

Research Article

Design and Implementation of an Artificial Vision System for Quality Control in the Labeling of Commercial Products

Diseño e Implementación de un Sistema de Visión Artificial para el Control de Calidad en Etiquetado de Productos Comerciales

Omar Barahona^{1*} and Gabriel Moreano^{1,2}

IV INTERNATIONAL
CONGRESS OF
ENGINEERING FACULTY OF
MECHANICS (IV COINTI
2021)

¹Facultad de Mecánica, Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador
²Pontificia Universidad Católica del Perú, Lima, Perú

ORCID

Omar Barahona: <https://orcid.org/0000-0002-1624-9746>

Corresponding Author: Omar Barahona; email: omar.brahona@gmail.com

Published: 31 August 2022

Production and Hosting by
Knowledge E

© Omar Barahona and Gabriel Moreano. This article is distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use and redistribution provided that the original author and source are credited.

Abstract

Large industries are continually looking for strategies and tools that help them increase their productivity, to cover larger markets in shorter times. However, on certain occasions, these types of strategies with the increase in production speed can affect the product quality, sometimes triggering the discomfort of the clients or final recipients. This is why companies with continuous production lines, in addition to seeking productive optimization strategies, nowadays seek strategies that help them maintain the quality of their products. This work presents the development of a classifier algorithm through road inspection that verifies the correct labeling of different commercial products, thus helping to maintain the final quality of the products in a production line. The algorithm has been developed in an open-source programming language with a camera of medium characteristics, to reduce commercial costs and make it a strategy for easy massification. In addition, the algorithm can be easily adapted for different types of products, that is, it is an open and undeveloped strategy for a single type of label. The algorithm has an interface that will help users and interested parties in its use. At the end of the work, an artificial vision labeling identification system is available that is capable of detecting flaws in any type of label at an estimated speed of 2 sec per product.

Keywords: *artificial vision, Python, OpenCV, labeling, automatic classification.*

Resumen

Las grandes industrias continuamente andan buscando estrategias y herramientas que las ayuden a incrementar su productividad, para cubrir mercados más amplios en tiempos más cortos. En ciertas ocasiones este tipo de estrategias sacrifican la calidad de los productos por el incremento de velocidad de producción, desencadenando en algunas ocasiones el malestar de los clientes o destinatarios finales. Es así como las empresas con líneas de producción continua, a más de buscar estrategias de optimización productiva, hoy en día buscan estrategias que les ayuden a mantener la calidad de sus productos. En este trabajo se presenta el desarrollo de un algoritmo clasificador mediante inspección visual que verifica el correcto etiquetado de distintos productos comerciales, aportando así a mantener la calidad final de los productos en una línea productiva; el algoritmo se ha desarrollado en un lenguaje de programación de código abierto y con una cámara de características medias, con el fin de reducir los costos comerciales, y de que sea una estrategia de fácil masificación. Además, el algoritmo se puede adaptar de forma sencilla para distintos tipos de productos, es decir, es una estrategia abierta y no desarrollada para un único tipo de etiquetas. El algoritmo cuenta con una interfaz propia

 OPEN ACCESS



que ayudará a los usuarios y partes interesadas en su utilización. Al final del trabajo se dispone de un sistema de identificación de etiquetado por visión artificial que es capaz de detectar fallas en cualquier tipo de etiqueta a una velocidad estimada de dos segundos por producto.

Palabras Clave: *visión artificial, Python, OpenCV, etiquetado, clasificación automática.*

1. Introducción

Alrededor del mundo varias fábricas y empresas productoras han optado por incluir en sus líneas de producción la visión artificial para realizar la automatización de varios procesos, entre los cuales están la inspección de calidad, materiales, ensamble de partes, pintura, soldadura y seguridad [1]. Sin embargo no en todos los países se ha dado este desarrollo en la misma magnitud, según informes [2]. Este sector de la automatización es liderada por América del Norte, Europa y Asia-Pacífico, teniendo grandes inversiones y proyectándose a un crecimiento cada vez mayor. En América Latina, la visión artificial tiene una aplicación muy reducida, siendo Colombia el país con mayor inversión en implementar este sistema en sus empresas [3]. En el Gráfico 1 se compara el total de inversiones que cada una de las regiones antes mencionadas realizan en la inclusión de visión artificial en las industrias dentro de su territorio.

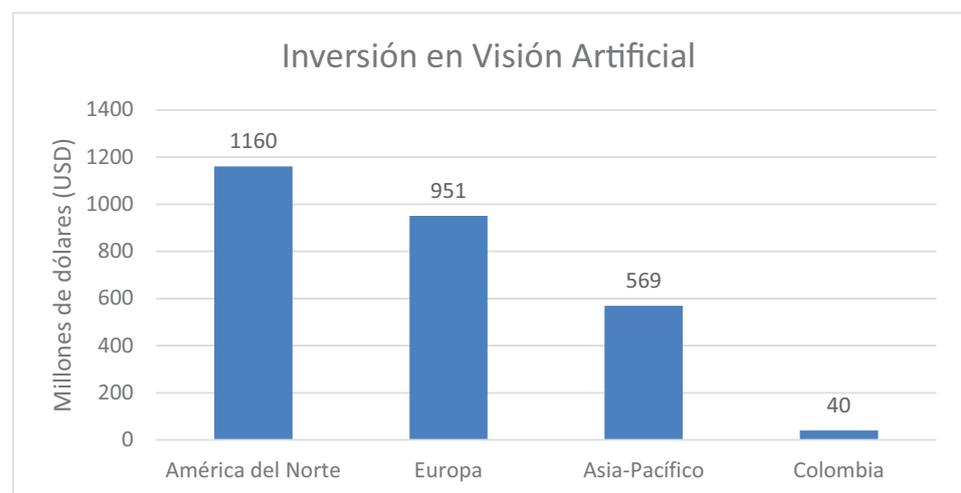


Figure 1

Inversión de regiones y países en inclusión de la visión artificial.

A nivel nacional, Ecuador presenta una aplicación tecnológica de este tipo muy baja o casi nula [3]. considerando que diversos sistemas que han sido investigados y desarrollados por personas o instituciones académicas solamente se han quedado en proyectos o ideas [4], ya que las industrias consideran que la automatización de sus procesos con la implementación de la visión artificial conlleva gastos elevados por



cuestiones de modificación en líneas de producción, licencias de software y adquisición de equipos tecnológicos avanzados, por lo que optan por mantener al personal humano en tareas de inspección visual o control de calidad.

La inspección visual tiene como ventaja de que casi cualquier producto puede ser inspeccionado, lo que lo hace un proceso de amplio campo de aplicación, como por ejemplo en la industria energética, alimentaria, petroquímica, de transporte y estructural [5] sin embargo también presenta algunas desventajas, como el hecho de que solo se pueden evaluar condiciones superficiales y que es un proceso repetitivo que causa fatiga en el operario prematuramente. En resumen, la inspección industrial puede ser vista como el análisis de características del producto final, procurando que estas se encuentren dentro de valores esperados [6].

El sistema que en el presente trabajo se propone se localizará al final de la línea de producción, donde se realice el proceso de empaquetado y etiquetado, con el fin de reconocer y reducir los problemas en etiquetado o impresión de las mismas. Estos errores a más de significar pérdidas de recursos de manera interna, pueden conllevar a la pérdida de credibilidad o preferencia de sus clientes, por lo que para reducir estas fallas, las industrias dedicadas al envasado y/o distribución de productos alimenticios, bebidas o productos en general, recurren a procesos de inspección, sea humana o con la implementación de visión artificial [7]. Es en este punto donde entra la automatización de la inspección visual, la cual, a pesar de ser un recurso muy importante ya que mejoraría considerablemente la producción y contribuiría a la optimización de recursos [8] ha sido implementada a un ritmo muy bajo si se lo compara con otros procesos donde actualmente no se necesita que el ser humano intervenga, esto se puede deber a varios factores, pero se considera que los dos primordiales son: los sistemas de inspección autónomos pueden requerir de varios sistemas conectados entre sí, lo que lo hace complejo de implementar y manejar; y el otro factor es el económico, ya que la implementación de un Sistema de Inspección Visual Automatizado (SIVA) o AVI (por sus siglas en inglés) requiere de la adquisición de varios equipos, software e incluso la modificación de la línea de producción [6].

Un sistema de este tipo, si bien es diseñado para ejecutar una sola función, también cuenta con un microprocesador, el cual es de tamaño reducido que facilita la implementación en un entorno reducido pero que a su vez es lo suficientemente capaz para ejecutar dicha función sin importar la complejidad [9] este tipo de sistemas son denominados como sistema embebido por ser dedicado a ejecutar una sola función de manera continua desde su activación hasta que el operario decida pausar o finalizar el proceso [10]. Uno de los componentes más importantes de cualquier sistema embebido es el apartado del software, mismo que gracias al avance tecnológico y a



la constante investigación se ha vuelto más accesible ya que actualmente existen softwares sofisticados de código libre, donde cualquier persona que tenga conocimientos en programación puede hacer cambios que considere pertinentes para cubrir sus necesidades [9].

Uno de los lenguajes de código abierto más usado en los últimos años es Python, lenguaje que es multiplataforma [11]. Python, al ser un lenguaje de tipado dinámico contribuye a desarrollar una programación más fluida e interactiva, esto permite realizar pruebas mientras el código aún se encuentra en desarrollo para encontrar fallas durante la escritura del mismo [12]. Cabe recalcar que Python proporciona el lenguaje, pero requiere de un editor de texto para que se pueda escribir las líneas de código que especifique las funciones que se desea desarrollar, que sea capaz de incluir y leer las distintas librerías necesarias para cubrir con la necesidad del programador, uno de los editores de texto más usados por científicos, ingenieros y estudiantes independientes, es Spyder, el cual permite codificar y ejecutar proyectos de manera sencilla e intuitiva [13].

La visión por computadora o comúnmente llamada visión artificial es un subcampo de la inteligencia artificial y es el conjunto de técnicas y modelos que permiten a un computador adquirir, procesar, analizar y explicar cualquier tipo de información tomada del mundo real obtenida en forma de imágenes digitales, con el fin de simular el proceso que realiza el ser humano para entender su entorno físico [14]. Implementar algoritmos dedicados a la visión artificial en áreas industriales es un tema de alto interés así como de alta complejidad, considerando que estos deben ser diseñados con una robustez, precisión y velocidad que sea capaz de resolver cientos de problemas en un instante [15]. La visión artificial puede ser comprendida de mejor manera si se la compara con la visión del ser humano, pasando por las siguientes etapas: etapa de adquisición, donde netamente se percibe (captura) la imagen a través de los ojos (cámara), la segunda etapa es la de procesamiento, es aquí donde se clasifica las figuras captadas dentro de la imagen (sometimiento a filtro o procesamiento de imagen) para posteriormente compararlo con algo ya conocido (clasificar imagen), finalmente se puede mencionar la etapa de reconocimiento y toma de decisiones, es decir, luego de haber reconocido la imagen se decide que acción realizar [16]. Existen varias herramientas y/o librerías que facilitan uso y aplicación de la visión artificial, sin embargo, OpenCV es una de las bibliotecas de código abierto más grandes y usada [13]. Al implementar esta biblioteca en Python, se puede analizar patrones de imágenes, reconociendo diversas características a través de espacios vectoriales y operaciones matemáticas [17].



2. Materiales y Métodos

Una vez definida la metodología del presente trabajo como cuantitativa, se procede a delimitar y especificar las etapas que se desarrollarán a lo largo del proyecto, mismas que se irán ejecutando de forma progresiva y secuencial, estas etapas se pueden resumir en la Tabla 1:

Table 1

Etapas del proyecto de clasificación automática.

Etapa	Descripción
Transporte	Se desarrollará un prototipo de una banda transportadora la cual movilizará los productos a través del sistema clasificador.
Control de señales	Se logrará utilizando una placa controladora (Arduino) la cual enviará las señales de activación a los actuadores que controlan el movimiento de la banda transportadora y del sistema de clasificación.
Reconocimiento y Clasificación	Se requerirá la implementación de visión artificial a través de una cámara web y el desarrollo de un algoritmo para el registro y procesamiento de imagen, para posteriormente clasificar el producto según parámetros establecidos.
Comunicación	Se realizará la comunicación entre el algoritmo clasificador y los actuadores controlados por la placa Arduino, y se implementará una interfaz que contribuya a la comunicación hombre-máquina.

Elaborado por: Autores

2.1. Diseño y construcción del prototipo

Considerando que el presente trabajo se centra en la implementación de un sistema de visión artificial para el control de calidad automatizado, se realizará un prototipo que simule el funcionamiento de una banda transportadora que tenga la suficiente robustez para mantenerse firme mientras se realicen pruebas de funcionamiento con productos comerciales comunes. Las dimensiones propuestas para dicho prototipo son de 1 metro de largo y 35 centímetros de ancho aprovechable de banda, esto con el fin de poder ubicarla en casi cualquier sitio para realizar pruebas de calibración y mostrar su funcionamiento. Para poder tener una idea clara de las dimensiones de todos los elementos que conformarán el prototipo, en un inicio se lo diseñó y modeló en un software CAD (SolidWorks) para de esta manera dimensionar cada parte de manera tal que sea funcional y estéticamente atractivo, en la Figura 1 se muestra el modelado final, donde se puede apreciar a breves rasgos las partes que lo conforman y que posteriormente será detallados.

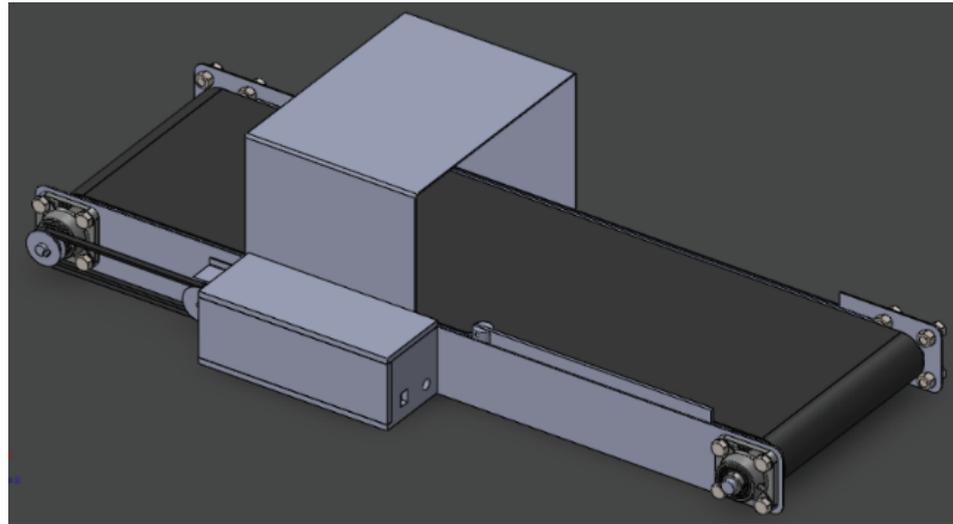


Figure 2

Modelado 3D del prototipo.

2.1.1. Elementos estructurales

La mesa, los laterales de soporte y la cubierta para la cámara web están fabricados con planchas de acero de 2 milímetros de espesor (en el caso de la cubierta es de 1.1 milímetros de espesor) para brindar robustez al prototipo a la vez que se lo hace relativamente ligero para fácil transporte. Estas planchas luego de haber sido cortadas, dobladas y perforadas fueron recubiertas por dos capas de pintura para evitar la corrosión y brindarle un acabado atractivo. En este soporte estructural se montan el resto de elementos como las chumaceras, mismas que están colocadas en los cuatro extremos para soportar los rodillos que tensorán y darán movimiento a la cinta transportadora, estas chumaceras son de $\frac{3}{4}$ de pulgada de diámetro interno. Los mencionados rodillos son fabricados en acero y mecanizados en un torno convencional para brindar un acabado superficial fino, ya que sobre uno de ellos se deslizará la cinta; el otro rodillo es recubierto, por proceso de vulcanizado, por caucho, con el fin de generar el arrastre necesario para transportar varios tipos de productos comerciales; los rodillos tienen un diámetro externo final de 2.5 pulgadas y sus extremos son ejes de $\frac{3}{4}$ de pulgada para que se acoplen perfectamente con las chumaceras elegidas. La cinta soportada por los rodillos es de PVC con 3 milímetros de espesor y denominada “sin fin”, ya que en la unión tiene un proceso de vulcanizado para que no interfiera en el transporte de productos. Esta selección de materiales y elementos contribuyen a que el prototipo sea funcional, ligero y robusto, para ser usado bajo condiciones controladas, es decir evitando temperaturas altas y sobrecargar la banda transportadora. A partir del



modelado inicial se extraen algunos parámetros iniciales, representados en la Tabla 2, para calcular un estimado de la carga máxima que podrá transportar el prototipo.

Table 2

Parámetros iniciales para diseño de banda transportadora.

Parámetro	Valor inicial	Unidad
Longitud de banda (de rodillo a rodillo)	1	m
Ancho de banda	30	cm
Velocidad de avance de la banda	0.4	m/s
Ángulo de inclinación	0	grados

Elaborado por: Autores.

Con estos parámetros iniciales y haciendo uso de las ecuaciones y constantes que presentan Valencia y Aymara (2011), se puede calcular la potencia requerida para que la banda tenga un funcionamiento continuo:

$$Q = (3.6 * p * v) + D$$

Ecuación 1: Capacidad real de transporte de una banda transportadora.

Donde:

Q : Capacidad real de transporte.

v : Velocidad de avance de la banda (m/s)

p : Peso por metro de la banda más rodillos en kg, que según bibliografía es igual a 1.5kg

D : Capacidad horaria de la cinta transportadora en t/h, que según bibliografía para la banda seleccionada es igual a 3 t/h.

$$P = \frac{C * R * L * Q * \cos(\theta)}{270}$$

Ecuación 2: Potencia requerida del motor.

Donde:

C : Coeficiente de compensación por suciedad en rodillos y cinta, según bibliografía para una longitud entre rodillos de 1m es igual a 13.

R : Coeficiente de rozamiento de rodillo, que, según bibliografía, es igual a 0.02 para el material seleccionado de banda.

L : Longitud de la banda transportadora (m).

θ : Ángulo de inclinación de la banda transportadora.



2.1.2. Sistema de transmisión y clasificador

A partir de las ecuaciones anteriores se obtuvo la potencia requerida calculada de 0.005 HP (3.73W), por lo que se puede proceder a elegir un motor de bajas revoluciones (76 RPM) y alto torque (45 kg*cm), teniendo una potencia aproximada de 0.05 HP (35.12W), motor que cumplirá con los requerimientos del prototipo sin entrar en sobrecarga o recalentarse. El motor seleccionado funciona con una fuente de alimentación de 12 voltios y 5 amperes (relación de transmisión 131:1).

Al ser un prototipo de exigencias limitadas no se puede hacer un diseño analítico y preciso del sistema de transmisión por banda y poleas [18]. por lo que se diseñará en base a la disponibilidad de mercado y a las dimensiones preestablecidas en la sección anterior, de esta manera se parte de una correa HM-24 (perfil en V) y se puede asegurar que resistirá y funcionará correctamente, ya que este elemento es diseñado para soportar torques más elevados en temperaturas exigentes [19]. En base a esta banda se diseñan y construyen las poleas, una de ellas, fabricada en acero, irá acoplada al motor seleccionado y será ajustada al mismo con ayuda de un prisionero; la otra polea, fabricada en nylon industrial, irá montada en un extremo del rodillo motriz (vulcanizado) fijada con una chaveta de acero para asegurar que no exista deslizamiento.

El sistema cuenta con un mecanismo de clasificación, el cual es capaz de separar a los productos de mala calidad. Este sistema es conformado por un servomotor con un torque de 11 kg*cm (1.079 n*m), el cual da movimiento a un brazo fabricado en nylon industrial, siendo lo suficientemente ligero para moverlo por encima de la cinta transportadora, pero a la vez es capaz de empujar y desviar los productos sin presentar una flexión elevada que comprometa la integridad del sistema. Finalmente, el prototipo cuenta con una caja de control, que, si bien no presenta botones físicos para controlar el prototipo, es donde se ubicarán todos los elementos electrónicos que alimenten, controlen y reciban información de los actuadores y sensor óptico.

2.1.3. Algoritmo e interfaz de usuario

El algoritmo es escrito en lenguaje Python con la ayuda de un procesador de texto, en esta ocasión se utiliza el software Spyder. Para que en el desarrollo del algoritmo se pueda incluir el procesamiento de imagen, el desarrollo de una interfaz, operaciones con imágenes y que se capturen, visualicen y guarden imágenes es necesario la importación de algunas librerías que en ciertos casos deben ser instaladas manualmente antes de incluirlas en el código de programación, en la Tabla 3 se muestran todas las librerías necesarias, así como las funciones principales que se utilizaron para



Figure 3

Prototipo ensamblado y listo para implementación de algoritmo.

dar estructura al algoritmo clasificador. Una vez detalladas las librerías que se incluirán en el algoritmo, en la Figura 3 se muestra el diagrama de flujo que rige el procesamiento de imagen.

Para la puesta en marcha del equipo clasificador no basta con que se automatice todo el sistema o que funcione correctamente todos los elementos que lo integran, sino también se requiere que cuente con una que sea fácil de entender y manejar, sin la necesidad de planificar una capacitación adicional para el manejo del sistema. Esta interfaz es diseñada y programada en el mismo editor de texto de Python, Spyder, y cuenta con cuatro pestañas para el mejor desenvolvimiento del operario que se detallan en la Tabla 4.

3. Resultados

Una vez puesto en marcha el sistema clasificador, mediante la interfaz, es importante señalar que se debe esperar un lapso de tiempo (menos de un segundo) para que la cámara pueda iniciarse de manera correcta y registrar el entorno donde está siendo utilizada, ya que posteriormente utilizará este fondo para ubicar y reconocer el producto a analizarse. Al inicio del sistema es necesario contar con un producto considerado como de alta calidad para capturar una imagen del mismo y utilizarla como guía para que el algoritmo sea capaz de diferenciar entre los productos que debe aceptar o rechazar. Como ya se detalló anteriormente la función de cada pestaña dentro de la



Table 3

Comandos básicos para el desarrollo del algoritmo clasificador.

Proceso	Librerías	Funciones	Aplicación
Interfaz	tkinter	interfaz=Tk() interfaz.geometry() interfaz.title() interfaz.resizable() interfaz.iconbitmap()	Inicialización de interfaz, configuración de tamaño, título, tamaño e ícono de ventana.
		pestaña=tk.Frame(pestañas) pestañas.add(pestaña,text="INICIO")	Creación y registro de pestañas, colocando orden y título de cada pestaña.
		frame1=Frame() frame1.config() frame1.place()	Creación de <i>frames</i> o secciones dentro de pestañas, configuración de nombre, tamaño, color, bordes y posición dentro de ventana.
		Label1=Label() Label1.config() Label1.place()	Creación de etiquetas o cuadros de diálogo dentro de <i>frames</i> , configuración de nombre, tamaño, color, bordes y posición dentro de ventana.
		Boton=Button() Boton.config() Boton.place() Boton.config()	Creación de botones dentro de ventana o <i>frame</i> , configuración de nombre, tamaño, color, bordes, posición y tipo de cursor.
Iniciación de Cámara	cv2	cap1 = cv2.VideoCapture(0, cv2.CAP_DSHOW) cap1.release()	Inicialización la cámara, el número "0" dependerá de qué cámara se desee activar.
Captura y Almacenamiento de Imagen	cv2	ret, frame=cap.read() cv2.imwrite("Name",frame) cap.release()	Captura el fotograma registrado por la cámara y pausa la función de captura constante. Luego lo guarda en el almacenamiento interno del algoritmo.
Lectura de Imagen	cv2 imutils	Etiqueta=cv2.imread("Direccion") Eti- queta=imutils.resize(Etiqueta, height=325) Etiqueta=cv2.cvtColor(Etiqueta, cv2.COLOR_BGR2RGB) im=Image.fromarray(Etiqueta) img=ImageTk.PhotoImage(image=im)	Permite leer la imagen almacenada anteriormente, reajustar su tamaño y fijarla en un <i>frame</i> .
Operación y Procesamiento de Imagen	ORB numpy	orb=cv2.ORB_create() bf=cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True) kp1=orb.detect(Etiqueta,None) kp1, des1 = orb.compute(Etiqueta, kp1) gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) dif=cv2.absdiff(gray, bgGray) _,th=cv2.threshold(dif,40,255, cv2.THRESH_BINARY) cnts = cv2.findContours(th, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)[0]	Las líneas de código presentadas permiten detectar y registrar las características especiales de imágenes (guía y procesada) para posteriormente almacenarlas en dos variables y compararlas para encontrar que tan semejantes son dichas imágenes.
Visualización de Imagen	PIL imutils	ret, frame = cap.read() frame = imutils.resize() frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) im = Image.fromarray(frame) img=ImageTk.PhotoImage(image=im) Etiq=Label() Etiq.configure(image=img) Etiq.config() Etiq.place() Etiq.image = img Etiq.after(10, visualizar)	Muestra la imagen registrada por la cámara, iniciada anteriormente, en un <i>frame</i> , y al ser un proceso continuo muestra constantemente los fotogramas capturados para dar la ilusión de registro constante.

Elaborado por: Autores

interfaz en los párrafos siguientes se detallará como se desenvuelve el sistema y las posibles limitaciones que puede presentar.

Antes de dar por iniciado el proceso de clasificación se recomienda hacer una prueba de funcionamiento de los periféricos, con el fin de asegurar que todos están conectados

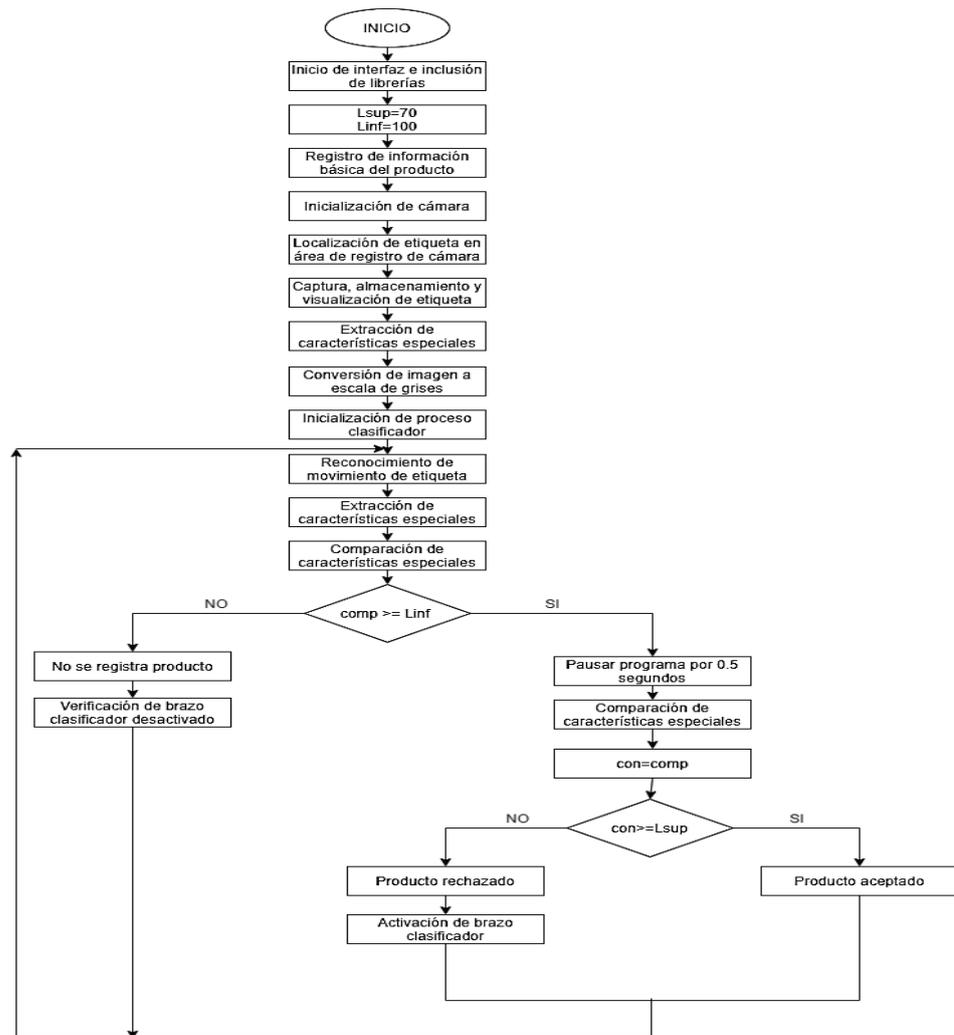


Figure 4

Algoritmo de procesamiento de imagen y clasificación.

correctamente y ejecutan las funciones que se espera y evitar contratiempos durante el proceso. Para la ejecución de estas pruebas cabe recalcar que el sistema debe estar parado ya que si no lo está el programa mostrará un mensaje de error, mismo que seguirá apareciendo hasta que no se detenga el proceso. En la Tabla 5 se puede visualizar los productos analizados, comparándolos entre aceptados y rechazados (los rechazados son representados por productos con su etiqueta parcialmente cubierta), de igual manera se muestra el valor que toman las dos variables principales, encargadas de reconocer y clasificar el producto.

Los valores principales detallados en la Tabla 5 (Lsup y Linf) fueron obtenidos de manera experimental ejecutando pruebas de funcionamiento de manera continua, en estas pruebas se procedió a imprimir en consola (Spyder) los valores que tomaba la variable que almacena el número de semejanzas que presentan la imagen guía



Pestaña	Función	Interfaz
Inicio	Registrar información del producto a analizarse. Activar la cámara para localizar y capturar la etiqueta guía. Estado del sistema. Inicializar proceso de verificación y clasificación.	
Monitoreo	No se puede realizar ninguna acción (solo visualización). Muestra la imagen capturada en la ventana Inicio. Muestra la imagen que capta la cámara en vivo.	
Registro y Parada	Pestaña de resumen. Información del producto y cuántos productos han sido analizados en total. Parar el proceso de clasificación o de salir y finalizar programa.	
Pruebas	Realizar pruebas de funcionamiento de todos los periféricos (cámara, banda transportadora y brazo clasificador)	

Elaborado por: Autores.

Figure 5

Detalle de acciones dentro de cada pestaña de la interfaz diseñada.

capturada en un inicio con el producto que estaba siendo analizado. Estos valores son registrados en la Tabla 6 y representan las semejanzas existentes (de forma resumida) desde que el producto entra en el rango de visión de la cámara web, de esta manera se pudo tomar un valor referencial (Linf) para que el algoritmo se detenga por medio segundo, y así, el producto completo se encuentre dentro del rango de captura, y vuelva a realizar el proceso de comparación para mostrar un nuevo valor (Lsup) que sirve para tomar la decisión de separar o mantener el producto dentro de la línea de producción, este valor dependerá del producto que se analice y por esa razón es recomendable realizar una calibración previa.

Las columnas A y R representan al análisis de productos considerados aceptados y rechazados, y los subíndices 1 y 2 corresponden al análisis de producto en la posición inicial e invertida, respectivamente.

Al ser una interfaz intuitiva, el usuario, con solo vigilar la pantalla del ordenador, podrá entender el proceso que realiza el algoritmo y constatar que es capaz de reconocer los productos de alta calidad y separar de la línea de producción a aquellos que



Figure 6

Interfaz del sistema clasificador autónomo.

Nº	Tipo de producto	Aceptadao	Rechazado
1	Caja pequeña Lsup=70 Linf=100		
2	Empaque plástico Lsup=70 Linf=100		
3	Enlatados Lsup=70 Linf=100		
4	Caja mediana Lsup=50 Linf=100		

Figure 7

Interfaz del sistema clasificador autónomo.

presentan fallas en su etiquetado. De igual manera, en la pestaña Registro y Parada, el operario puede actualizar el conteo de productos aceptados y rechazados de manera



Table 4

Resumen de datos adquiridos durante pruebas de número de coincidencias.

N°	Reconocimiento (Linf)								Lectura luego de 0.5s (Lsup)																	
	A ₁			R ₁			A ₂		R ₂			A ₁			R ₁		A ₂		R ₂							
1 (Caja pequeña)	44	68	71	34	34	36	67	75	80	43	34	27	91	91	115	61	59	46	96	100	112	67	66	48		
	59	85	109	43	44	59	88	107	117	41	54	62	122	154		40	24		127	140		47	32			
	118		133	67	77	104	110		129	86		100														
	130						125			108																
2 (Empaque plástico)	26	41	39	13	18	13	16	23	25	45	9	17	17	27	77	98	107	64	47	31	87	91	99	57	49	51
	36	41	75	22	25	59	65	68	102	42	60	84	113	118		16	12		116	118		37	23			
	88	105	118	75	102		107	112	113	102	110															
3 (Enlatado)	44	49	62	22	36	46	59	88	89	29	40	39	74	126	105	64	81	61	84	118	102	62	63	60		
	75	106	119	55	61	61	102		104	63	77	92	99	62		42	28		71	69		37	15			
	107		125	98	95	101	118		107	97	98	106														
	109						116	120																		
4 (Caja mediana)	34	46	38	57	56	76	33	32	38	43	45	63	59	75	50	33	33	26	65	81	58	33	42	20		
	43	50	78	72	75	68	41	45	48	72	69	87	67	62		15	7		50	52		10	8			
	88		123	69	100	101	64	89	122	84	92	105														
	125																									

Elaborado por: Autores.

constante para poder llevar un control de cuantos productos tienen defectos dentro de un lote específico. Cabe señalar que de igual manera el sistema puede ser detenido en cualquier momento, sea para calibrarlo, realizar el proceso manual o simplemente porque se pausó la producción, esta detención del proceso se lo controla con una ventana emergente con el fin de que no sea parado por equivocación y el proceso pueda continuar sin la pérdida de datos.

El algoritmo puede ser adaptado a una línea de procesos industriales, al ser de código abierto se lo puede modificar según los requerimientos y realidad donde se desee hacer uso del mismo. De esta manera, cualquier empresa o industria que adquiera el algoritmo, con ayuda del programador del mismo, podrá modificar pocas líneas de código para poder realizar las pruebas pertinentes con el fin de recopilar los datos resumidos en la Tabla 6. Como se puede observar la calibración consiste en modificar dos variables, Linf y Lsup, son estas dos variables las responsables de decidir si un producto es considerado como aceptado o rechazado.

Una de las limitantes principales del sistema es la velocidad de procesamiento, en promedio el sistema y equipo planteado en el presente trabajo puede procesar alrededor de 600 productos por hora, lo que al compararlo con líneas de producción en serie (hasta 2000 productos por hora [20]) es muy reducido, sin embargo, cumple con los objetivos planteados al inicio del trabajo, mostrando un algoritmo robusto capaz de procesar y clasificar una gran variedad de productos sin la necesidad de realizar



cambios sustanciales en la estructura de sus líneas programadas y escritas en lenguaje de código abierto (Python).

4. Conclusiones

La librería ORB es un excelente sustituto de las librerías SIFT y SURF (mismas que no pueden ser utilizadas de forma libre por lo que es imposible realizar comparaciones cuantitativas), ya que logra reconocer las características principales de las imágenes capturadas por la cámara de manera eficiente y con un margen de error muy reducido, además que tiene la ventaja de que es de uso libre.

El factor lumínico juega un papel crucial en el correcto reconocimiento de imagen, ya que luego de realizar las pruebas de funcionamiento se observó que, antes de colocar las luces LED, al tener un nivel de luz bajo se lograba registrar entre 150 y 250 características especiales; pero, al colocar las luces LED o al tener un nivel de luz indirecta alto, se pudo registrar hasta 400 características especiales, esto hace que se pueda tener mayor puntos de comparación para lograr una clasificación más refinada y precisa.

La implementación del algoritmo al prototipo diseñado y construido en el desarrollo del proyecto, contribuyó al entendimiento del funcionamiento del clasificador automático, el cual, gracias a su estructura interna, puede ser calibrado y modificado de tal manera que permite adaptarlo a cualquier circunstancia o aplicación, analizando cualquier tipo de productos sin la necesidad de realizar inversiones extras o de cambiar en su totalidad al código. De esta manera se brinda un sistema autónomo que podrá cumplir las funciones de un humano con menos riesgos.

References

- [1] Garcés AF. Sistema de reconocimiento facial con visión artificial para apoyar al ECU-911 con la identificación de personas en la lista de los más buscados. Ecuador: Universidad Técnica de Ambato; 2017.
- [2] InfoPLC. Crecimiento del mercado de visión artificial con 3D a la cabeza. Spain: InfoPLC; 2018. Available from: <https://www.infopl.net/plus-plus/mercado/noticias-mercado/item/104695-crecimiento-del-mercado-de-vision-artificial-con-3d-a-la-cabeza>.
- [3] Gómez C, Mey C, Martínez C, Martín A. La inteligencia artificial al servicio del bien social en América Latina y el Caribe: Panorámica regional e instantáneas de



- doce países. New York, NY: Banco Interamericano de Desarrollo (Grupo BID); 2018. p. 1–144. Available from: <https://publications.iadb.org/es/la-inteligencia-artificial-al-servicio-del-bien-social-en-america-latina-y-el-caribe-panoramica>
- [4] Gómez De León FC, González A. Inspección de materiales. Universidad de Murcia. Murcia. 2010.
- [5] Aldás DS, Collantes SM, Reyes JP. Procesamiento de imágenes con visión artificial para el reconocimiento de placas vehiculares. SATHIRI. 2016;11:133.
- [6] Ponce J. Manual de Insp. Visual. 2012 [Consulta: 21 mayo 2021]. Available from: https://www.academia.edu/16541538/MANUAL_DE_INSP_VISUAL
- [7] Platero Dueñas C. Inspección automatizada de superficies homogéneas mediante visión artificial con aportaciones al reconocimiento de formas [dissertation]. Spain: Higher Technical School of Industrial Engineering (UPM); 1998.
- [8] ClearView Imaging. Inspección de etiquetado [Internet]. 2018 [Consulta: 24 mayo 2021]. Available from: [https://cdn2.hubspot.net/hubfs/3780277/WebsiteAssets/pdf/SpanishPDF/Label Inspection-Inspección de etiquetado.pdf](https://cdn2.hubspot.net/hubfs/3780277/WebsiteAssets/pdf/SpanishPDF/Label%20Inspection-Inspección%20de%20etiquetado.pdf)
- [9] Ramos MA. Introducción a los sistemas de tiempo real. Universidad de Buenos Aires. Buenos Aires, Argentina. 2015.
- [10] Úbeda B. Introducción a los sistemas embebidos. Universidad de Murcia. Murcia, España. 2009.
- [11] Edutin Academy. Curso de Python [Internet]. USA: Edutin Academy; 2019 [Consulta: 3 junio 2021]. Available from: <https://app.edutin.com/>
- [12] Amaya I, Gomez K, Horst E, Losoyo OY, Reyes A. Visión automatizada. Centro de Investigación científica y de Educación Superior Ensenada. Ensenada, Baja California. México. 2016.
- [13] Rodriguez L. Python programación 3. Guayaquil: Escuela Superior Politécnica del Litoral; 2017.
- [14] SafetyCulture. Inspección visual: Plantilla gratis | iAuditor by SafetyCulture [en línea]. Australia: SafetyCulture; 2020 [Consulta: 21 mayo 2021]. Available from: <https://safetyculture.com/es/listas-de-verificacion/inspeccion-visual/>
- [15] González de Leon F. Vision artificial con Python y OpenCV. Madrid, Majadahonda. España. 2015. Video: 23:00 min. Available from: <https://www.youtube.com/watch?v=5Z4VZhHZk58>
- [16] Gonzalez L. Introducción al IDE Spyder [Internet]. AprendeIA; 2018 [Consulta: 3 junio 2021]. Video: 5:18 min. Available from: <https://aprendeia.com/ide-spyder-para-python/>



- [17] Vásquez CY, Cevallos J. Inteligencia artificial - Visión artificial. Universidad Técnica del Norte. Ibarra. Ecuador. 2013.
- [18] García S. I, Caranqui S. V. La visión artificial y los campos de aplicación. Tierra Infinita [Internet]. 2015;1:94–103. Available from: <http://revistasdigitales.upec.edu.ec/index.php/tierrainfinita/article/view/76>
- [19] Vélez JF, Moreno AB, Sánchez Á. Y, Sánchez JL. Visión por Computador. 2003.
- [20] Kulhary R. OpenCV - Overview [Internet]. Noida, India: GeeksforGeeks; 2019 [Consulta: 3 junio 2021]. Available from: <https://www.geeksforgeeks.org/opencv-overview/>
- [21] SKF. SKF power transmission belts contents. SKF Group; 2012.
- [22] DRB. Industrial belt [Internet]. DRB [Internet]; 2012 [Consulta: 1 agosto 2021]. Available from: http://www.drbworld.com/include/pdf_viewer.php?file={%}2FDATAS{%}2Fes_free9{%}2F{%}5BPower_Transmission_Belt{%}5D_Industrial_Belt.pdf
- [23] Winchell W. Inspection and measurement in manufacturing [Internet]. Michigan: Society of Manufacturing Engineers; 1996 [Consulta: 24 mayo 2021]. Available from: https://books.google.com.ec/books?id=be9ZWBfMpTgC&printsec=frontcover&dq=inauthor:{%}22William+Winchell{%}22&hl=es&sa=X&redir_esc=y#v=onepage&q&f=false
- [24] CAF. Las grandes oportunidades de la Inteligencia Artificial en el sector público de América Latina. CAF [Internet]; 2018 [Consulta: 1 diciembre 2021]. Available from: <https://www.caf.com/es/actualidad/noticias/2021/09/las-grandes-oportunidades-de-la-inteligencia-artificial-en-el-sector-publico-de-america-latina/>
- [25] Samaniego H. Diseño y construcción de un módulo de monitoreo y control de un sistema de calefacción a través de la internet. Quito: Escuela Politécnica Nacional; 2013.
- [26] Valencia GY, Amaya L. Control de calidad de un proceso de etiquetado empleando el robot industrial GRYPHON. Colombia: Universidad Pontificia Bolivariana; 2011.