

Detección y extracción de muestras falladas usando visión artificial y un brazo robótico.

Detection and Extraction of faulty samples using artificial vision and a robotic arm.

Ryan León*; Esteban Alvarado; Kevin Arevalo; Astolfo Maldonado; Antonella Polonio

Facultad de Ingeniería – Ingeniería Industrial, Universidad Privada del Norte, Mz. G Lt. 24 Urb. Dean Saavedra, El Cortijo, San Isidro - Trujillo, Perú.

*Autor correspondiente: ryan.leon@upn.edu.pe (R. León)

RESUMEN

Se realizó un trabajo de investigación con la finalidad de detectar y extraer muestras falladas usando visión artificial y un brazo robótico con el supuesto que dicha actividad logre una disminución de los costos y mejora en la optimización de procesos en el control de la calidad, mediante el reconocimiento de patrones RGB. El estudio tuvo carácter experimental y analítico. En el desarrollo del procesamiento se hizo uso del algoritmo matemático CBIR y la teoría RGB. El prototipo trabaja con un total de 6 tipos de imágenes, se realizaron 10 pruebas por imagen. El brazo robótico se tardó en promedio 0,33 segundos, para la identificación del error, mientras que el ojo humano promedio se tardó 17 segundos. También se corroboró lo que distintos autores como Sandoval, Silva. L. han realizado y recomendado en investigaciones similares; sin embargo, algo que no se ha podido concluir satisfactoriamente en las otras investigaciones es la variación de los colores, lo cual se mejoró en nuestro proyecto haciendo uso del teorema RGB. Asimismo, se utilizó un identificador bayesiano, como herramienta estadística para identificar características discriminantes con un error porcentual menor a 5%. Se determinó que el proyecto trabaja de manera óptima logrando una certeza de 97,5% en el primer nivel, de 96,41% en el segundo nivel y de 95,35% en el tercer nivel.

Palabras clave: visión artificial; algoritmo CBIR; teoría RGB

ABSTRACT

A research work was carried out with the determination to detect and extract failed samples using artificial vision and a robotic arm with the assumption that said activity achieves a decrease in costs and improves the optimization of processes in quality control, through RGB pattern recognition. The study was experimental and analytical. In the development of the processing, the mathematical algorithm CBIR and the RGB theory were used. The prototype works with a total of 6 types of images, 10 tests were performed per image. The robotic arm took an average of 0.33 seconds to identify the error, while the average human eye took 17 seconds. The various authors such as Sandoval, Silva. L. have also carried out and recommended similar investigations; however, something that has not been successfully concluded in the other investigations is the variation of the colors, which was improved in our project using the RGB theorem. Likewise, a Bayesian identifier was identified as a statistical tool to identify discriminant characteristics with a percentage error of less than 5%. It was determined that the project works optimally achieving a certainty of 97.5% in the first level, 96.41% in the second level and 95.35% in the third level.

Keywords: artificial vision; CBIR algorithm; RGB theory

1. INTRODUCCIÓN

Desde hace varias décadas el hombre se ha empeñado en automatizar las tareas que requieran un comportamiento inteligente, para lo cual ha desarrollado máquinas que imitan rasgos de la inteligencia humana. Con este fin ha creado técnicas basadas en el análisis formal y estadístico del comportamiento humano ante diferentes problemas, aliviando así áreas tales como el control de sistemas, la planificación automática, la habilidad de responder a diagnósticos y a consultas de los consumidores, el reconocimiento de la escritura, el

habla y los patrones. En los últimos años ha adquirido particular relevancia el empleo de algoritmos de visión artificial basados en técnicas de reconocimiento de patrones, para incrementar las prestaciones de manipuladores robóticos en muchos escenarios, destacándose la clasificación automática de productos agrícolas, basada en criterios de forma, textura, tamaño y color, en líneas de embalaje de los mismos. El procesamiento digital de imágenes es una de las ramas de la ciencia que más desarrollo ha experimentado en la última década, debido fundamentalmente a dos factores: el avance vertiginoso que ha experimentado el hardware, y la importancia que han adquirido las aplicaciones de esta técnica en la vida moderna Gonzales, Martínez y Rodríguez, 2011 asimismo el procesamiento digital de imágenes se centra en 3 etapas consecutivas como las son umbralización, limpieza y filtrado; según los autores cada una de estas desarrolla una serie de operaciones que permiten obtener la información requerida utilizando fotos RGB (Multicapa de 3 colores) Berrocal, Löfgren y Lundgren, 2016.

Los países de América Latina presentan grandes diferencias tecnológicas con los países desarrollados, hecho que se evidencia en la capacidad de los primeros para absorber, imitar, adaptar e innovar tecnologías que puedan ser empleadas en sus diferentes procesos productivos. De esta manera, es fundamental conocer el estado tecnológico de las diferentes industrias de la región para facilitar el planteamiento de proyectos de investigación que faciliten el mejoramiento de los procesos productivos, y aprovechar las oportunidades que brindan la ciencia la tecnología y la innovación Ruge, Pérez, 2017. Hoy en día muchas de las empresas del Perú dedicadas a producción, envasado y exportación de diversos productos buscan optimizar sus procesos y disminuir tiempos al momento de realizar una operación y poder así lograr sus metas de producción. Según Sáenz, 2016 plantea el desarrollo de dos algoritmos de procesamiento de imágenes para la identificación del café idóneo para producción, uno de acuerdo al color de este (estado de maduración) y otro para detectar la plaga de la broca para lo cual los resultados mostrados por la curva ROC indican que la red neuronal de base radial se desempeña de manera óptima como clasificador para detectar la maduración del café. Además, usando un algoritmo con bajo costo computacional y simple de implementar y fue posible detectar adecuadamente un orificio dejado por una broca sobre un fruto de café. Asimismo Boudy, Leon y Estrada, 2011 formula y señala que para la identificación inteligente de objetos mediante reconocimiento de patrones utilizando redes neuronales artificiales funciona de manera satisfactoria, identificando los tipos de objetos predeterminados con un alto porcentaje de certeza, y fue capaz de determinar la posición de los mismos dentro del área de trabajo del robot, sin embargo si varía la iluminación del área de trabajo por lo cual se requiere reajustar manualmente el umbral de filtrado de la imagen, con la implementación del método de Otsu que, al trabajar la imagen de forma local, es menos ruidoso; y que al estar basado en el análisis discriminante de la misma, permite tener un umbral autoajutable que dotaría al sistema de un mayor grado de independencia en lo relativo a la iluminación del entorno de trabajo del robot.

La propuesta de solución del trabajo se basa en el procesamiento de imagen y parte desde un algoritmo de reconocimiento de errores en el cual el sistema tiene que identificar cuando el producto está defectuoso, mediante el brazo robot se podrá apartar el producto defectuoso una vez detectado. Para lo cual se realizó pruebas a manera de un juego de competencia en la que se compara la agilidad mental del ser humano con la máquina. Con este proyecto se busca reemplazar el ojo humano por visión artificial ya que hoy en día muchas de las empresas del Perú dedicadas a producción, envasado y exportación de diversos productos buscan optimizar sus procesos y disminuir tiempos al momento de realizar una operación y poder así lograr sus metas de producción. El proceso de control de calidad es fundamental en todo proceso industrial que implique obtener un producto terminado de acuerdo a los estándares de calidad; por lo cual, hemos desarrollado un proyecto orientado a la optimización de dicho proceso que además tienen varias ventajas entre ellas: minimizar costos, reducir riesgos para obreros, disminuir tiempos de operación, producción en serie etc.

2. MATERIALES Y MÉTODOS

Para realización del prototipo se usaron los siguientes materiales:

Tabla 1. Materiales que se usaron para la elaboración del proyecto

Materiales	Cantidad	Precio Soles
Arduino Mega	1	80
Brazo robot de metal de 6 grados de libertad ^a	1	800
Cámara web	1	80
Servomotor de alto torque	1	120
Servomotor de bajo torque	4	140
Regulador de voltaje step Down	1	55
Garra metálica	1	25

Materiales	Cantidad	Precio Soles
Cables DuPont	1	30
Fuente ATX	1	55
Tiras de led de 12 v	2	20
Cable de poder	1	10
Cartón maqueta negro	3	30
Una plancha de melamina	1	50
Pegamentos	1	12
Tornillos	30	13
Total		1,520

Nota. Todos los dispositivos son fáciles de conseguir. El precio es muy cómodo.

^a Brazo robot de metal de 6 grados de libertad necesario tener en cuenta los 6 grados de libertad para un adecuado trabajo.

Para funcionamiento de nuestro brazo robótico que hace uso de la visión artificial, se debe tener en cuenta las especificaciones del entorno ya que esto ayudara a la sensibilidad de nuestra cámara web para la detección de objetos. El brazo se debe establecer en un entorno iluminado para mejorar la detección de indicadores físicos y ópticos de nuestras piezas.

Imagen digital.

La imagen digital está formada por una serie de matrices numéricas de ceros y unos que se almacenan en una memoria informática y que definen las características de una fotografía.

La trayectoria que sigue la cámara para formar la imagen digital es la siguiente forma: La luz que detecta el objetivo de la cámara llega hasta el sensor de imagen, denominado CCD formado por multitudes de receptores fotosensibles denominados "fotodiodos".

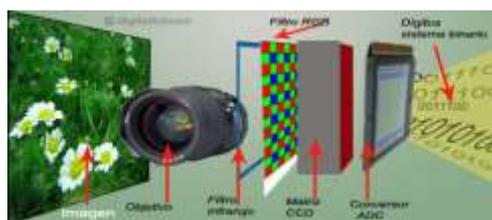


Figura1: Composición de una imagen digital.

El sistema RGB es el más común en sistemas digitales y se usa mucho en el manejo del color por código. Simplemente se trata de especificar la intensidad con la que cada canal está encendido, cada pixel o led (en caso de leds para pantallas de gran formato o instalaciones) puede “encenderse” en los tres canales básicos rojo (r), verde (g) o azul (b). Cada canal puede ir de completamente apagado a completamente encendido, cuando todos están apagados el pixel es negro y cuando todos están encendidos el pixel es blanco.

(Hernández-Castro, 2016) Una imagen digital en color se crea con los parámetros en R G B, por la famosa síntesis aditiva, el color rojo, verde y azul. Ver síntesis aditiva:

Si anteriormente necesitábamos 8 bits para captar una imagen de 256 tonos de un solo color, ahora precisamos 8 bytes, es decir 24 bits:

- * 8 bits de color rojo.
- * 8 bits de color verde.
- * 8 bits de color azul.

Teniendo en cuenta el patrón RGB, se utilizó por la razón de que el color es un perfecto descriptor, el cual nos ayuda a reconocer, excluir y extraer objetos de una imagen. Veamos la utilización y las características en la siguiente Figura 2 del código de visión artificial realizado en Matlab.

```
function FORMATODropDownValueChanged(app, event)
    value = app.FORMATODropDown.Value;
    app.vid =
    videoinput('winvideo', app.CMARADropDown.Value, value);
    app.ListoButton.Enable = 'on';
    app.vid.ReturnedColorspace = 'rgb';
    app.VistadepruebaButton.Enable = 'on';
    app.imgObjFlag=1;
end
```

Figura 2: RGB

Procesamiento de Imagen: Una vez obtenida la imagen y ordenada en matrices, es posible aplicarle operaciones matemáticas. En este punto existen múltiples alternativas para establecer similitud entre imágenes así como algoritmos basados en sistemas SURF (speeded Up Robust Features), HOG (histogram Of Gradients), LBP (Local Binary Patterns).

El algoritmo CBIR denominado, sistemas de recuperación de imágenes basados en contenido es el utilizado como la técnica que permite reconocer la composición y el contenido de las imágenes, y tenemos que tener en cuenta que esta utiliza su propio contenido para conseguir la recuperación, consta de los siguientes pasos:

- Seleccionar las características de imagen a recuperar
- Crear el grupo de características
- Indexar las imágenes a comprobar
- Ordenar las imágenes dependiendo del número de características que cumplen.
-

Por lo tanto, para realizar nuestro entrenamiento y poder aplicar el algoritmo CBIR, tenemos que tener nuestra imagen ya en formato RGB, acondicionada y convertida. Para poder realizar la extracción de las características y poder construir el patrón de la imagen.

Mostraremos en la siguiente figura 3. parte del código, en donde se trabaja con la técnica CBIR aplicada a la figura Avdangers nivel de dificultad 2, el código esta desarrollado en el software Matlab.

```
function AvengersSwitchValueChanged(app, event)
    value = app.AvengersSwitch.Value;
    if strcmp('On', value)
        app.OjosSwitch.Value='Off';
        app.EstrellasSwitch.Value='Off';
        app.sSwitch.Value='Off';
        app.NsSwitch.Value='Off';
        app.KmetricsSwitch.Value='Off';
        app.queryImage = imread([app.Dpath 'Fail
Avengers\fail.jpg']);
    end
end

function sSwitchValueChanged(app, event)
    value = app.sSwitch.Value;
    if strcmp('On', value)
        app.OjosSwitch.Value='Off';
        app.EstrellasSwitch.Value='Off';
        app.AvengersSwitch.Value='Off';
        app.NsSwitch.Value='Off';
        app.KmetricsSwitch.Value='Off';
        app.queryImage = imread([app.Dpath 'Fail 8s\fail.jpg']);
    end
end
```

Figura 3.Código con técnica CBIR

Cálculos matemáticos y eléctricos

En el área de trabajo se consideran los 8cm de la garra debido a que ésta deberá encontrarse en posición vertical sobre los cubos para poder extraerlos y colocarlos, pero sí se consideran para la ubicación del sensor de color, que estará sobre la fuente atx.

Una vez dimensionada la estructura es necesario conocer la capacidad de carga del brazo, para eso hacemos uso de las características del fabricante.

Torque MG96 Servo:| (1)

$$Torque_{5v} = \frac{4,8(v) \times 11(kg - cm)}{6(v)} = 8,8(kg - cm)$$

Servomotor utilizado para la garra y los dos grados de libertad que conforman la muñeca.

Torque MG946R SERVO

$$Torque_{5v} = \frac{4,8(v) \times 13(kg - cm)}{6(v)} = 10,4(kg - cm)$$

Mayor capacidad de carga dado que se utilizará para la articulación del codo (lo que implica mayor distancia desde el motor hasta la punta de la garra donde se encontrará el objeto a manipular) y el eje de rotación.

Utilizado para la articulación del hombro, es el servomotor más potente del sistema, ya que será el que realice mayor esfuerzo cuando el brazo se encuentre al borde del área de trabajo.

Todos los servomotores, trabajarán a 5v por lo que haciendo un cálculo con su torque a 6v:

$$Torque_{5v} = \frac{5(v) \times 17(kg - cm)}{6(v)} = 14,167(kg - cm)$$

Cálculo de potencia de los servomotores:

MG996: $8,8(kg - cm) = 0,8629 \text{ W.s}$

MG946: $10,4(kg - cm) = 1,0198 \text{ W.s}$

HD1501MG: $14,167(kg - cm) = 1,3893 \text{ W.s}$

A fin de simplificar los cálculos se considerará a esta sección como una barra uniforme de peso 0,653Kg (peso hallado experimentalmente) y longitud 30cm, ya que el máximo torque será experimentado cuando toda esta sección del brazo se encuentre extendida.m

Por lo que:

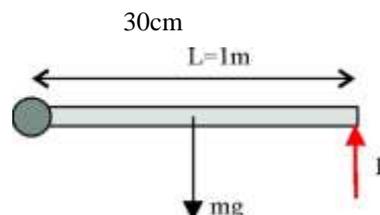


Figura 4. Gráfica para hallar el torque.

Considerando a la sección del brazo como una barra rígida uniforme:

$$Torque = 0,653Kg \times 15cm = 9,795(kg - cm)$$

Entonces, se cumple que el torque que proporciona el servomotor es mayor al torque requerido para mover el brazo sin carga, entonces se puede calcular el peso máximo que puede mover el brazo a 5v:

$$\begin{aligned} 9,795 + Torque_{max} &< 14,167 \times 75\% \\ Torque_{max} &< 0,83(kg - cm) \\ Peso_{max} &< \frac{0,83}{30} \\ Peso_{max} &< 0,027Kg \rightarrow 27gr \end{aligned}$$

*nota: se considera el 75% del torque máximo para evitar hacer que los servomotores trabajen a máxima capacidad

Para el cálculo de las rutinas es necesario calcular la velocidad angular de cada servomotor, necesaria para realizar sus correspondientes recorridos en el mismo tiempo, ya sea que mientras un grado de libertad debe moverse, por ejemplo, de 0° a 150° y otro de 45° a 60° por lo que si realizamos una gráfica posición angular vs tiempo de los 6 grados de libertad obtendremos lo siguiente:



Figura 5. Posición angular vs tiempo.

Por lo que, siendo la pendiente de cada recta la velocidad angular necesaria para que todos servomotores realicen el recorrido en el tiempo determinado 1,6 segundos para este caso, (puede variar en la aplicación del proyecto a base de prueba y error).

Cada una de estas pendientes se puede calcular haciendo uso de la ecuación de la recta que pasa por dos puntos:

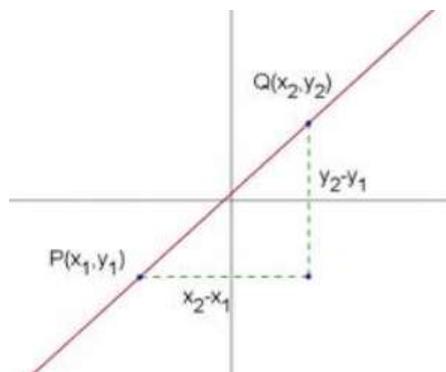


Figura 6. Gráfica de la pendiente.

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1)$$

Entonces, para cada recorrido, encontraremos la ecuación: Esta ecuación se ingresará al Arduino para detectar la posición angular de cada servomotor del brazo robótico

$$\theta = m \cdot t + \theta_i \tag{2}$$

Donde:

m: es la pendiente de la recta que representa la velocidad angular

t: tiempo en milisegundos

θ_i : el ángulo inicial

El controlador realizará el cálculo hallando el ángulo para cada servomotor y enviando esa orden una vez cada milisegundo, enviando en total 1,600 órdenes a cada servo a lo largo de todo su recorrido.

Tabla 2. Voltaje y corriente de los servomotores.

Componente	Voltaje(v)	Corriente(mA)
Servomotor MG996R	5	1,500
Servomotor MG996R	5	1,500
Servomotor MG996R	5	1,500
Servomotor MG946R	5	2,100
Servomotor MG946R	5	2,100
Servomotor HD1501 ^b	6,7	2,800
Arduino Mega	5	165
Total		11,66525

Nota. Corriente necesaria para saber el máximo de potencia a consumir.

^b Servomotor HD1501 utilizado en la base del brazo robot

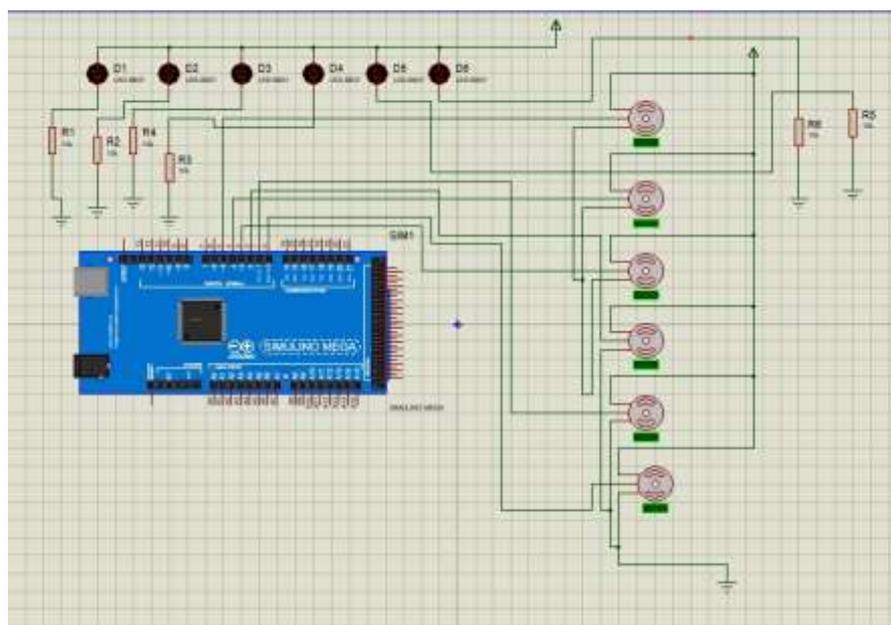


Figura 7. Interacción entre Arduino y servomotores.

El circuito de la figura 7, esta implementado como simulación para su implementación de forma real, realizado en el software Proteus, es utilizado para el control de los ángulos de los 6 servomotores, el Arduino Mega es el controlador donde va instalado el programa que va a mover cada grado de libertad del brazo robot, en este caso cada grado de libertad es un servomotor, el programa instalado tiene que enviar señales PWM las cuales tienen un ciclo de 20 ms, en donde podemos enviar diferentes señales para ir variando los ángulos dentro de ese ciclo, por ejemplo por los pines de salida del Arduino para un ángulo de 0 grados se envía una señal de 1 ms, para un ángulo de 90 grados se envía una señal de 1.5 ms y para un ángulo de 180 grados se envía una señal por el pin de salida de 2 ms. Los servomotores como los que hemos usado necesitan ser activados con corrientes pequeñas es por eso que el Arduino Mega les da las corrientes necesarias para que se activen las cuales están en 40 ma, Todos los servomotores van conectados a las salidas del Arduino Mega al Pin2, Pin3, Pin4, Pin5, Pin6 y Pin7. También tenemos 6 Leds desde el D1 hasta el D6 solo como indicadores.



Figura 8. Inclusión de los servomotores a la interfaz de entrada que tiene como objetivo trasladar la información al microcontrolador. Para el procesamiento de los datos, en este caso de los colores primarios



Figura 9. Detalle del movimiento de nuestro brazo robótico por cada servo (pe1, pe2, pe3, pe4, pe5, pe6), expresado en milisegundos (500 milisegundos).

DISEÑO DE LAS CONEXIONES ELECTRICAS

En nuestros diseños tendremos 6 servomotores para los cuales necesitamos 6 cables de alimentación los cuales son el positivo que es el cable rojo, el negativo que es cable marrón y el envío de datos (PWM) modulación por ancho de pulso que es el cable naranja.

Cable rojo: 5V para todos menos para el servomotor de alto torque el cual ira conectado a un regulador de servomotor debido a que tiene un voltaje de 6,7v.

*nota1: las corrientes de los servomotores usadas para el cálculo son las corrientes en máximo torque

*nota2: El total de los voltajes es el mismo porque todos los componentes se encuentran en paralelo

*Entonces para manejar nuestro sistema, será necesaria una fuente capaz de proporcionar 5v y 11,66525 mA, por lo que se usará una fuente AT

Imágenes usadas en el prototipo

Para la prueba del prototipo se usaron un total de 6 imágenes y 12 cajas. El diseño representa la relación existente entre las muestras clasificadas adecuadamente y las muestras que no pertenecen a la clase pero fueron clasificadas como tal.

Medida del cubo: 3,5 cm de largo* ancho

Tipos de Niveles

1 Nivel: Números

2 Nivel: Logos

3 Nivel: Figuras geométrica



Figura 10. Imagen de Matlab Nivel 1



Figura 11. Imagen de Matlab Nivel 2

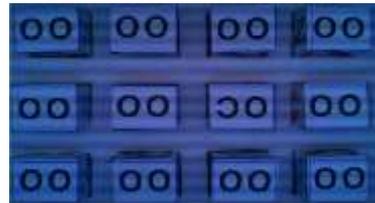


Figura 12. Imagen de Matlab Nivel 3

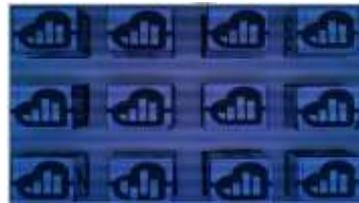


Figura 13. Imagen de Matlab Nivel 4



Figura 14. Imagen de Matlab Nivel 5



Figura 15. Imagen de Matlab Nivel 6

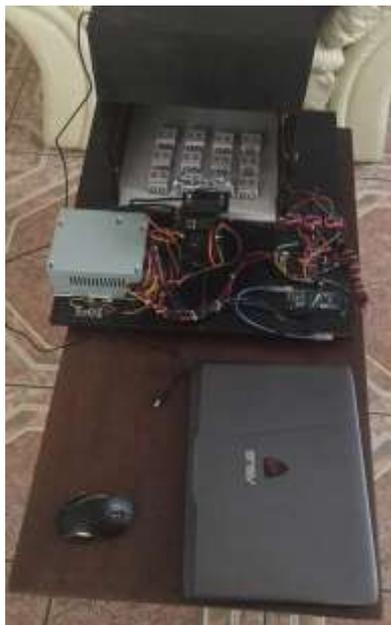


Figura 16. Proyecto terminado apagado



Figura 17. Proyecto terminado encendido

METODOLOGÍA

La serie de procedimientos realizados fueron a través de la técnica CBIR en cuanto a la clasificación y reconocimiento de los objetos. Se aplicó la metodología a fotos de los objetos en todos sus niveles. Se escogió esta técnica para que defina detalladamente lo que contiene cada imagen, de manera que detecte y estudie su contenido a través de la información que pueda derivar de la imagen, ya sea sus colores, formas, texturas, entre otros. El número total de imágenes adquiridas para el estudio fue de 360 fotografías para 10 pruebas realizadas en cada nivel, de las cuales era 1 fotografía por 12 muestras para 3 diferente niveles.

Para que el proceso de reconocimiento de fallas tenga éxito se debe realizar una correcta adquisición de la imagen y para ello, se eligió la cámara web con una resolución de 1280 x 720 pixeles, puesto que previo a su elección, se analizó las características que debe de tener y que serán necesarias para el proceso que se va a realizar, entre ellas, ser de una cámara color y digital, la cual estaría ubicada a una distancia de 30 cm respecto a las muestras a analizar. Esta se posicionó en una recámara de 40 cm x 35 cm para darle mejores condiciones de procesamiento de imagen a la cámara, siendo esta de fondo negro opaco con una bombilla halógena que garantizaría un fondo homogéneo y una iluminación correcta que evite reflejos y sombras en los contornos, para poder capturar con mayor detalle los objetos a explorar.

Por lo tanto para llevar a cabo la metodología, se consideraron 3 factores: luminosidad, orientación y escala, los cuales permitieron la extracción de características. Se trabajó la metodología CBIR a través de sus dos etapas; la primera fue de entrenamiento, la cual es considerada como una entrada del sistema, con bases de imágenes de los objetos en condiciones ideales, como lo es la iluminación y la resolución; la segunda es la etapa de recuperación, donde se da la salida del sistema y se identifican aquellas imágenes que no tienen defectos. Es por ello que a partir de la base de datos donde se encuentran todas las imágenes, se hace una lectura de estas a través del formato RGB. Asimismo, se le realizó el acondicionamiento para eliminar anomalías, posteriormente se convirtió las imágenes de formato RGB a HSI, de manera que a través de esta se obtendrá información necesaria para el análisis. Finalmente, se hizo una extracción de los puntos de interés y de características estadísticas (media y desviación estándar), obteniendo como resultado finalmente a un patrón normalizado con las características obtenidas por cada imagen.

MÉTODOS ESTADÍSTICOS

Se implementó un clasificador bayesiano, basado en la regla de Bayes del mínimo error. Un objeto, con unas características determinadas, pertenece a una clase si la probabilidad de pertenecer a ésta clase es mayor que la probabilidad de pertenecer a cualquier otra clase, como se muestra en la siguiente ecuación:

$X \in \Omega_i$, si $P(\varphi_i)p(X/\varphi_i) > P(\varphi_j)p(X/\varphi_j)$ donde Ω es el espacio de características, que está dividido en regiones Ω_i , $i = 1, 2, \dots, N$ donde N es el número de clases. $P(w_i)$ es la probabilidad a priori por la cual un objeto con características X , pertenece a la clase w_i y $p(m/w_i)$ es la función de probabilidad condicional de la clase w_i para

X (Niño, Z. L. S.; Ortiz, F. A. P, 2007). Por lo tanto, el presente trabajo definirá una característica discriminante X1 llamado grado de linealidad, dividiendo el número de píxeles del contorno del trazo vertical izquierdo por el número de filas entre el primero y el último píxel de dicho contorno, es decir:

$$X1 = \left[\frac{\text{número de píxeles del trazo izquierdo}}{\text{número de filas entre píxeles extremos}} \right]^{-1}$$

El clasificador a usar para identificar el objeto diferente de la muestra de la investigación es un indicador bayesiano, contando así con variables aleatorias y distribuciones normales. Mi indicador X1 se define de acuerdo a los píxeles de la imagen de cada objeto y para ello se seleccionó una media de 0.05 y se le designó a cada nivel una dispersión significativa con respecto a esta, de manera que para el primer nivel su desviación estándar fue de 0.40, para el segundo nivel de 0.25 y el tercer nivel de 0.05. Se le asignó una desviación menor al nivel más complejo, debido a que el brazo demoró más tiempo en detectar la falla en este nivel, llegando a tener mayor probabilidad de error, con un tiempo específicamente de 0.33 segundos para detectar la pieza fallada, mientras que en el primer y segundo nivel, 0.25 segundos y 0.28 segundos, respectivamente. La característica de cada indicador detalló ser de 0.834, 0.50 y 0.134 de grado de linealidad, para nivel 1, 2 y 3, respectivamente. Utilizando 2.398 de píxeles del trazo izquierdo para los objetos de nivel 1, 4 píxeles de trazo para el nivel 2 y 14.925 para el nivel 3; cada una entre 2, por ser el número de filas entre píxeles extremos. De acuerdo a las características del indicador, en el primer nivel se obtiene una eficiencia de 97.50%, en el segundo 96.41% y en el tercer nivel de 95.35%.

El error porcentual para el primer nivel es de 2.50%; para el segundo, 3.59% y para el tercer nivel, 4.65%.

3. RESULTADOS Y DISCUSIÓN

Los resultados de esta investigación confirman la gran utilidad de los detectores de muestras falladas, además de corroborar lo que distintos autores han realizado y recomendado en investigaciones similares. Como claramente recalcó Díaz en la conclusión de su trabajo, es mucho más eficiente utilizar una escala de colores que la escala de grises a la hora de evaluar la calidad de los frutos de café. Esto se evidencia en los resultados obtenidos por otros autores, como Sandoval que obtuvo un 95% de rendimiento y Silva y Lizcano obteniendo 96,36%, pero en su caso clasificando piñas de acuerdo a su maduración. A su vez, Sobrano Malpartida, al utilizar un identificador bayesiano en su investigación identificó características discriminantes que presentan aleatoriedad entre dos figuras, por lo que en la presente investigación al identificarse la misma situación en objetos con defectos en tres diferentes niveles, se usó el teorema de Bayes, como herramienta estadística para determinar un error porcentual menor a 5%, obteniendo como rendimiento 97.5%, 96.41% y 95.35% en el primer, segundo y tercer nivel, respectivamente. Por otro lado, se comprueba que el brazo robótico logra superar y ser mucho más rápido que la mente del ser humano al detectar el error en el objeto, por lo que podríamos concluir que el ojo humano puede ser reemplazado fácilmente por una máquina que identifica los tipos de objetos predeterminados con un alto porcentaje de certeza, además de ser capaz de determinar la posición de los mismos dentro del área de trabajo del robot, de manera que su aplicación en las industrias, originaría una disminución en los costos y mejoraría la optimización de procesos. Asimismo, las investigaciones mencionadas, sirvieron como base en este proyecto para mejorar la variación de la iluminación y obtener óptimos resultados.

4. CONCLUSIONES

Para desarrollar una aplicación que pueda controlar inteligentemente un brazo robot manipulador de seis grados de libertad el cual usa la visión artificial para comparar, detectar y extraer la muestra con falla, se tuvo que realizar una metodología para aplicar la técnica CBIR a las muestras de análisis, de manera que el sistema nos regresa las imágenes más parecidas, permitiendo identificar cual es el objeto que tiene la falla. Por otro lado, es importante que se desarrolle cada etapa, analizando las imágenes en condiciones ideales. Asimismo, se determinó características discriminantes en los objetos, según su nivel, mediante un método estadístico que permitió obtener resultados favorables, respetando que el nivel de error sea menor a 5%, obteniendo así un error porcentual de 2.50% para el primer nivel; 3.59% para el segundo nivel y 4.65% para el tercer nivel, equivalente a un rendimiento de 97.5%, 96.41% y 95.35%, respectivamente. Por lo tanto, se muestra que, si se pudo desarrollar una aplicación para el control inteligente de un robot manipulador de seis grados de libertad, a partir de la metodología CBIR, para la detección y extracción de muestras falladas mediante el reconocimiento de patrones.

REFERENCIAS BIBLIOGRÁFICAS

- Benavides-Alvarez, C.; Villegas-Cortez, J.; Román-Alonso, G.; Avilés-Cruz, C. 2015. Reconocimiento de rostros a partir de la propia imagen usando tecnica cbir. In X Congreso Espanol sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados MAEB (pp. 733-740).
- Berrocal, C. G.; Lundgren, K.; Löfgren, I. 2016. Corrosion of steel bars embedded in fibre reinforced concrete under chloride attack: state of the art. *Cement and Concrete Research*, 80: 69-85.
- Boudy González, O.; León Martínez, J. A.; Estrada Rodríguez, Y. E. 2011. Sistema para el control inteligente de un robot manipulador. *Revista Ciencias Técnicas Agropecuarias*, 20(2): 24-30.
- Hernández-Castro, F. 2016. Teoría del Color (Ingredients). Instituto Tecnológico de Costa Rica. Repositorio TEC: <https://repositoriotec.tec.ac.cr/handle/2238/7314>.
- Malpartida, E. A. S. 2003. Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot. Pontificia universidad católica del Perú, Perú.
- Martínez-Corral, L.; Martínez-Rubín, E.; Flores-García, F.; Castellanos, G. C.; Juárez, A. R.; López, M. J. D. 2009. Desarrollo de una base de datos para caracterización de alfalfa (*Medicago sativa* L.) en un sistema de visión artificial. *Phyton (Buenos Aires)*, 78(1): 43-47.
- Muro, J. S.; Lorente, P. N.; Garau, P. A.; Andrés, C. F.; Al-Hadithi, B. M. 2014. Solución basada en visión artificial para la inspección automatizada de gajos de mandarina.
- Niño, Z. L. S.; Ortiz, F. A. P. 2007. Caracterización de café cereza empleando técnicas de visión artificial. *Revista Facultad Nacional de Agronomía-Medellín*, 60(2): 4105-4127.
- Ortiz, O. J. R.; Mejía, M.; Castelblanco, J. S. U. 2019. Técnicas de inteligencia artificial utilizadas en el procesamiento de imágenes y su aplicación en el análisis de pavimentos. *Revista EIA*, 16(31):189-207.
- Ruge, R.; Adonayt, I.; Pérez Holguín, W. J. 2017. Diagnóstico tecnológico del uso de dispositivos programables en la industria boyacense. Caso de estudio: cadena agroindustrial de la panela. *Tecnura*, 21(52): 130-147.
- Sáenz, K. D. J. B. 2016. Clasificación de los frutos de café según su estado de maduración y detección de la broca mediante técnicas de procesamiento de imágenes. *Revista Prospectiva*, 14(1): 15-22.
- Segura-Castruita, M. A 2011 Identificación de yemas axilares en plántulas de papa basada en un sistema de visión con lógica difusa.
- Silva, L. A.; Lizcano, S. 2012. Evaluación del estado de maduración de la piña en su variedad perolera mediante técnicas de visión artificial. *Iteckne*, 9(1)