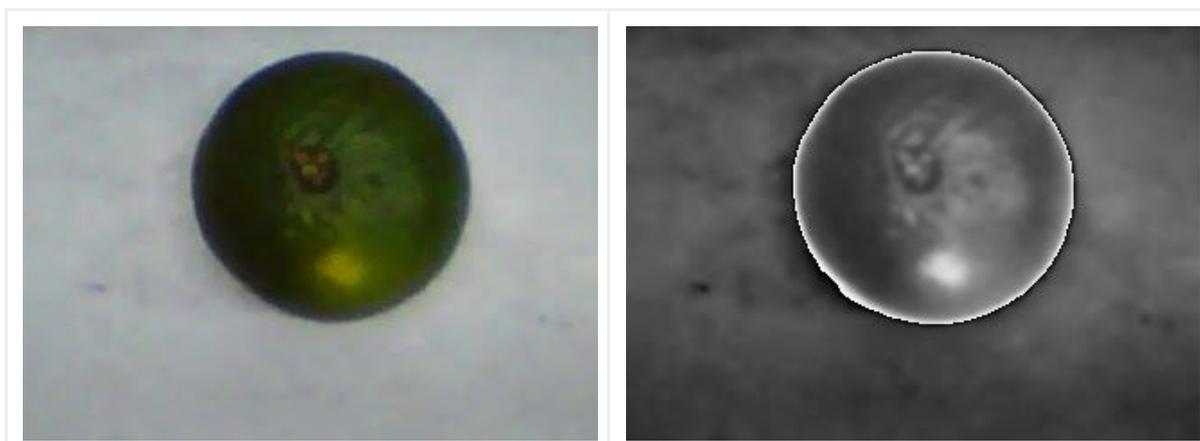


Figura 4, Imagen Original a Escala de Grises, Filtro Gaussiano aplicado a imagen en gris Fuente, Elaboración propia.

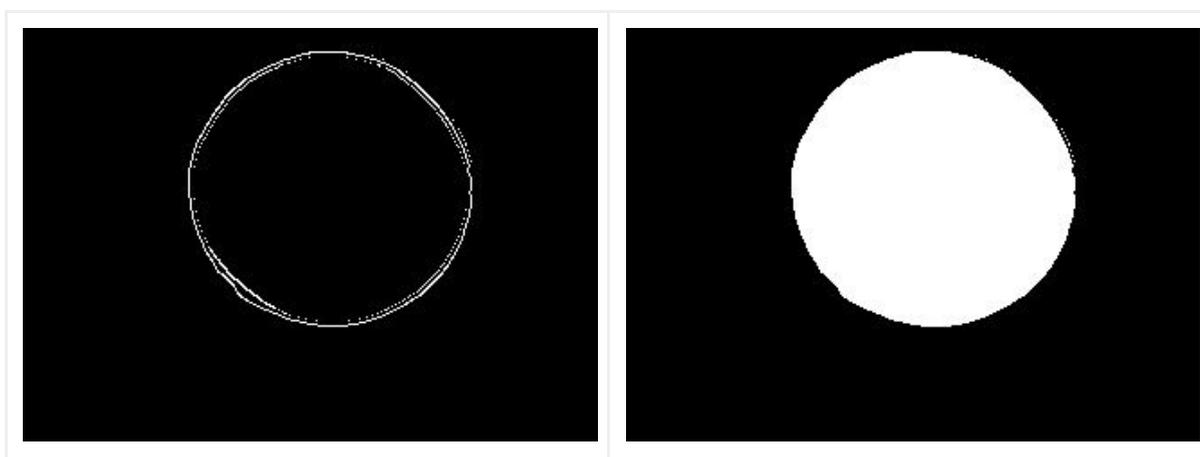


Figura 5, Aplicación del algoritmo de detección de bordes Canny, rellenado del contorno y binarización de la imagen mediante máscara de zeros. Fuente, Elaboración propia.

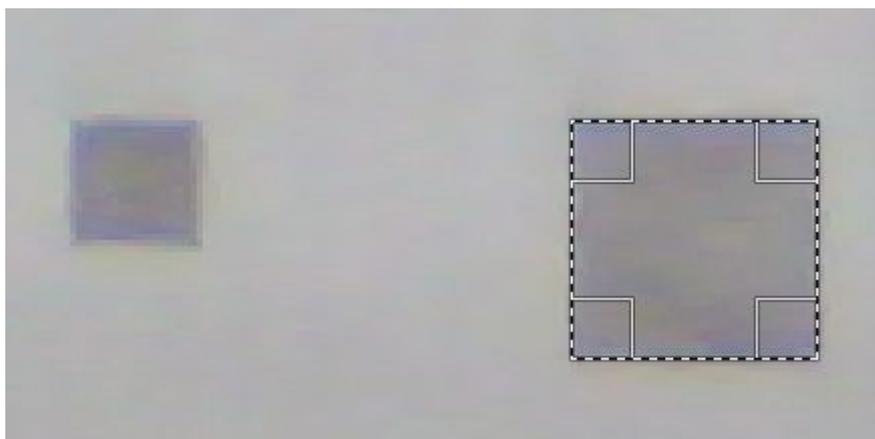


Figura 6, Imagen patrón de la proyección sobre el plano, el cuadrado seleccionado representa 4cm^2 en área, aplicando la detección de área a este patrón hallamos el factor de relación. Fuente, Elaboración propia.

Para el cálculo del tamaño de la naranja, usamos un aumento del contraste multiplicando matriz de grises por un porcentaje determinado experimentalmente, en nuestro caso elevamos el contraste al doble (2) en el caso de las naranjas y en 4 en el caso del área patrón, esto lleva a una mejor detección de bordes.

Una vez “oscurecido” la imagen a analizar, binarizar la imagen y hallamos el área encerrada entre los bordes, esta área está relacionada a los píxeles de la imagen, de esta manera calcular el tamaño en cm^2 .

De los análisis realizados a la muestra patrón se tiene el siguiente gráfico de distribución.

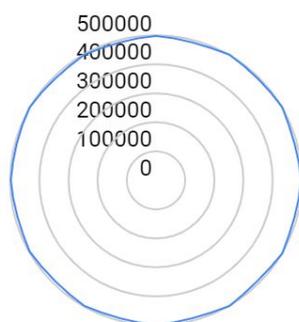


Figura 7, Diagrama de frecuencias del patrón de 2cm^2 , la mediana del patrón es 497760. Fuente, Elaboración propia.

El patrón tiene un “valor” analizado de 497760, por lo que por una relación simple podemos linealizar la salida en cm^2 de un valor determinado de una imagen de una naranja.

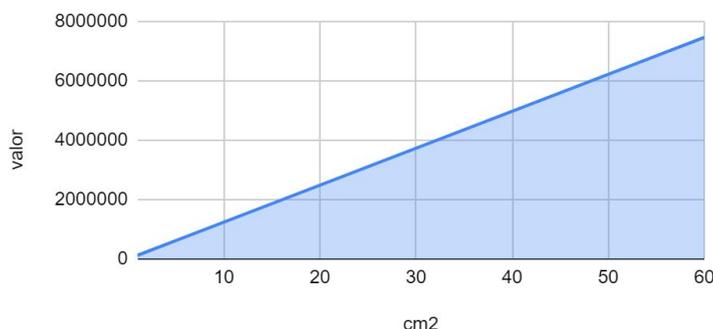


Figura 8, Gráfico de cm^2 versus el patrón esperado de “valor” encerrado en el área encontrada. Fuente, Elaboración propia.

III. RESULTADOS

Se observó que los valores de “borde” para la histéresis se pueden mejorar si incrementamos el contraste de la imagen al convertirla en escala de grises.

Como la detección de bordes usa grises como entrada, las sombras toman un papel importante ya que puede modificar el borde a detectar, en esta aplicación controlamos esa variable usando una iluminación elevada y aumentando el contraste, esto multiplicando la matriz de grises por una constante entera, todo esto con *numpy* por detrás.

Obtuvimos buenos resultados de threshold adaptativo sencillo, usando la mediana estadística, observamos que la mayoría de datos se encuentra entre los cuartiles 1 y 3 por lo tanto usamos un valor de borde inferior al cuartil 1. Gracias a este método el valor de threshold se “adapta” al ambiente de la imagen.

Al finalizar obtuvimos una aplicación sencilla que puede evaluar y validar el tamaño de una naranja en una faja transportadora, esta aplicación se puede usar no solo en este caso específico sino se puede extrapolar a distintas variedades de productos alimentarios haciendo pequeñas modificaciones al código.

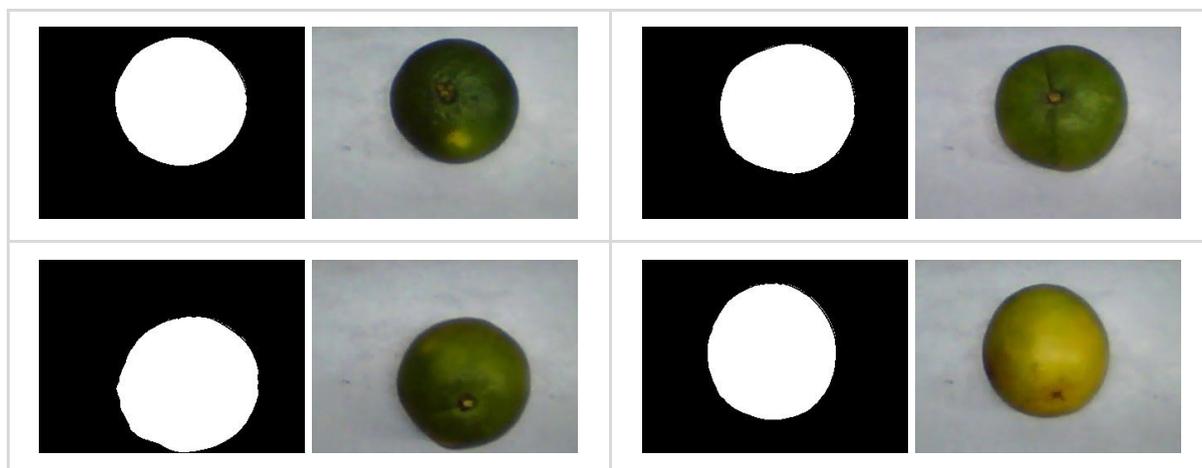


Figura 9, Comparativa de las imágenes de las regiones de interés y la salida binaria de la detección de bordes, se nota una buena respuesta en la detección de bordes. Fuente, Elaboración propia.

Tabla 1

Ejemplo de 04 medidas hechas con la aplicación desarrollada.

Serie	Fecha	Medida	cm ²	Muestra	Radio(cm)	Diámetro(cm)
0	2020-12-31	5099109	41	000	3.61	7.22
1	2020-12-31	4981351	40	001	3.57	7.14
2	2020-12-31	5027485	40	002	3.59	7.17
3	2020-12-31	5150097	41	003	3.63	7.26
4	2020-12-31	5007821	40	004	3.58	7.16
Promedios					3.61	7.22

Nota. Se muestra la salida del sistema en 4 muestras (para el estudio se hicieron muchas más), el dato *Medida* es corregido con el factor hallado del patrón de 2cm² y por relación se halla el radio y diámetro de la naranja. Fuente, Elaboración Propia.

IV. DISCUSIÓN

Experimentalmente se observó el siguiente escenario, cuando la imagen era tomada de costado, esta ocasiona sombra, está siendo gris, no se puede eliminar antes de la aplicación de Canny, como se describe a continuación.

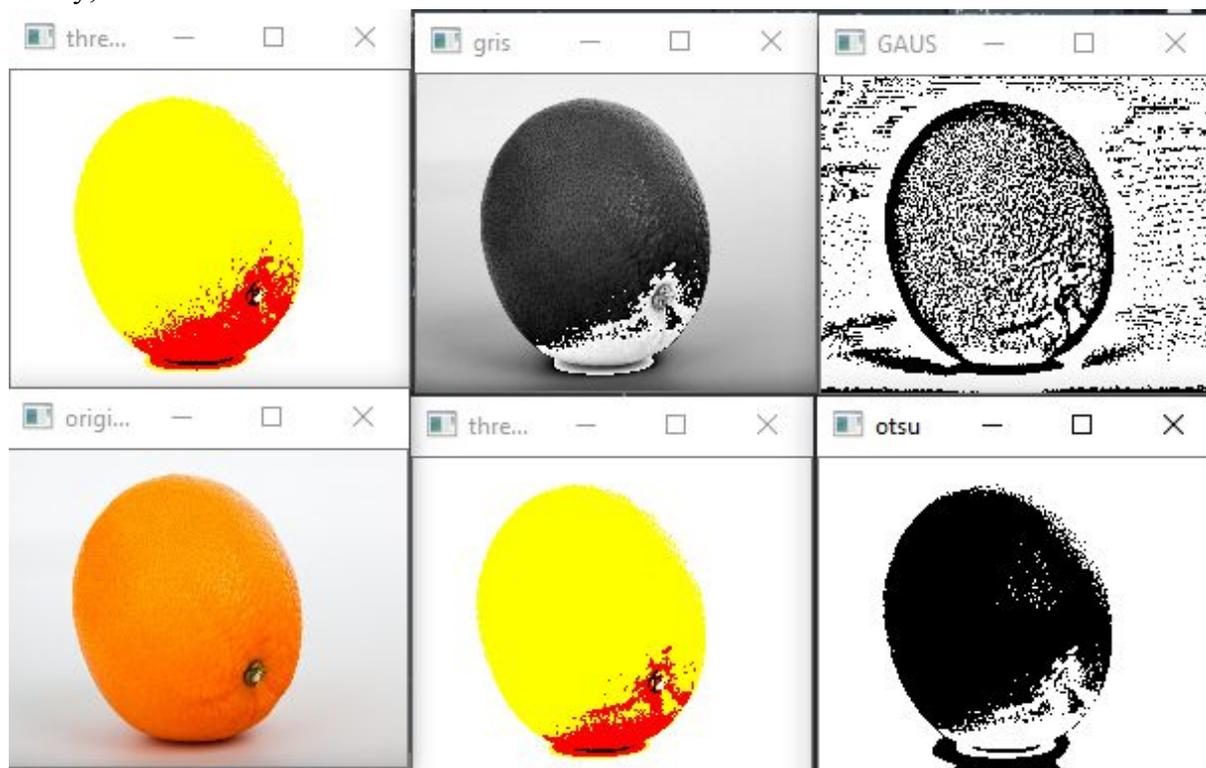


Figura 10, Escenario donde la imagen es tomada de costado, la sombra del producto crea una falsa área al ser esta de color gris. Fuente, Elaboración propia.

En la parte de abajo se observa qué la “sombra” al momento de la binarización se convierte a negro, incluso después de un aumento de contraste. Por lo que se recomienda para esta aplicación la toma de imágenes desde una ubicación elevada respecto al producto (Ehsan Akbari Sekehravani, 2020)

La aplicación se puede mejorar si usamos algún algoritmo de reconocimiento junto al de detección de borde, este puede ser K-NN con *haarcascade* o quizá *YOLO (You Only Look Once)*, de esta manera el algoritmo de reconocimiento reconoce la fruta y el algoritmo Canny trabajaría sobre la región de interés detectada, hay que notar que el algoritmo *YOLO* es muy rápido para ser de solo una pasada (Redmon, 2018).

V. CONCLUSIONES

Se propone Canny como un algoritmo rápido para cálculo de tamaños de productos en fajas transportadoras. El presente estudio usó naranjas como aplicación pero el mismo principio se puede aplicar a cualquier producto cuyo tamaño se puede calcular mediante una proyección plana, por ejemplo pescados, frutas cítricas, tubérculos.

REFERENCIAS

- Canny John (1986), *A Computational Approach to Edge Detection*, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. PAMI-8, NO. 6, NOVEMBER 1986
- Redmon Joseph, Farhadi Ali (2018), *YOLOv3: An Incremental Improvement*, arXiv:1804.02767v1 [cs.CV] 8 Apr 2018
- Himani Singh Rana, Himanshu Sirohia (2018), *Comparative Study Between Canny and Sobel Edge Detection Techniques*, EIJO Journal of Engineering, Technology And Innovative Research (EIJO–JETIR) (2018)
- Ehsan Akbari Sekehravani, Eduard Babulak, Mehdi Masoodi (2020), *Implementing canny edge detection algorithm for noisy image*, Bulletin of Electrical Engineering and Informatics Vol. 9, No. 4, August 2020, pp. 1404~1410

Fecha de recepción: 06/01/2021

Fecha de aceptación: 01/03/2021