



Licencia Creative Commons Atribución- No Comercial – Compartir Igual 4.0 Internacional

Lámpsakos | N° 15 | pp. 43-50 | enero-junio | 2016 | ISSN: 2145-4086 | Medellín-Colombia

Desarrollo e Implementación de un Sistema de Visión Artificial Basado en Lenguajes de Uso Libre para un Sistema Seleccionador de Productos de un Centro Integrado de Manufactura (CIM)

Development and Implementation of an Artificial Vision System Based on Free Use Languages for the Coach System Products on the Integrated Manufacturing Center (IMC)

Sebastián Amaya-Zapata, Est.

Grupo GEA, Institución Universitaria Salazar y Herrera
Medellín, Colombia
saz.2192@gmail.com

David Pulgarín-Velásquez, Est.

Grupo GEA, Institución Universitaria Salazar y Herrera
Medellín, Colombia
davidpulgarin15@gmail.com

Ingrid-Durley Torres-Pardo, PhD(C).

Grupo GEA, Institución Universitaria Salazar y Herrera
Medellín, Colombia
indrid.torres@iush.edu.co

(Recibido el 01-09-2015. Aprobado el 16-12-2015)

Estilo de Citación de Artículo:

S. Amaya-Zapata, D. Pulgarín-Velásquez, I.D. Torres-Pardo, "Desarrollo e Implementación de un Sistema de Visión Artificial Basado en Lenguajes de Uso Libre para un Sistema Seleccionador de Productos de un Centro Integrado de Manufactura (CIM)", *Lámpsakos*, N° 15, pp 43-50, 2016.
DOI: <http://dx.doi.org/10.21501/21454086.1702>

Resumen. Este artículo presenta el desarrollo de un sistema seleccionador de productos guiado por visión artificial, utilizando una cámara web integrada al software libre *Python*, que trabaja en conjunto con las librerías *open cv*. El principio de funcionamiento del seleccionador, está basado en un posicionador de objetos, previamente configurado por el usuario usando el sistema de colores *RGB* frente a la lente de la cámara, la cual se encargará de capturar la imagen y procesarla por medio del lenguaje de software. El sistema, permite informarle al usuario si la pieza ingresada al sistema es de los colores y la forma deseada por él. Con esto se logra una clasificación de objetos eficaz tanto en color, como en forma.

Palabras clave: Visión artificial; Manufactura; Línea de producción.

Abstract. This project is a products selector system guided by machine vision, using an integrated webcam by python free software in conjunction with the OPEN-CV software. The operating principle is positioning objects of a color previously set by the user, using the RGB color system in front the lens, which take care of image capture and processing through language software. The system, permits inform the user if the piece have the color entered and the shape requested. With this program we made an effective identification both in color and shape.

Keywords: Artificial Vision; Manufacturing; Production line.

1. INTRODUCCION

La visión artificial trata de simular los procesos visuales del hombre y analizarlos por medio de un cerebro al igual que los seres humanos, de tal manera que si los humanos pueden transmitir imágenes tomadas por medio de la vista y analizarlas utilizando pulsos enviados al cerebro, una máquina puede, utilizando una cámara web, captar imágenes y enviarlas a un procesador, listo para analizarlas de tal manera que la máquina pueda examinar el color y la forma de ciertos objetos. Aunque en esencia el funcionamiento parece simple, la tarea de identificar desde la cámara, un objeto específico, con el procesador, para que sea elegido por el seleccionador, resulta demandante en su programación. Este caso, constituye una situación real de aplicación de un sistema de visión artificial, articulado con un seleccionador de productos, de un centro Integrado de Manufactura, el cual permite al usuario, interactuar mediante una interfaz gráfica, para seleccionar el objeto en forma y color, que requiere sea elegido por el seleccionador, a fin de automatizar procesos de producción y de igual manera se busca observar cómo el software libre permite el uso y el diseño de un mayor número de aplicaciones para el reconocimiento de imagen y la clasificación de objetos, lo que conlleva a una mejor calidad en términos de identificación de variables y su tratamiento y al mismo tiempo permite una mayor compatibilidad con diferentes plantas lo que facilita su utilización.

Con el fin de brindar más detalle del caso de aplicación desarrollado, este artículo se organiza como sigue: la sección 2 detalla cada etapa del método propuesto; la sección 3 se ocupa de contextualizar los principales conceptos asociados con el tema de aplicación; la sección 4, implementa una a una las etapas del método pero esta vez orientadas a la consecución del robot objetivo. La sección 5, recopila los resultados obtenidos y finalmente, en la sección 6, se presentan las conclusiones y trabajos futuros.

2. TRABAJOS RELACIONADOS

Actualmente se han hecho muchos avances con respecto a la visión artificial y a los sistemas de procesamiento de imágenes, por el hecho de que este es un campo que tiene mucho potencial y muchas instituciones, tanto universidades como industrias, han estado investigando cómo aplicar y cómo mejorar los actuales sistemas de visión artificial y procesamien-

to de imagen; en esta sección se mostrarán algunos trabajos relacionados con el tema de la visión artificial y su utilidad en el desarrollo de la investigación.

El informe "Measuring system of the slab's position in a metallurgical plant using artificial vision techniques" [4], muestra la creación de un sistema capaz de utilizar cámaras de resoluciones comunes que le permitan al usuario identificar las mejores posibilidades lumínicas del ambiente, esto puede aportar a la investigación el análisis de luz para el debido reconocimiento de la imagen ya que la luz del ambiente y la luz de las lámparas fluorescentes pueden afectar la medición; a su vez las condiciones de luz en las que va a trabajar el sistema de reconocimiento de imágenes son relativamente variables y el hecho de analizar la luz ambiental y tener un método que analice variables como saturación y luminancia le pueden aportar mucha más precisión al sistema a desarrollar [1].

Una de las implementaciones de los sistemas de visión artificial más interesantes posee gran similitud a lo que se busca en esta investigación ya que habla de la implementación de un sistema clasificador de papas el cual usa un sistema de cámaras tanto ópticas como laser, que le permiten al sistema analizar el color y el diámetro de las papas para su posterior clasificación, clasificando 750 papas por hora, el conocimiento de este sistema sería un gran aporte a la implementación del sistema propuesto [3].

[6] A la hora de trabajar con visión artificial la información del procesamiento de imágenes es uno de los aspectos más importantes en lo que a visión artificial se refiere y que es un tema de discusión ya que el fundamento principal para un sistema de visión artificial es el procesamiento de imagen de los humanos, ¿Cómo trabaja el ojo humano? , ¿Cómo procesa el ojo humano?, al mismo tiempo [7] un robot totalmente capaz de interactuar con los humanos basándose en sistemas de reconocimiento facial para identificar la persona con la cual interactúa y además de reconocimiento de gestos faciales es una gran muestra del alcance de la visión artificial.

Otros avances en términos de visión artificial han sido la implementación de sistemas de visión artificial a robots motorizados tanto terrestres (carros) [5], como aéreos [2], estos aplicativos muestran como la visión artificial puede dar cierta autonomía a una máquina.

3. CENTRO INTEGRADO DE MANUFACTURA (CIM)

Un Centro Integrado de Manufactura, es aquel sistema que proporciona una estructura que agiliza la descripción, la ejecución, y el planteamiento de un proceso de producción. El seleccionador de productos, por su parte, es un elemento clave, en la articulación de la elaboración, ya que permite seleccionar y distribuir las piezas por tipo de material y color.

Actualmente, los seleccionadores actúan de dos formas, i) articulados con un mecanismo semiautomático que permita la producción a través de la un hardware y la intervención de un humano, o, ii) a través de un programa de software que automatice el proceso de selección. En este último caso, una de las técnicas más exitosamente aplicada, está representada en un programa de software, que actué con “inteligencia”, comprendiendo lo que desea el usuario y comparándolo con distintos algoritmos a fin de detectar mediante la segmentación el reconocimiento de ciertos objetos.

La propuesta de este artículo, relacionada con el mecanismo funcional del seleccionador, se apoya en la utilización de una cámara web, y un programa de software, que selecciona el color de la pieza, y el tipo de forma de la misma, mediante una técnica de segmentación, de esta manera identifica el objeto, que el usuario desea seleccionar en el proceso de producción industrial. El software, está conectado a los inputs del PLC del hardware del seleccionador, de esta manera la selección digital, se ejecuta en real. El motor y las electroválvulas están conectados a los outputs del PLC, tal como se muestra en fig. 1.

Este modelo se puede utilizar en diversas industrias ayudando a la selección y distribución de distintas piezas por el tipo de material y color, pudiendo mejorar la productividad de la empresa, reduciendo los costos de producción y mejorando la calidad de la misma.

El prototipo de selector automático de materiales y colores nos permite realizar las operaciones difíciles de controlar manualmente [11], simplificando el trabajo de forma que el proceso no requiera de constantes verificaciones en la selección de las piezas por tipo de material y color.

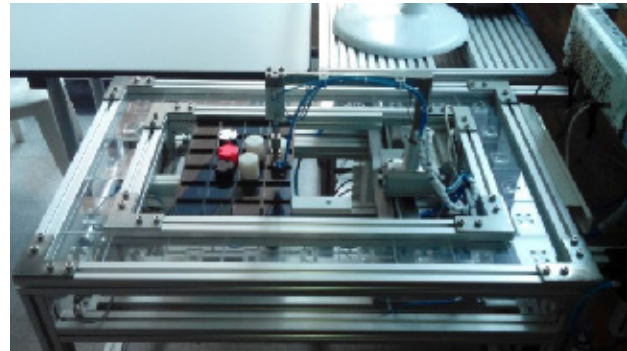


Fig. 1 Sistema seleccionador de productos

4. ARQUITECTURA DEL SISTEMA PROPUESTO

El seleccionador de productos actuará sobre la identificación de las características básicas de color y forma de un objeto, utilizado para ello, un software, que se encargará de hacer los múltiples filtros necesarios para la identificación de imágenes; además de contar con una interfaz gráfica donde el usuario ingresará los valores de los colores y la forma que desea identificar; el ordenador donde se hará el procesamiento de la imagen, actuará en concordancia con una cámara web conectada de manera directa, que reconocerá los objetos al igual que los reconoce el ojo humano, y un micro controlador (arduino nano) encargado de hacer la comunicación con el PLC y las diferentes piezas con las cuales se comprueba el funcionamiento de un sistema de control [10]. Los elementos, que conforman la articulación del sistema pueden ser visualizados en la fig. 2, donde sus principales componentes se agrupan en hardware y software.

4.1. Hardware

En hardware se citan las siguientes partes:

a) **Cámara** o sensor que permita capturar la imagen y transmitirla a un ordenador en donde serán procesadas y se tomarán las debidas decisiones para su posterior clasificación.

b) Un mecanismo de procesamiento de imagen que necesita un procesador y algoritmos que permitan hacer un filtrado de la imagen, reconocimiento de las formas y su clasificación

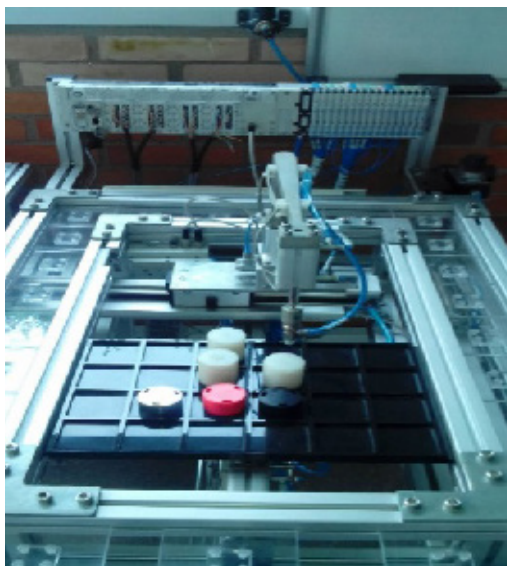


Fig. 2 Sistema seleccionador de productos

c) Un sistema de clasificación de productos el cual está conformado por distintas piezas de hardware, con las cuales es posible ejecutar las actividades necesarias para la debida clasificación de las piezas según su color y su forma.

El sistema seleccionador de productos requiere ciertas piezas fundamentales para su funcionamiento, tales como, un controlador que alimenta al sistema de piezas que por medio de actuadores electro-neumáticos se encargan de clasificar las piezas por forma y color según lo desee el usuario (ver fig. 2).

Una plataforma de desarrollo libre, como arduino la cual tiene como tarea hacer la comunicación entre el computador y el PLC (control lógico programable). Este último, encargado de generar los procesos que surtirán de piezas el sistema y que posteriormente clasificarán según como los pida el usuario (ver fig. 3).

4.2. Software

El programa de software, es el que ejerce la acción real de visión humana y está representado principalmente por un algoritmo seleccionador de imágenes el cual para el caso ha sido diseñado, utilizando lenguajes [13] y librerías de desarrollo libre [12]. La tabla 1, presenta un breve análisis de los lenguajes analizados, para su selección.

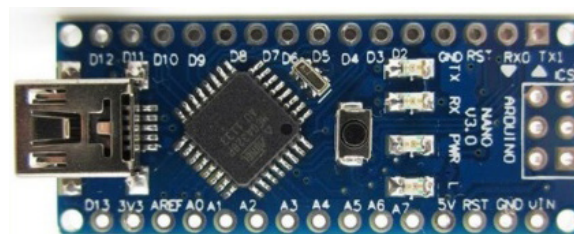


Fig. 3 Arduino nano

Para alcance de este trabajo se ha seleccionado Python, debido a su versatilidad, al mismo tiempo que la facilidad de ser utilizado con los protocolos de internet como HTTP, FTP, SMTP, XML-RPC, POP, IMAP, CGI.

5. IMPLEMENTACIÓN FUNCIONAL DE LA ARQUITECTURA

Como puede apreciarse en la fig. 4, el sistema es disparado por el humano, quien interactúa a través de una interfaz gráfica del software para seleccionar y elegir el objeto (en forma y color). Los detalles de los pasos funcionales son descritos a continuación, además de mostrar el resultado de los análisis para la selección del lenguaje y la librería sobre la cual se implementaría el sistema, y las diversas partes del código diseñado para el funcionamiento del sistema de identificación de imágenes.

5.1. Selección del lenguaje y la librería

Una vez analizados los lenguajes se determina que el mejor lenguaje para trabajar el sistema de visión artificial que se integrará en el centro integrado de manufactura es el lenguaje python junto con la librería open cv, con la desventaja de que esta es la única librería que maneja para el tratamiento de imágenes; python es un lenguaje de alto nivel que nos da la posibilidad no solo de modificar el programa en cualquier momento sino que la estructuración del lenguaje se realiza en un código simple y fácil de entender por si otro programador desea analizar el código implementado; además por ser un lenguaje libre existe gran cantidad de información acerca de este lenguaje, desde literatura básica para el uso de comandos, como cursos virtuales que le permiten al usuario aprender a manejar este lenguaje fácil y rápido; por último y lo más importante es el consumo de recursos que emplea este lenguaje, siendo un lenguaje de

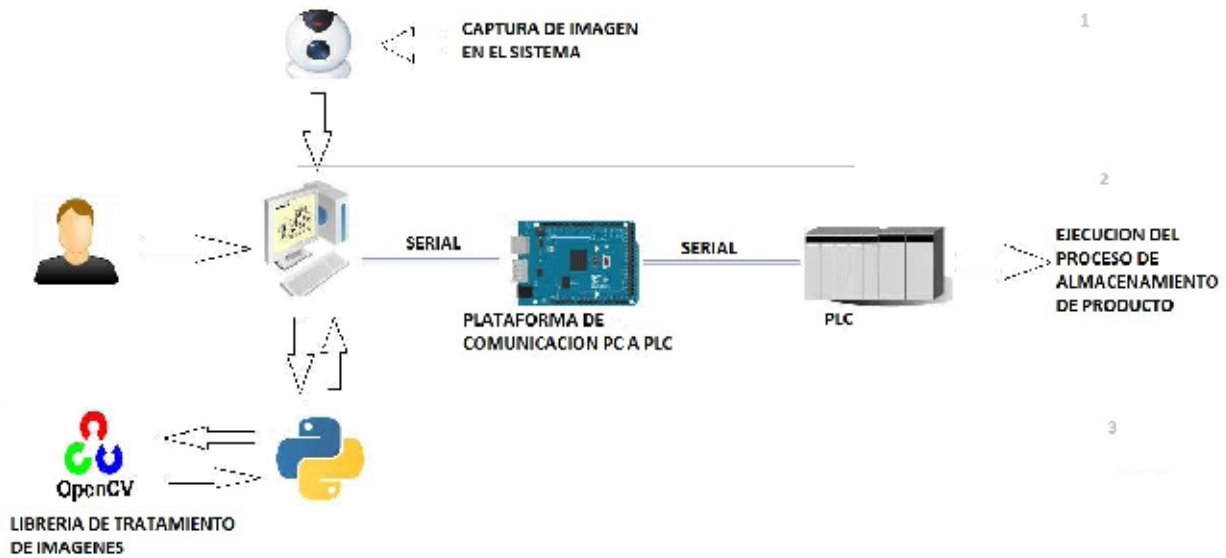


Fig. 4 Arquitectura del sistema

Tabla 1.

Comparación de los lenguajes de software libre

Lenguaje	Librerías compatibles	Ventajas	Desventajas
C++	Torch3vision, VLX, openCV	Es un lenguaje de bajo nivel lo que nos permite modificar todos los detalles del código si ningún problema, su ejecución es rápida ya que no consume gran cantidad de recursos, es compatible con todas las librerías.	La complejidad de sus códigos es alta lo cual lo hace difícil de leer y de ejecutar en plataformas diferentes.
Java	openCV	Es un lenguaje interpretado, lo que nos permite ejecutarlo bajo cualquier plataforma sin tener que recompilar el programa, hay mucha bibliografía disponible.	El consumo de recursos es alto por procesos innecesarios que se ejecutan, la coordinación con la librería resulta bastante complicada y la lectura del código aún es muy compleja.
Python	openCV	Es un lenguaje interpretado lo que permite ejecutarlo en cualquier plataforma sin necesidad de recompilar, su escritura y lectura de código es bastante simple facilitando la construcción de programas complejos y la integración a múltiples librerías y tiene una gran cantidad de bibliografía disponible.	Su consumo de recursos es alto aunque no tanto en comparación con otros lenguajes de alto nivel.

alto nivel se supondría que su consumo de recursos es muy elevado, sin embargo, es un lenguaje con un consumo de recursos bajo en comparación con los otros lenguajes de alto nivel lo que lo hace el lenguaje libre más viable para su utilización en el diseño del software de visión artificial.

5.2. diseño del software

El diseño se compone de tres partes las cuales se explicarán a continuación.

Interfaz gráfica: esta interfaz desarrollada en la librería PYQT como extensión del PHYTON nos permite interactuar fácilmente con el programa, otorgando una interfaz amigable e intuitiva para los usuarios.

Constructor: esta sección se encarga de ensamblar la interfaz gráfica y el programa de identificación de imágenes en un solo programa dándole así la funcionalidad a la interfaz.

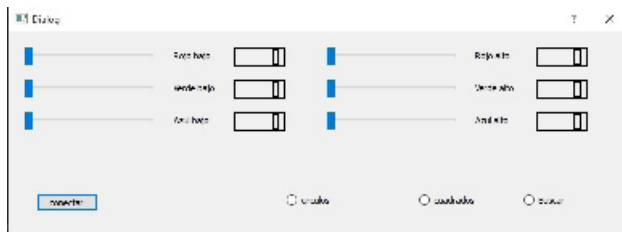


Fig. 5 Interfaz del sistema

Cuerpo principal: para finalizar nos encontramos con el cuerpo principal del programa, el cual nos permite generar la aplicación compatible con Windows y así poder ejecutar el programa en cualquier computador con Windows.

5.3. Etapas de funcionamiento

Etapa de Configuración: en esta etapa el usuario se encarga, por medio de la interfaz, de darle la orden al sistema, de los rangos de colores en RGB que serán necesarios para la identificación del objeto en su color. Mientras la forma será seleccionada a través de la selección de una de las opciones de los tipos de objetos disponibles (ver fig. 5).

Etapa de Visión: una vez seleccionados los valores de rango en los cuales se desea identificar el objeto, se pasa a la etapa de visión donde por medio de una cámara física, la cual debe ser integrada por medio de código en el sistema (ver fig. 6), se capta la imagen que luego es identificada.

Etapa de Reconocimiento: la etapa de reconocimiento se encarga de filtrar las imágenes captadas y enviadas por la cámara; este reconocimiento se hace mediante el análisis de fotogramas [9] (análisis de una foto por lapso de tiempo), en la cual se identifica en primer lugar el color (ver fig. 7), una vez se tenga el color que el usuario determinó en la interfaz se analizan los fotogramas (los fotogramas por segundo son modificables en el código); con estos fotogramas y los filtros de formas, el software determinará la forma del objeto, si es un círculo un cuadrado entre otras figuras.

Etapa de Clasificación: esta es la etapa final del proceso en donde una vez se tenga determinado el color y la forma de la pieza se manda una instrucción a un arduino (ver fig. 8), que es una plataforma de

```
def iniciar_camara(self):
    ser = serial.Serial('COM5', 9600)
    while True:
        com_ent = ser.read()
        if com_ent == 'c':
            ser.write("r")
            break
    cap = cv2.VideoCapture(1)
```

Fig. 6 Etapa de visión

programación y prototipado, la cual es el medio de comunicación y el encargado de enviar las instrucciones requeridas por el ordenador al sistema electro neumático.

6. RESULTADOS

Las siguientes figuras muestran los resultados obtenidos en la identificación, para obtener el reconocimiento de color y forma mostrados, en la fig. 9 y fig. 10 se ubicó la ficha en frente de la cámara, para continuar con la modificación aleatoria de los diferentes valores utilizando el rango de colores RGB (red, Green, blue) debido a que la calidad de la identificación se ve afectada por dos factores básicos [8], el primero es la resolución de la cámara, si se posee una cámara de mayor resolución el color sería más real y no se alterarían los colores con base en el color primario del lente, en este caso el azul, y como segundo factor influyente es la luz, esta altera también el color de la ficha ya sea aclarándolo u oscureciéndolo.

El sistema seleccionador de productos una vez terminada su implementación obtuvo los siguientes resultados en la adquisición y reconocimiento de las imágenes (ver fig. 9)

Se observa como el sistema está identificando los objetos de color plateado y al mismo tiempo esta identificando y afirmando que la figura es un círculo.

Se muestra cómo el sistema identifica el color naranja y resalta el hecho de que es un cuadrado (ver fig. 10).

```
_, imagen = cap.read()
# Establecemos el rango de colores que vamos a detectar
color_altos = np.array([ra, ga, ba], dtype=np.uint8)
color_bajos = np.array([rb, gb, bb], dtype=np.uint8)
# Crear una mascara con solo los pixeles dentro del rango de color
mask = cv2.inRange(imagen, color_bajos, color_altos)
```

Fig. 7 Etapa de reconocimiento

```
elif self.ui.boton_buscar.isChecked()== True and self.ui.boton_cuadrados.isChecked()== True:
    i = 0
    for extension in areas:
        if extension > 2000:
            actual = contours[i]

            # Aproximar el numero de vertices
            approx = cv2.approxPolyDP(actual, 0.01 * cv2.arcLength(actual, True), True)

            # la deteccion del objeto se hace por el numero de vertices detectados
            if len(approx) == 4:
                cv2.drawContours(imagen, [actual], 0, (0, 0, 255), 2)
                ser.write('d')

            #enviar caracter de deteccion al arduino
```

Fig. 8 Etapa de clasificación

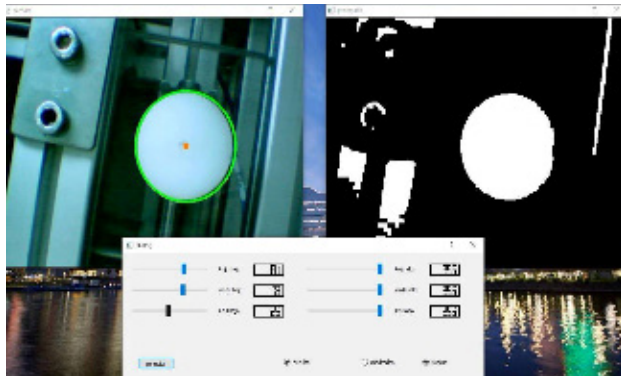


Fig. 9 Reconocimientos de círculos

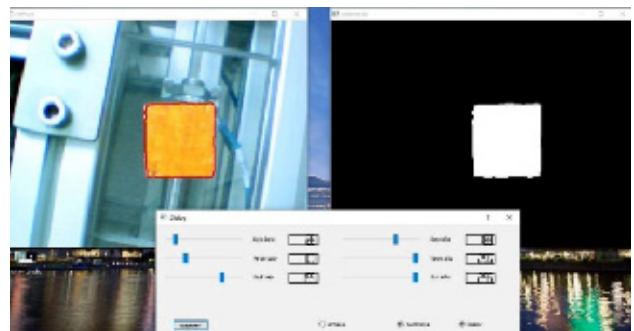


Fig. 10 Reconocimientos de cuadrados

7. TRABAJO FUTURO

Como trabajo se espera, ampliar el conjunto de objetos con los cuales pueda trabajar el seleccionador, variando en materiales como sintéticos, cueros y texturas textiles.

8. CONCLUSIONES

La visión artificial es una herramienta muy útil al momento de automatizar un proceso de control.

Actualmente la visión artificial es cada vez más fácil de implementar gracias a los lenguajes y plataformas de desarrollo libre. Este trabajo, es una evidencia de una actividad simple de reconocimiento de objetos y su articulación para que el sistema trabaje de manera coordinada, y automatizar así procesos de producción.

Los lenguajes libres otorgan al usuario una mayor comodidad a la hora de programar un entorno de visión artificial, no solo por su gran compatibilidad con múltiples sistemas y librerías, sino también por su estructura de código simple y fácil de entender lo

que le permite al usuario adaptar estos sistemas en muchos más ambientes tanto en la industria como en la academia.

El uso de visión artificial en un seleccionador de productos es una herramienta de gran ayuda que permite agilizar los procesos de selección y clasificación de objetos.

Como trabajo futuro, se propone incrementar la base de datos de objetos y ampliar la escala de color, además de permitir que el sistema pueda ser adaptado a otros modelos de producción.

REFERENCIAS

- [1] J. Angulo y J. Serra, "Image Color Segmentation using Bi-variate Histograms in Luminance/Saturation/Hue Polar Color Spaces," *Comp. y Sist.*, vol. 8, no. 4, pp. 303-316, 2005.
- [2] A. Gaujens, A. Benini, A. Mancini y S. Longhi, "Autonomous navigation, landing and recharge of a quadrotor using artificial vision," de *IEEE/ASME 10th International Conference on, orlando*, 2014.
- [3] Y. Fernandez, A. Sariana y R. Swenson, "Fernandez, Y., Sariana, 'Development of a prototype for classification of potato mini-tubers based on artificial vision'," de *Electrical Engineering, Computing Science and Automatic Control, CCE, 2009 6th International Conference on, Toluca*, 2009.
- [4] L. Frangu y S. Caraman, "Measuring system of the slab's position in a metallurgical plant using artificial vision techniques," de *Electronics, Circuits and Systems, 1999. Proceedings of ICECS '99. The 6th IEEE International Conference on (Volume:2)*, Pafos, 1999.
- [5] F. Gonzalez, J. Guarnizo y G. Benavides, "Emulation System for a Distribution Center Using Mobile Robot, Controlled by Artificial Vision and Fuzzy Logic," *Latin America Transactions, IEEE (Revista IEEE America Latina)*, vol. 12, no. 4, pp. 557-563, 2014.
- [6] J. Liu y X. Gou, "Information processing model of artificial vision prosthesis," de *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on (Volume:2)*, Chengdu, 2010.
- [7] C. Salcedo, C. Pena, J. de Jesus Cerqueira y A. Lima, "Designing a Real Time Artificial Vision System for Human Interaction with an Omnidirectional Mobile Platform," de *Robotics Symposium and Latin American Robotics Symposium (SBR-LARS), 2012 Brazilian*, Fortaleza, 2012.
- [8] J. Humberto, *Visión artificial : rasgos descriptores para el reconocimiento de objetos*, Madrid: Ma-Ca S.A., 2014.
- [9] J. Humberto y R. Rodriguez, *Procesamiento y Análisis Digital de Imágenes*, Madrid: Ma-Ca S.A., 2011.
- [10] J. Monso, *Sistemas de identificación y control automáticos (II)*, Barcelona: Marcombo, S.A., 1994.
- [11] A. Sanchez, *Aplicaciones de la visión artificial y la biométrica informática*, Madrid: Editorial Dykinson, 2005.
- [12] Comunidad Open CV, "open CV", Documentación [Online]. Available: www.opencv.org
- [13] Comunidad Python, "Python Programming", [Online]. Available: <https://docs.python.org/2/faq/general.html#what-is-python-good-for>.