



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN MECATRÓNICA

**“DESARROLLO DE UN SISTEMA PARA DETECTAR LOS COLORES EN
TEJIDOS CON HILO UTILIZANDO VISIÓN ARTIFICIAL”**

TRABAJO DE TITULACIÓN PARA OBTENER EL TÍTULO DE INGENIERA EN
MECATRÓNICA

AUTOR: OSEJOS ROCHA SANTIAGO MARTÍN

DIRECTOR:

ING. IVÁN IGLESIAS NAVARRO

IBARRA - ECUADOR

2022



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1004400766		
APELLIDOS Y NOMBRES:	OSEJOS ROCHA SANTIAGO MARTÍN		
DIRECCIÓN:	Pasaje la Alborada y Juan de Velasco - Atuntaqui		
EMAIL:	smosejosr@utn.edu.ec		
TELÉFONO FIJO:	2925128	TELÉFONO MÓVIL:	0986062419

DATOS DE LA OBRA	
TÍTULO:	“DESARROLLO DE UN SISTEMA PARA DETECTAR LOS COLORES EN TEJIDOS CON HILO UTILIZANDO VISIÓN ARTIFICIAL”
AUTOR:	Santiago Martín Osejos Rocha
FECHA: DD/MM/AAAA	09/09/2022
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	Ingeniero en Mecatrónica
ASESOR /DIRECTOR:	Ing. Iván Iglesias Navarro

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 9 días del mes de septiembre de 2022

EL AUTOR:

A handwritten signature in blue ink, appearing to read 'Santiago Martín Osejos Rocha', with a stylized flourish underneath.

Nombre: Osejos Rocha Santiago Martín
Cédula: 1004400766



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN DIRECTOR

En calidad de tutor del trabajo de grado titulado: “DESARROLLO DE UN SISTEMA PARA DETECTAR LOS COLORES EN TEJIDOS CON HILO UTILIZANDO VISIÓN ARTIFICIAL” certifico, que el presente trabajo fue desarrollado por el egresado OSEJOS ROCHA SANTIAGO MARTÍN, bajo mi supervisión.

Ibarra, a los 9 días del mes de septiembre de 2022

A handwritten signature in blue ink, consisting of a stylized 'I' followed by a large, circular flourish.

Ing. Iván Iglesias Navarro
DIRECTOR DE TESIS

AGRADECIMIENTOS

Todo el honor y gratitud es para Dios, quien me dio la fortaleza y sabiduría para cumplir con todas las metas trazadas en todas las etapas de mis estudios. A mis padres por su ejemplo de vida, quienes me enseñaron el compromiso y dedicación. A mis hermanas y familiares, quienes me brindaron su cariño en todos los momentos que lo necesite.

En especial quiero agradecer a todos los CEO de Ingenious Works y Casatex por ser los impulsores de este trabajo de investigación, quienes me dieron la mano cuando más lo necesite. Demostrando su gran corazón y pasión por impulsar a jóvenes a cumplir sus sueños.

Agradezco la ayuda incondicional de mi tutor de tesis Ingeniero Iván Iglesias, quien me guió por el camino correcto en todo el proceso de titulación y a todos mis docentes quienes me compartieron de su experiencia y conocimiento en todos los semestres cursados.

Gracias totales a todos mis compañeros con quienes compartí momentos únicos, que hacían de la Universidad un lugar más llevadero.

Martín Osejas

DEDICATORIA

Este trabajo de titulación es dedicado a mis sobrinos *Alejita Casa* y *Felipe Casa*, quienes son mi inspiración para salir adelante y seguir cumpliendo mis metas y objetivos personales. Y de manera muy especial a mi querida madre

Ligia Gomez.

Martin Osejas

ÍNDICE DE CONTENIDO

RESUMEN	xii
ABSTRACT	xiii
CAPÍTULO I	1
INTRODUCCIÓN	1
1.1 Revisión de antecedentes	1
1.2 Planteamiento del problema y justificación	2
1.3 Objetivos	3
1.3.1 Objetivo general	3
1.3.2 Objetivo específico	3
1.4 Alcance	3
1.5 Justificación	3
CAPÍTULO II	5
MARCO TEÓRICO	5
2.1 Definición de Visión Artificial	5
2.2 Colores en la Visión Artificial	5
2.3 Componentes de color	7
2.4 Cámara	8
2.4.1 Adquisición de la imagen	9
2.4.2 Procesamiento de la imagen	9
2.4.3 Esquema de demostración de imágenes	10
2.4.4 Artefactos en la demostración de imágenes	11
2.4.5 Segmentación de imágenes	12
2.5 Iluminación en la Visión Artificial	13
2.5.1 Propiedades reflectantes	14
2.5.2 Fuentes de iluminación	15
2.5.3 Técnicas de iluminación	16
2.6 Diseño CAD	17
2.6.1 STL	18
2.7 Python	18
2.7.1 Descripción general de bibliotecas Python	18
CAPÍTULO III	21
METODOLOGÍA	21

3.1	Fase 1: Requerimiento de usuario	22
3.2	Fase 2: Diseño del sistema	22
3.3	Fase 3: Implementación del sistema de visión artificial	23
3.4	Fase 4: Validar el funcionamiento del sistema	23
CAPÍTULO IV		24
4.1	Requerimientos.....	24
4.1.1	Requerimientos del usuario	24
4.1.2	Requerimientos del técnico	25
4.1.3	Planteamiento de alternativas de solución	26
4.1.4	Ponderación de criterios	26
4.1.5	Ponderación de alternativas con cada criterio	27
4.1.6	Análisis de componentes	29
4.2	Diseño del sistema	30
4.2.1	Desarrollo de algoritmo de visión artificial	31
4.2.2	Diseño de carcasa.....	33
4.2.3	Impresión 3D de carcasa	37
4.2.4	Implementación del sistema embebido.....	40
4.2.5	Conexión de cámara.....	41
4.2.6	Implementación del sistema de luminosidad	42
4.3	Validación de la funcionalidad del sistema.....	44
4.3.1	Matriz de confusión	49
CAPÍTULO V		53
5.1	Conclusiones	53
5.2	Recomendaciones.....	54
REFERENCIAS.....		55
ANEXOS		59
	Anexo 1. Entrevistas a los usuarios.....	60
	Anexo 2. Matriz QFD.....	62
	Anexo 3. Algoritmo de visión artificial	63

ÍNDICE DE TABLAS

Tabla 1. Fuentes de iluminación	16
Tabla 2. Técnicas de iluminación	17
Tabla 3. Requerimientos del usuario.....	24
Tabla 4. Ponderación de requerimientos	24
Tabla 5. Requerimientos del técnico	25
Tabla 6. Solución alternativa 1	26
Tabla 7. Solución alternativa 2	26
Tabla 8. Nomenclatura de criterios	27
Tabla 9. Comparación de criterios	27
Tabla 10. Comparación criterio de alto nivel.....	27
Tabla 11. Comparación criterio de procesamiento	27
Tabla 12. Comparación criterio de calidad de imagen.....	28
Tabla 13. Comparación criterio de luminosidad	28
Tabla 14. Comparación criterio de interfaz de usuario	28
Tabla 15. Comparación criterio de material.....	28
Tabla 16. Resultados de la ponderación de criterios.....	29
Tabla 17. Descripción de componentes	29
Tabla 18. Características de filamentos	37
Tabla 19. Propiedades mecánicas del PLA	38
Tabla 20. Características de la cámara.....	41
Tabla 21. Matriz de observación.....	51
Tabla 22. Medidas de clasificación.....	52

ÍNDICE DE FIGURAS

Figura 1. Subconceptos de visión artificial.....	1
Figura 2. Ejemplo de reconocimiento de frutas utilizando el umbral óptimo.	6
Figura 3. Localización de bloques	6
Figura 4. Espacio de color RGB	7
Figura 5. Modelo de color HSV.....	8
Figura 6. Canalización típica de generación de imágenes dentro de una cámara digital	9
Figura 7. Matriz de filtros de color	10
Figura 8. Desenfoque en la imagen demostrada.	11
Figura 9. Falsos colores generados sobre una zona texturizada.	12
Figura 10. Segmentación de fondo.	12
Figura 11. Niveles de luminancia y disposición de atenuadores	13
Figura 12. Ángulos fotométricos	14
Figura 13. Ilustración de reflejos de luz lambertiana y especular	15
Figura 14. Fases para el cumplimiento de objetivos.....	21
Figura 15. Pareto de requisitos de calidad	25
Figura 16. Sistema de visión artificial	30
Figura 17. Diagrama de flujo.....	31
Figura 18. Ejemplo de algoritmo sin condiciones de ambiente controlado.....	33
Figura 19. Estructura de carcasa	34
Figura 20. Modelado pieza A	35
Figura 21. Modelado pieza B.....	35
Figura 22. Modelado pieza C.....	36
Figura 23. Carcasa de Raspberry pi4	37
Figura 24. Impresión 3D pieza A.....	38
Figura 25. Modelado pieza B.....	39
Figura 26. Modelado pieza C.....	39
Figura 27. Impresora Prusa	40
Figura 28. Sistema embebido.....	41
Figura 29. Técnica frontal de iluminación.....	43
Figura 30. Sistema de iluminación	44
Figura 31. Funcionamiento del sistema	44

Figura 32. Interfaz de usuario	45
Figura 33. Configuración AutoStart de Raspberry	46
Figura 34. Botón iniciar video	47
Figura 35. Botón capturar imagen	47
Figura 36. Botón detectar color	48
Figura 37. Ventana Detector de color	48
Figura 38. Matriz de confusión.....	50

RESUMEN

La perspectiva del color varía dependiendo de cada persona, ya sea por condiciones del medio ambiente o por problemas oculares de las personas. En este trabajo se presenta un sistema de visión artificial para la detección de colores en tejidos con hilo con el propósito de evitar la identificación errónea del color por medio de la visión humana. Se utilizan librerías de visión artificial como OpenCV y Numpy para leer archivos donde se definen los rangos RGB deseados. El algoritmo de programación se realiza en Python, cuenta con la implementación de un sistema embebido raspberry pi4, un sistema de iluminación led, y una cámara compatible con raspberry. La estructura para la implementación de los componentes del sistema es parcialmente cerrada, lo que permite la entrada de la muestra y delimita el área de detección, creando un ambiente de iluminación controlado. La comunicación entre el sistema y los usuarios está definida por una interfaz de usuario realizada en QtDesigner. El sistema de detección de color tiene una precisión promedio de 0,90, una sensibilidad de 0,87 y una especificidad de 0,97 en los colores propuestos.

Palabras claves: Visión artificial, interfaz de usuario, matriz de confusión, RGB.

ABSTRACT

The color perspective varies depending on each person, either due to environmental conditions or eye problems. In this work, an artificial vision system is presented for detecting colors in threaded fabrics to avoid erroneous identification of color by means of human vision. Machine vision libraries such as OpenCV and Numpy are used to read files where the desired RGB ranges are defined. The programming algorithm is done in Python, it has the implementation of an embedded raspberry pi4 system, a led lighting system, and a camera compatible with raspberry. The structure for implementing the system components is partially closed, allowing the sample to enter and delimit the detection area, creating a controlled lighting environment. The communication between the system and the users is defined by a user interface made in QtDesigner. The color detection system has an average precision of 0.90, a sensitivity of 0.87, and a specificity of 0.97 in the proposed colors.

Keywords: Artificial vision, user interface, confusion matrix, RGB.

CAPÍTULO I

INTRODUCCIÓN

1.1 Revisión de antecedentes

A diferencia de las técnicas de inteligencia artificial, como las redes neuronales, que se usaron considerablemente desde la década de 1940, las técnicas de visión por computadora comenzaron a existir de manera significativa a fines de la década de 1960 y principios de la década de 1970 (Kakani et al., 2020).

Las técnicas de visión artificial inicialmente fueron basadas en conceptos como la imitación del sistema visual humano a través de diversos conocimientos sobre cómo funcionan los esquemas de cámara, las proyecciones y la fotogrametría. En pocas décadas, la visión por computadora hizo su existencia combinada en todos los campos posibles como se observa en la Figura 1, el reconocimiento de patrones, el aprendizaje automático, los gráficos por computadora, las reconstrucciones 3D, la realidad virtual y la realidad aumentada (Fuentes et al., 2015).

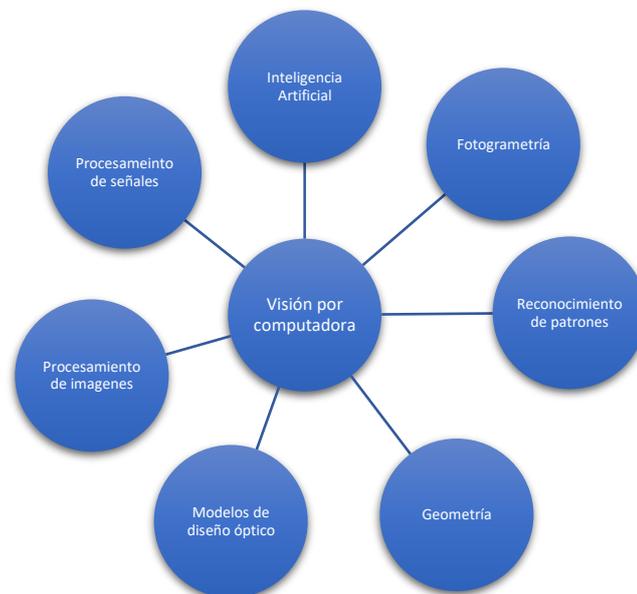


Figura 1. Subconceptos de visión artificial

Fuente: Adaptado de (Kakani et al., 2020)

Hay muchos campos en los que interviene la visión por computadora, incluido el mapeo terrestre y aéreo de recursos naturales, el monitoreo de cultivos, la agricultura de precisión, la robótica, el guiado automático, la inspección no destructiva de las propiedades del producto, el control de calidad y la clasificación en las líneas de procesamiento y, en general, automatización de procesos. Esta amplia gama de aplicaciones es el resultado del hecho de que los sistemas de visión artificial proporcionan cantidades sustanciales de información sobre la naturaleza y los atributos de los objetos presentes en una escena (Cubero et al., 2011).

1.2 Planteamiento del problema y justificación

Hablar de tejidos con hilo es referirse a una de las principales industrias del país, como lo es la industria textil. Esta es la tercera industria más grande en el sector manufacturero del Ecuador, la observación visual de la materia prima en esta industria es importante para una correcta elección del material. El hilo como materia prima cumple con un rol importante en las empresas textiles y la selección de un color incorrecto puede cambiar la perspectiva de un tejido agradable, en un tejido poco atractivo (Mucho Mejor Ecuador, 2021).

Aunque la observación visual a la hora de elegir un hilo es importante, la manera en que las personas interpretan visualmente los colores varía dependiendo de factores en la iluminación como lo es el brillo, saturación, matiz, contraste, entre otros factores lumínicos. En la mayoría de las personas el sistema visual funciona de la misma manera, pero las interpretaciones de color varían en cada observador (Palacios, 2017).

Se ha evidenciado que en las distribuidoras de hilos solo existe el reconocimiento de color de manera visual por partes de los trabajadores. Existiendo diferencias en la interpretación del color de las muestras de tejidos con hilo entre trabajadores y clientes, generando dudas y errores en la compra del hilo como materia prima para la fabricación de sus productos. Perjudicando económicamente al cliente y a la empresa.

En este trabajo de investigación se propone abordar el problema de identificación visual de los colores de tejidos con hilo mediante el uso de visión artificial. Por lo que se propone el desarrollo de un sistema capaz de identificar el color del hilo de una imagen y etiquetarlo con su respectivo nombre según los catálogos de colores de una distribuidora de hilos.

1.3 Objetivos

1.3.1 Objetivo general

Desarrollar un sistema para detectar los colores en tejidos con hilo utilizando visión artificial.

1.3.2 Objetivo específico

- Determinar las necesidades y requerimientos presentados en una distribuidora de hilos.
- Diseñar un sistema de visión artificial de acuerdo con los requerimientos obtenidos.
- Implementar el sistema desarrollado.
- Validar el funcionamiento del sistema.

1.4 Alcance

El presente proyecto de investigación está orientado al desarrollo de un sistema de visión artificial, mediante un algoritmo en software libre que permita detectar los siete colores principales en tejidos con hilos. Para lo cual se realizará un prototipo que cuente con una carcasa en impresión 3D que sirva como soporte para los componentes externos a implementar como lo son: una cámara que cumpla con las características requeridas de compatibilidad de software libre, para la captura y procesamiento de imágenes, un sistema embebido para la ejecución y conexión del código y una interfaz de usuario para la interacción del usuario con el sistema.

1.5 Justificación

La implementación de un sistema de visión artificial para la detección de colores de tejidos con hilo se fundamenta en los siguientes ámbitos: En el aspecto tecnológico, en la actualidad se han desarrollado varias herramientas tecnológicas enfocadas en ayudar a las personas de diversas maneras, una de estas es por medio de visión artificial. La extracción, procesamiento e interpretación repetitiva y a gran escala de imágenes hace que la visión artificial de resultados de valor superiores comparados con la visión humana. Es por eso, que utilizar visión artificial para la identificación de colores en los tejidos con hilo es una de las mejores opciones, ya que

el ojo humano no puede realizar tareas repetitivas de identificación de colores por un largo tiempo.

En el aspecto económico, el uso del detector de colores por medio de visión artificial tendría una alta demanda por el conjunto de pequeñas y medianas empresas. Por lo cual, su desarrollo beneficiaría a gran parte de las empresas de hilos al tener un costo menor a máquinas industriales detectoras de color que son muy costosas y sobredimensionadas para el uso de las pequeñas y medianas empresas.

En el aspecto social, con la realización de este dispositivo para detectar colores, se plantean soluciones para la sociedad en el campo de visión artificial, aplicando principios de visión por computador para desarrollar un dispositivo útil para un amplio número de pequeñas y medianas empresas dedicadas al sector textil, mejorando su economía y la calidad de servicio. Además, que ayudará a reducir los problemas de salud relacionados con el daño de la vista. Evitando en las personas enfermedades visuales, cansancio mental y pérdidas económicas por identificación de colores erróneos. Abordando la responsabilidad social con la mancomunidad impartida por la Universidad Técnica del Norte.

CAPÍTULO II

MARCO TEÓRICO

2.1 Definición de Visión Artificial

Según McGuinness et al. (2021), la visión artificial se emplea comúnmente en las fábricas para el control de calidad y la inspección de piezas, donde la colocación y las dimensiones de las piezas suelen estar muy bien definidas y su ambiente es homogéneo y controlado. De hecho, un sistema de visión artificial es mucho más objetivo, más rápido y menos propenso a errores que la inspección manual en este ambiente.

Según el estudio de Zandonadi et al. (2005), también se lo aplica en la agricultura aun que, hasta el momento, no existe un sentido completo de éxito por las condiciones de heterogeneidad del ambiente y su variación de intensidad y espectro de luz, aunque su nivel de estimación en algunas ocasiones ha sido más alto que el ojo humano, por ejemplo, la identificación de plantas de maíz atacadas por plagas mediante el uso de imágenes infrarrojas.

2.2 Colores en la Visión Artificial

En su estudio Bulanon et al. (2002), mencionó que el ojo humano al igual que la visión artificial también es más sensible al color que al brillo en condiciones normales de iluminación. El color es un descriptor eficaz para realzar un objeto en una imagen, lo que simplifica la identificación y extracción de objetos de una imagen como se observa en la Figura 2.

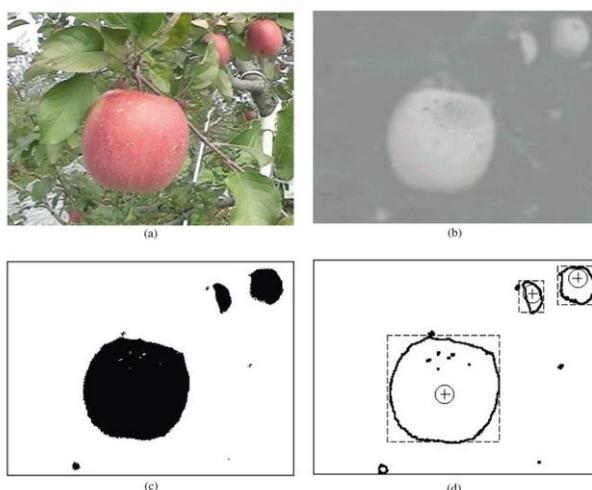


Figura 2. Ejemplo de reconocimiento de frutas utilizando el umbral óptimo.

Fuente: (a) imagen en color original de una manzana Fuji; (b) imagen mejorada utilizando la diferencia de color rojo; (c) imagen después de la umbralización; (d) imagen después del borde y extracción de características (Bulanon et al., 2002)

El número de píxeles y la velocidad de fotogramas suelen estar determinados por la tecnología de visualización. El tamaño del bloque no puede ser arbitrariamente pequeño, ya que un bloque debe contener suficientes píxeles para tener una probabilidad aceptable de detectar el color correcto. También existe una compensación entre el número de colores utilizados y la tasa de error (Sun et al., 2021). Para identificar el color de un píxel recibido, el método tradicional consiste en obtener la distancia euclidiana entre el valor real de rojo, verde y azul (RGB) recibido y el valor RGB de los colores disponibles, primero obteniendo la localización del área de trabajo como se muestra en la Figura 3. Las fuentes de luz en el entorno aportan ruido aditivo al valor RGB del píxel recibido, lo que provoca una desviación del color transmitido en el píxel recibido. Para minimizar los errores de decodificación causados por la identificación incorrecta del color de un píxel en el receptor, los colores se seleccionan para que tengan la máxima distancia en el espacio de color (Du et al., 2017).

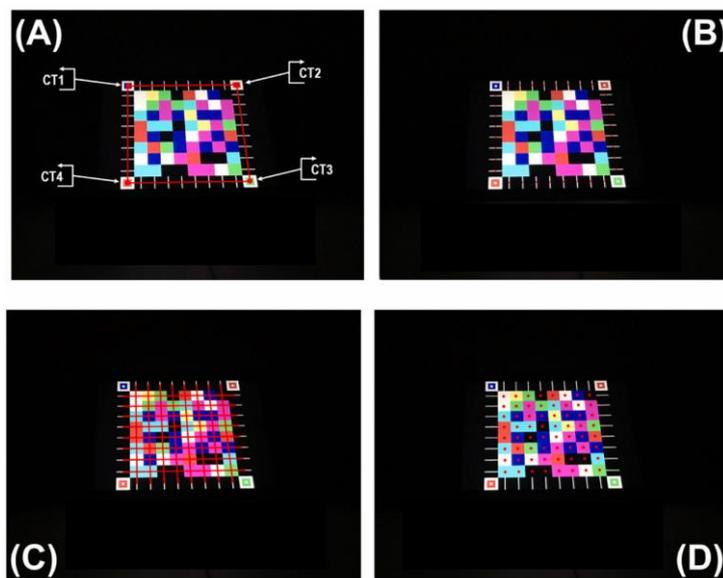


Figura 3. Localización de bloques

Fuente: (a) identificar esquinas, (b) ubicar líneas de seguimiento, (c) señalar el centro de cada bloque y (d) obtener la posición del punto central de cada uno (Sun et al., 2021)

2.3 Componentes de color

La mayoría de las cámaras funcionan en el espacio RGB, indicando la intensidad de los tres colores primarios (rojo, verde, azul) de los cuales se puede descomponer una variedad de colores. Este espacio de color RGB se basa en el sistema de coordenadas cartesianas como se muestra en la Figura 4. Cada color consta de componentes espectrales primarios es decir el color rojo estaría definido por (255,0,0), el color verde (0,255,0), el color azul (0,0,255), además de colores como el negro y blanco definidos por (0,0,0) y (255,255,255) respectivamente. La escala de 0 a 255 a lo largo de cada eje define el brillo (Dorj et al., 2017).

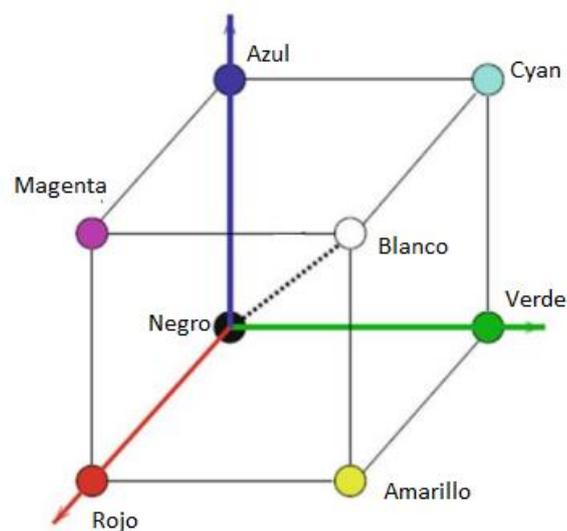


Figura 4. Espacio de color RGB

Fuente: Adaptado de (Dorj et al., 2017)

Este modelo de color se basa en una síntesis aditiva simple, no es completamente intuitivo desde el punto de vista perceptivo. Una de las grandes desventajas de usar el espacio de color RGB es que luego es difícil separar la información de color del brillo. En lugar de utilizar colores primarios, el tono, la saturación y el valor (HSV) o HSB se utilizan como descripciones de color para proporcionar un mecanismo más intuitivo para los usuarios. El modelo HSV ofrece un espacio de color más interesante que es perceptiblemente uniforme como se muestra en la Figura 5 (Al-Osaimi et al., 2011) .

- El tono (H) de un color se refiere a qué color puro se parece. Todos los tintes, tonos y matices de rojo tienen la misma tonalidad. Los tonos se describen mediante un

número que especifica la posición del color puro correspondiente en la rueda de colores, como una fracción entre 0 y 1. El valor 0 se refiere al rojo; 1/6 es amarillo; 1/3 es verde; y así sucesivamente alrededor de la rueda de colores.

- La saturación (S) de un color describe qué tan blanco es el color. Un rojo puro está completamente saturado, con una saturación de 1; los tintes de rojo tienen saturaciones inferiores a 1; y el blanco tiene una saturación de 0.
- El valor (V) de un color, también llamado luminosidad, describe qué tan oscuro es el color. Un valor de 0 es negro, con una luminosidad creciente alejándose del negro.

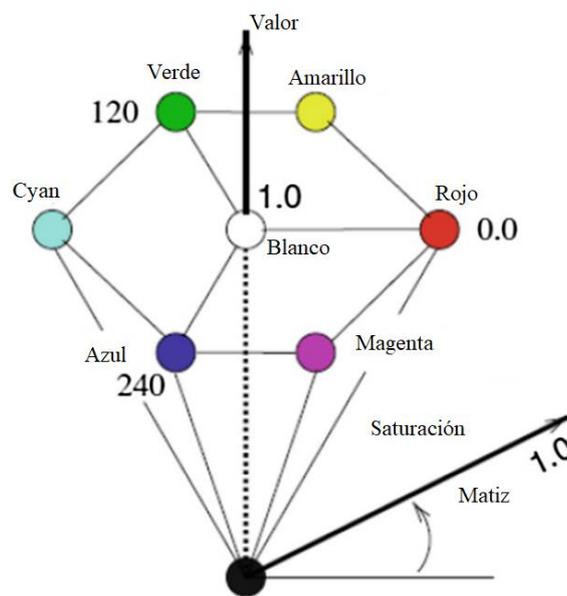


Figura 5. Modelo de color HSV

Fuente: Adaptado de (Al-Osaimi et al., 2011)

2.4 Cámara

El funcionamiento de una cámara digital se muestra en la Figura 6, donde se captura una canalización de alto nivel para la generación de imágenes dentro de una cámara digital, que involucra una secuencia de pasos de procesamiento de hardware y software. Los rayos de luz de la escena ingresan a la cámara a través de un conjunto de lentes, seguidos por un filtro anti-aliasing, conjuntos de filtros de color (CFA) y, finalmente, el sensor de imágenes. El sensor convierte la señal analógica en digital y produce una imagen RAW. Esta imagen se mejora aún más y, por lo general, implica pasos de procesamiento como demostración, corrección gamma,

compresión y otras operaciones antes de la generación de la imagen final (Bennabhaktula et al., 2022).

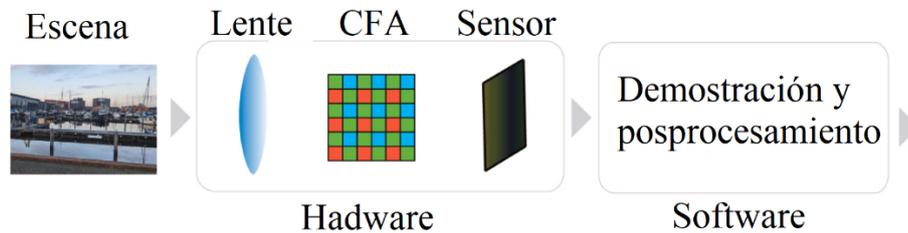


Figura 6. Canalización típica de generación de imágenes dentro de una cámara digital

Fuente: Adaptado de (Bennabhaktula et al., 2022)

El paquete de la cámara generalmente consiste en el sensor de imágenes y sistema óptico. El sistema óptico consta de lente, apertura y obturador. Estos componentes se capturan en la matriz de la cámara a través de la distancia focal, el diámetro de apertura, el campo de visión y la función de transferencia óptica. En cambio, los sensores de imágenes están influenciados por la resolución, la velocidad de fotogramas y profundidad (Heimberger et al., 2017).

2.4.1 Adquisición de la imagen

Para Ji et al. (2009), es esencial diseñar un sistema de adquisición de imágenes apropiado, es la base del sistema de visión artificial. El propósito es la adquisición de imágenes de la calidad correspondiente para satisfacer los requisitos del análisis. La confirmación de una adquisición depende en gran medida de que los investigadores comprendan el objeto estudiado y esperen la calidad de la imagen.

2.4.2 Procesamiento de la imagen

Según Ji et al. (2009), los pasos generales del procesamiento de imágenes son: transformación del espacio de color, filtrado, atenuación, detección de bordes, entre otras formas de acuerdo con el requerimiento del estudio. Según McGuinness et al. (2021), se recomienda cuatro pasos básicos para el procesamiento de imágenes: preprocesamiento, segmentación, extracción de características e interpretación. Esta regla se la puede romper dependiendo de la necesidad de investigación.

El procesamiento de imágenes es uno de los principales procedimientos en la aplicación de visión artificial. Sin embargo, todavía no es común en algunos campos de la industria abiertos ya que existen dificultades asociadas con la automatización en ambientes al aire libre, como los efectos de iluminación, la variabilidad del producto, la suciedad y clima. Además, la ubicación y las dimensiones de los objetos no están bien definidas en comparación con un entorno de fábrica controlado (Ji et al., 2009).

2.4.3 Esquema de demostración de imágenes

La mayoría de las cámaras a color están equipadas con un sensor de dispositivo acoplado de carga (CCD) único como se ve en la Figura 7. La superficie de dichos sensores está cubierta por una matriz de filtros de color (CFA), que consiste en un mosaico de filtros espectralmente selectivos, de modo que cada elemento CCD muestrea solo uno de los tres componentes de color rojo (R), verde (G) o azul (B). Bayer CFA es la matriz de filtros más utilizada para proporcionar una imagen CFA en la que cada píxel se caracteriza por un único componente de color. Para estimar el color (R, G, B) de cada píxel en una imagen de color verdadero, se deben determinar los valores de los dos componentes de color que faltan en cada píxel de la imagen CFA (Losson et al., 2010).

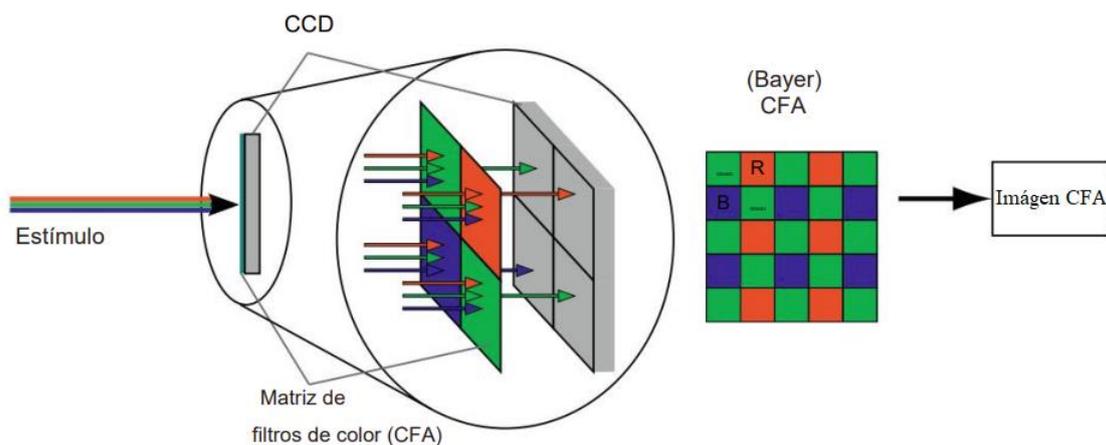


Figura 7. Matriz de filtros de color

Fuente: Adaptado de (Losson et al., 2010)

2.4.4 Artefactos en la demostración de imágenes

Los principales artefactos causados por la demostración son el desenfoque, los colores falsos y el efecto cremallera (Losson et al., 2010).

- El desenfoque se encuentra en áreas donde la información de alta frecuencia, que representa detalles o bordes precisos, se altera o borra. La Figura 8 ilustra diferentes niveles de desenfoque según el esquema de demostración (Losson et al., 2010).

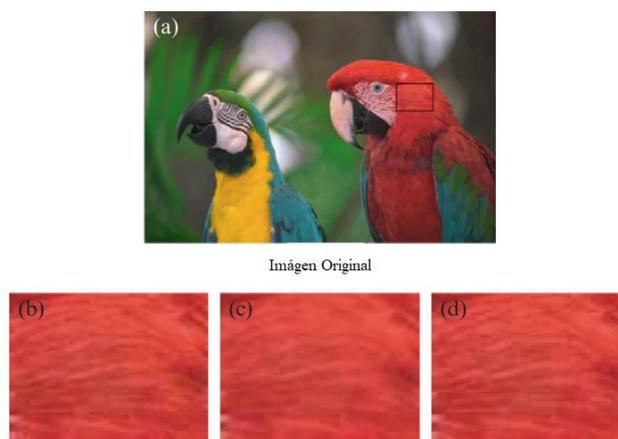


Figura 8. Desenfoque en la imagen demostrada.

Fuente: (a) Imagen original de los loros. La imagen (b) es un extracto de la imagen original ubicada por un cuadro negro. Las imágenes (c) y (d) son los extractos correspondientes de las imágenes, respectivamente, demostrada por interpolación bilineal (Losson et al., 2010).

- Efecto cremallera, la razón principal de este artefacto es la interpolación de niveles que pertenecen a áreas homogéneas que representan diferentes objetos. Ocurre en cada píxel donde la dirección de interpolación (horizontal o vertical) es cercana a la del gradiente de color calculado en la imagen original (Losson et al., 2010).
- El color falso en un píxel corresponde a una gran distancia entre el color original y el estimado, en el espacio de color de adquisición RGB. La Figura 9 muestra que este fenómeno no se caracteriza por una estructura geométrica específica en la imagen. La estimación incorrecta de los componentes de color puede causar colores falsos

perceptibles, particularmente en áreas con frecuencias espaciales altas (Losson et al., 2010).



Figura 9. Falsos colores generados sobre una zona texturizada.

Fuente: (a) Imagen original (b) Extracto de la imagen original ubicada por un recuadro negro. (c) Extracto demostrado por el esquema de Wu & Zhang (2004), con artefactos en un círculo negro (Losson et al., 2010).

2.4.5 Segmentación de imágenes

En el diseño de un sistema de visión artificial es importante la segmentación del objeto a identificar del fondo como se muestra en la Figura 10, ya que este puede contener objetos diferentes con una variedad de colores y texturas que pueden ocasionar un error en el buen funcionamiento del algoritmo. La segmentación incorrecta del fondo reduce la precisión de los procesos de visión artificial posteriores (Slaughter et al., 2008).

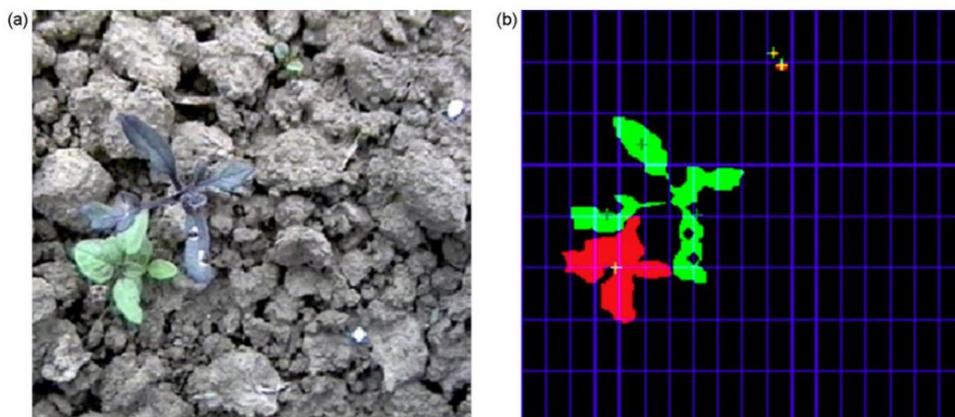


Figura 10. Segmentación de fondo.

Fuente: a) Imagen a color, b) Imagen segmentada (Slaughter et al., 2008).

La segmentación exitosa requiere la selección de un nivel de umbral óptimo. El umbral óptimo brinda la mejor separación del objeto y el fondo; sin embargo, la dificultad radica en la selección del umbral óptimo (Sabzi et al., 2018).

2.5 Iluminación en la visión artificial

La fuente de luz es un componente importante de un sistema de visión artificial y debe enfatizarse en el diseño del sistema. Hasta cierto punto, la elección y el diseño de las fuentes de luz es generalmente la clave para el éxito de cualquier sistema de visión artificial (Ji et al., 2009).

En la investigación de Kulappurath & Shamey (2021), recomienda utilizar niveles de luminancia más altos para las evaluaciones de percepción de las diferencias de color cuando sea posible y evitar grandes variaciones de luminancia para mejorar la repetibilidad y reproducibilidad de los datos de evaluación visual y las concordancias con los valores calculados, en la siguiente Figura 11 se muestra la diferencia de los niveles de luminancia.

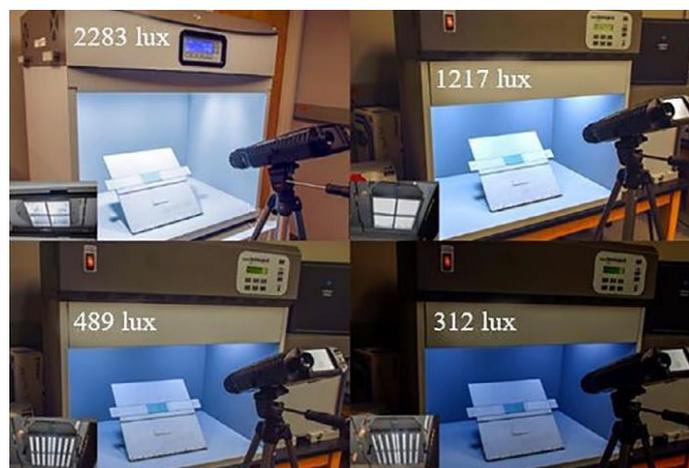


Figura 11. Niveles de luminancia y disposición de atenuadores

Fuente: Adaptado de (Kulappurath & Shamey, 2021)

Las condiciones de la imagen se componen de dos casos, es decir, interior y exterior, de los cuales, el más complejo son las condiciones exteriores que se dividen en soleado, nublado, nublado, tarde y así sucesivamente. Las condiciones interiores se dividen en luz natural, incandescente, fluorescente y otras. Bajo diferentes condiciones, las imágenes muestran

diferentes características. Como las sombras obvias de los objetos observados bajo el sol o la luz incandescente, además, el brillo de la imagen fluctúa con la variación de la iluminación (Søgaard & Olsen, 2003).

2.5.1 Propiedades reflectantes

El sistema visual humano proporciona constancia del color de un objeto en un rango visual bastante grande dando como resultado que el color del objeto sea el mismo a pesar de las diversas condiciones de luz, por ejemplo, el sol provoca una reflectancia variable de los objetos. Sin embargo, en los sistemas de visión artificial los cambios en las condiciones de iluminación pueden generar imágenes de mala calidad, lo que provoca un reconocimiento de objetos incompleto e inadecuado. Dado que las imágenes contienen información sobre la reflectancia de los objetos, las condiciones de iluminación tienen un impacto significativo en las características de la imagen (Bulanon et al., 2002).

Las superficies que vemos a nuestro alrededor suelen ser brillantes. Pueden parecer muy brillantes, bastante mate o algo intermedio, pero prácticamente todas las superficies que nos rodean exhiben reflejos en diversos grados, es así como se representan los ángulos fotométricos en la Figura 12. Los reflejos no son la única fuente de variación de intensidad en una superficie; incluso con una iluminación uniforme de una superficie mate de color uniforme, habrá un sombreado suave debido al ángulo de incidencia de la iluminación entrante en relación con la superficie normal (Shafer, 1985).

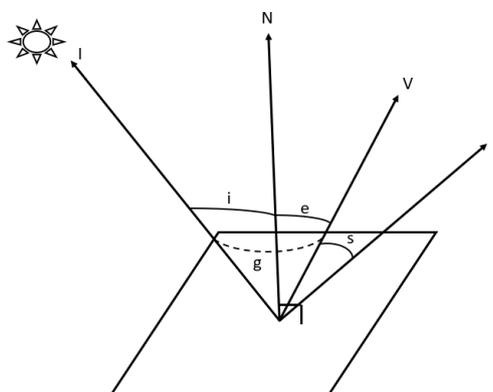


Figura 12. Ángulos fotométricos

Fuente: (Shafer, 1985)

Donde:

i = ángulo de incidencia

I = dirección de iluminación

N = superficie normal

e = ángulo de salida

V = dirección de observación

J = dirección de reflexión especular

s = ángulo no especular

g = ángulo de fase

Según Al-Osaimi et al. (2011), hay dos tipos de reflejos de luz en las superficies, a saber, Lambertiana y Especular. Una superficie Lambertiana refleja la luz incidente por igual en todas las direcciones, pero una superficie Especular refleja la luz incidente principalmente en la dirección de reflexión del espejo como se muestra en la Figura 13.

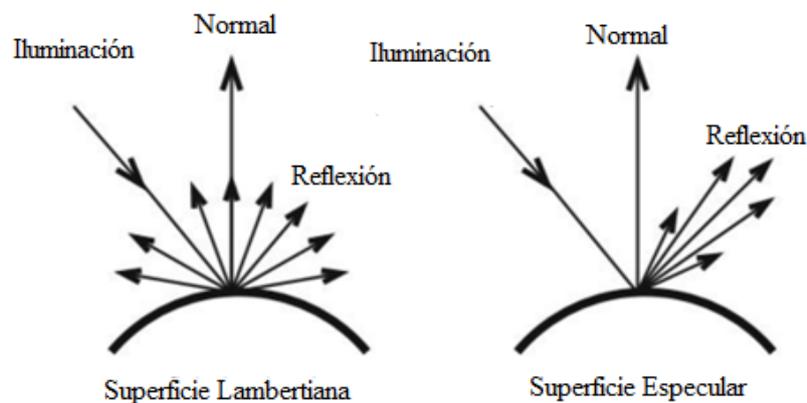


Figura 13. Ilustración de reflejos de luz lambertiana y especular

Fuente: (Al-Osaimi et al., 2011)

2.5.2 Fuentes de iluminación

El propósito de las fuentes de iluminación es controlar la manera en que la cámara ve las diferentes muestras del objeto y determinar si cumple con los requisitos para su adquisición. Si se utiliza una buena iluminación se podrá optimizar de mejor manera la solución del algoritmo, por el contrario, si se tiene una mala iluminación la adquisición de la imagen será pésima y por

lo tanto la solución al algoritmo será imposible en muchos casos. Las fuentes de luz más comunes se describen en la Tabla 1 (Albert & Antiga, 2012).

Tabla 1. Fuentes de iluminación

Fuente de luz	Ventaja	Desventaja
Halógenas	<ul style="list-style-type: none"> • Muy luminosos • Ajustable 	<ul style="list-style-type: none"> • No se puede utilizar como luz estroboscópica • Caliente • Se degrada con el tiempo
Incandescencia	<ul style="list-style-type: none"> • Económicas 	<ul style="list-style-type: none"> • No se puede utilizar como luz estroboscópica • Caliente • Se degrada con el tiempo
Fluorescentes	<ul style="list-style-type: none"> • Multitud de formas y tamaños • Iluminación difusa minimiza los reflejos 	<ul style="list-style-type: none"> • No se puede utilizar como luz estroboscópica • Parpadeo • Difícil de atenuar • Se degrada con el tiempo
Laser	<ul style="list-style-type: none"> • Se puede utilizar como luz estroboscópica • Dirigible • Útil en 3D 	<ul style="list-style-type: none"> • Caro • Textura granulosa • Frágil • Seguridad ocular para los trabajadores
Xenón	<ul style="list-style-type: none"> • Se puede utilizar como luz estroboscópica • De alta intensidad en períodos cortos (flash) 	<ul style="list-style-type: none"> • Se degrada con el tiempo
LED	<ul style="list-style-type: none"> • De larga duración (10.000 horas o más) • Se puede utilizar como luz estroboscópica • Disponible en longitudes de onda específicas (colores) • Bajo consumo 	<ul style="list-style-type: none"> • Relativamente caro

Fuente: (Albert & Antiga, 2012)

2.5.3 Técnicas de iluminación

En el estudio de Albert & Antiga (2012), se mencionan diferentes técnicas de iluminación para resaltar los objetos a analizar. A continuación, se muestra la matriz con las diferentes técnicas de iluminación en la Tabla 2.

Tabla 2. Técnicas de iluminación

Iluminación	Aplicación	Ventajas	Inconvenientes
Frontal	Indicada para superficies con pocos reflejos: papel, tela. Detección de marcas. Diferencia de colores y caracteres.	Elimina sombras, se puede utilizar a grandes distancias cámara/objeto.	Intensos reflejos sobre superficies reflectantes.
Lateral	Indicada para resaltar bordes, rayas y fisuras en una dirección determinada.	Resalta los relieves por pequeños que sean, resultando una sombra muy definida.	Con ángulos pequeños respecto a la horizontal, la luz producirá sombras en todos los relieves y en el contorno de la pieza.
Campo Oscuro	Indicada para resaltar incrustaciones y códigos alfanuméricos con poco contraste en metal sobre metal o gris sobre gris.	Destaca los detalles en superficies con muy poco contraste.	No es recomendable en superficies que absorban la luz.
Contraste	Indicada para la inspección de la silueta del objeto.	Permite inspecciones de siluetas e impurezas.	No permite reconocer los detalles superficiales del objeto, códigos, inscripciones, etc.
Axial difusa	Indicada para la inspección superficies planas reflectantes.	Permite inspecciones de códigos en materiales altamente reflectantes.	No permite reconocer relieves en el objeto.
Difusa tipo domo	Indicada para la inspección de superficies tales como instrumental médico, espejos, compact disk, latas, etc.	Eliminación de sombras y minimizador de arrugas, polvo y relieves.	Coste elevado.
Láser	Ajuste de procesos de corte, control de profundidad de objetos.	No le influye la iluminación externa.	Si se utilizan lentes cilíndricas para conseguir una línea o un patrón concreto, el láser no tiene la misma intensidad lumínica a lo largo del patrón.

Fuente: (Albert & Antiga, 2012)

2.6 Diseño CAD

Para el control de iluminación se toma en cuenta el diseño de una carcasa utilizando el diseño asistido por computadora (CAD). Este modelo CAD es el punto de partida para muchas aplicaciones posteriores, como la generación de mallas, el análisis estructural, térmico y de fluidos, la creación rápida de prototipos, el mecanizado controlado numéricamente, la fundición, los gráficos por computadora y el renderizado en tiempo real. Cada una de estas aplicaciones posteriores tiene requisitos específicos para la definición y representación de la geometría. Por lo tanto, el éxito de la aplicación posterior depende en gran medida de la precisión y consistencia de la geometría de entrada (Patel et al., 2006).

2.6.1 STL

La estereolitografía (STL) es un formato de intercambio de datos ampliamente utilizado en la industria de creación rápida de prototipos. Para fabricar modelos correctamente, la geometría de entrada debe ser geométrica y topológicamente correcta. Los datos del modelo de diseño asistido por computadora (CAD) se pasan con frecuencia a varios procesos de fabricación de forma libre sólida utilizando la representación de facetas poligonales STL (Rock & Wozny, 1992).

2.7 Python

Python es actualmente el lenguaje de programación más popular en el campo del análisis de datos, probablemente atribuido a la licencia libre y la gran cantidad de bibliotecas disponibles. La ventaja de Python radica en las poderosas herramientas (por ejemplo, módulos, datos, estructuras, bibliotecas y funciones) que ofrece a programadores y analistas de datos (Al-jabery et al., 2020).

2.7.1 Descripción general de bibliotecas Python

Esta sección se revisa y explora las bibliotecas más populares que están integradas en el entorno de Python para el procesamiento de datos y visión artificial en general. Estas bibliotecas son:

Numpy: el módulo numérico Python **Numpy** es el paquete fundamental para la computación científica en Python. Representa la base para otras bibliotecas importantes como **Pandas** y **Scikit-learn**. **Numpy** proporciona estructuras de datos convenientes para leer y almacenar tipos de datos arbitrarios, lo que acelera el proceso de integración con las bases de datos. También comparte muchos conceptos con MATLAB, lo que brinda un nivel de comodidad para los programadores de MATLAB que usan Python. Aunque **Numpy** suele estar disponible con la mayoría de los paquetes de instalación de Python, debe importarse antes de poder utilizar sus funciones (NumPy, 2008).

Matplotlib: La biblioteca **Matplotlib** permite a los programadores de Python generar gráficos de matrices en 2D o 3D. **Matplotlib** genera gráficos con comandos muy similares a los disponibles en MATLAB, pero es independiente de MATLAB y se usa en el entorno de Python. Esta biblioteca se construyó sobre la extensión **Numpy** de Python. Este paquete permite a los usuarios de Python generar y representar sus resultados en figuras bien organizadas, similares a las figuras de MATLAB (Hunter et al., 2014) .

Pandas: La biblioteca de **Pandas** <https://pandas.pydata.org/> proporciona a Python herramientas eficaces y potentes para el análisis de datos al combinar la potencia informática de matriz de alto rendimiento de Numpy con la flexibilidad de la manipulación de datos de bases de datos y hojas de cálculo. **Pandas** consta de estructuras de datos de matrices etiquetadas, objetos de indexación que permiten una indexación simple y de varios niveles, herramientas integradas para la transformación de conjuntos de datos, herramientas de importación y exportación de datos, herramientas eficientes de almacenamiento de datos y estadísticas de ventanas móviles. En algunos niveles, el usuario puede ver que las estructuras de datos **Pandas** son una emulación de las proporcionadas por MATLAB. Esto se puede observar a través de la operación por columnas que se puede implementar en marcos de datos en **Pandas**. **Pandas** proporciona hojas de cálculo y lectores de archivos csv (Pandas, 2015).

Scikit-learn: es el módulo machine learning (ML) de Python que contiene valiosas herramientas para la implementación de algoritmos ML. Estos paquetes permiten a los no especialistas implementar los diferentes algoritmos de ML sin necesidad de profundizar en la codificación básica de los algoritmos. Las herramientas desarrolladas en esta biblioteca se diseñaron sobre **Numpy**, **Pandas** y **Matplotlib**. En general, **Scikit-learn** consta de seis secciones principales: clasificación, regresión, agrupamiento, reducción de dimensionalidad, selección de modelos y procesamiento de datos (Pedregosa et al., 2012).

OpenCV: Es una biblioteca de visión artificial de código abierto: <http://opencv.org> es una biblioteca de código abierto que incluye varios cientos de algoritmos de visión artificial (Bradski et al., 2000) .

- **Funcionalidad principal (core):** un módulo compacto que define estructuras de datos básicas, incluida la densa matriz multidimensional Mat y las funciones básicas utilizadas por todos los demás módulos.

- **Procesamiento de imágenes (imgproc):** un módulo de procesamiento de imágenes que incluye filtrado de imágenes lineales y no lineales, transformaciones de imágenes geométricas (cambio de tamaño, deformación afín y perspectiva, reasignación genérica basada en tablas), conversión de espacio de color, histogramas, etc.
- **Análisis de video (video):** un módulo de análisis de video que incluye estimación de movimiento, sustracción de fondo y algoritmos de seguimiento de objetos.
- **Calibración de cámara y reconstrucción 3D (calib3d):** algoritmos básicos de geometría de vista múltiple, calibración de cámara única y estéreo, estimación de pose de objeto, algoritmos de correspondencia estéreo y elementos de reconstrucción 3D.
- **Marco de funciones 2D (features2d):** detectores de funciones destacadas, descriptores y comparadores de descriptores.
- **Detección de objetos (objdetect):** detección de objetos e instancias de las clases predefinidas (por ejemplo, caras, ojos, tazas, personas, automóviles, etc.).
- **GUI de alto nivel (highgui):** una interfaz fácil de usar para capacidades de IU simples.
- **E/S de video (videoio):** una interfaz fácil de usar para la captura de video y los códecs de video.

CAPÍTULO III METODOLOGÍA

En este capítulo se plantea la metodología que se aplicó para el desarrollo del sistema de visión artificial, con el fin de describir los procesos y cumplir con los objetivos específicos de investigación. En la Figura 14 se muestra el desarrollo de actividades.



Figura 14. Fases para el cumplimiento de objetivos

A continuación, se presentaron las fases y actividades secuenciales para el desarrollo del sistema de visión artificial. Las fases se vincularon con cada uno de los objetivos específicos planteados.

3.1 Fase 1: Requerimiento de usuario

En esta fase de la investigación se definieron los requerimientos del sistema de visión artificial por medio de entrevistas a los encargados de venta de una distribuidora de hilos. Se analizaron las necesidades, expectativas obtenidas y se definieron los criterios de diseño utilizando la metodología de despliegue de la función de calidad (QFD).

Actividad 1: Entrevista; en esta actividad se determinaron los “¿Qué?” es decir, que quiere o necesita el usuario.

Actividad 2: Recopilación de requerimientos; se analizaron y clasificaron los “¿Qué?” del usuario.

Actividad 3: Determinación de los requerimientos; se describieron los criterios de diseño.

Actividad 4: Planteamiento del sistema; se establecieron los requisitos necesarios del sistema para que funcione de manera adecuada.

3.2 Fase 2: Diseño del sistema

En la segunda fase se propuso el diseño de un sistema de visión artificial mediante los criterios obtenidos de la matriz QFD. Se describieron los componentes físicos, algoritmo y demás elementos que componen el sistema.

Actividad 1: Planteamiento de alternativas; se propuso una solución de acuerdo con los requerimientos del usuario y del técnico.

Actividad 2: Análisis de componentes; se describieron los componentes del sistema de acuerdo con la mejor alternativa propuesta.

Actividad 3: Desarrollo de un algoritmo de visión artificial; se describió el funcionamiento del algoritmo que identifica colores utilizando librerías de visión artificial.

Actividad 4: Diseño de carcasa; se realizó el modelado e impresión 3D de una carcasa que sirve como base y soporte de los componentes a utilizar.

3.3 Fase 3: Implementación del sistema de visión artificial

En esta fase se implementaron todos los componentes del sistema para lo cual se describió el funcionamiento del componente de acuerdo con el proceso que ejecuta.

Actividad 1: Implementación del sistema embebido; se describió el sistema operativo, la instalación del lenguaje de programación y paqueterías de visión artificial.

Actividad 2: Conexión de cámara; se describió la cámara compatible con el sistema embebido a utilizar y se presentó su funcionalidad.

Actividad 3: Implementación del sistema de luminosidad; se implementaron luces al sistema de visión artificial.

3.4 Fase 4: Validar el funcionamiento del sistema

En la última fase, se desarrolló la validación del sistema aplicando la matriz de confusión, se analizaron los resultados de clasificación de color y se presentaron los resultados del sistema.

Actividad 1: Interfaz de usuario; se presentaron las acciones que realiza la interfaz de usuario y se describió su funcionamiento.

Actividad 2: Pruebas de funcionamiento; se validó el funcionamiento del sistema mediante la aplicación de una matriz de confusión.

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

Se realizó el análisis de cada una de las fases para el desarrollo del sistema de identificación de colores en tejidos de hilo con visión artificial mediante la determinación de los requerimientos del usuario, se desarrolló el algoritmo de visión artificial, se modeló una carcasa y se implementaron todos los componentes del sistema.

4.1 Requerimientos

Los requerimientos del cliente fueron recopilados en una matriz como se muestra en el Anexo 1. De acuerdo con las entrevistas realizadas a los encargados de venta de una distribuidora de hilos se realizó la matriz QFD para determinar los “¿Qué?” del usuario y los requerimientos del técnico.

4.1.1 Requerimientos del usuario

Mediante el análisis cualitativo realizado a las entrevistas se determinaron los requerimientos del usuario descrita en la siguiente Tabla 3.

Tabla 3. Requerimientos del usuario

Requerimientos	Descripción
Ligero	Que sea de bajo peso como una laptop
Móvil	Que se pueda transportar de un lado a otro
Pequeño	Que sea pequeña como una laptop
Buena precisión	Que sea preciso en identificar colores
Rápido procesamiento	Que tenga una rápida identificación
Fácil de manejar	Que se pueda utilizar fácil
Membretado	Que se indique el nombre y código del color

Se realizó un análisis estadístico para definir los pesos ponderados y los porcentajes de cada requerimiento. Cada requerimiento se evaluó en una escala de 1 al 10 con la participación de cada usuario como se muestra en la Tabla 4.

Tabla 4. Ponderación de requerimientos

Requerimiento	ID	Entrevistados					Total	Porcentaje	Peso Ponderado
		1	2	3	4	5			
Ligero	1	8	8	8	8	9	41	14.49%	8.20

Móvil	2	9	9	7	8	10	43	15.19%	8.00
Pequeño	3	9	8	8	7	8	40	14.13%	8.00
Buena precisión	4	8	7	7	8	8	38	13.43%	7.60
Rápido procesamiento	5	7	8	8	7	8	38	13.43%	7.60
Fácil de manejar	6	8	7	7	8	8	38	13.43%	7.60
Membretado	7	9	9	8	10	9	45	15.90%	9.00

Mediante el gráfico de **pareto de requisitos de calidad** mostrada en la Figura 15, se visualizó el requerimiento del membretado de los nombres y códigos de color como los requerimientos más importantes.

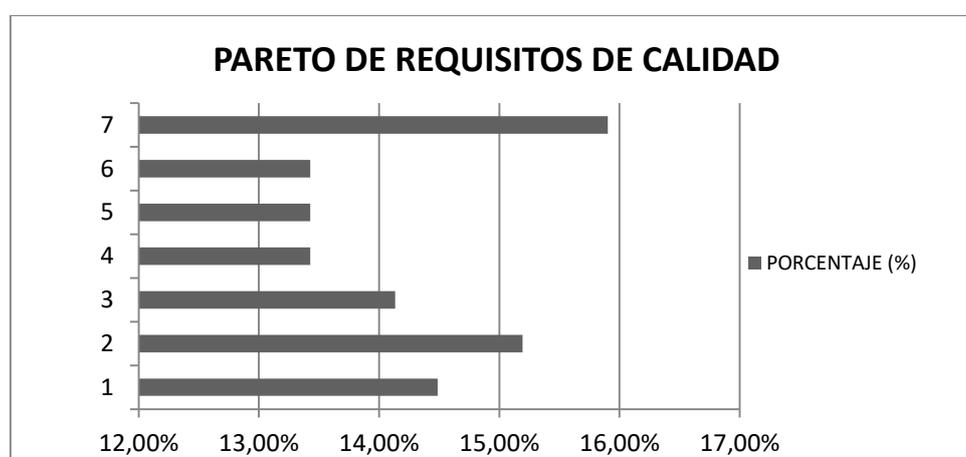


Figura 15. Pareto de requisitos de calidad

4.1.2 Requerimientos del técnico

Determinados los “Que” del usuario, se procedió a determinar los “Cómo” del técnico. La **Tabla 5** muestra los parámetros más ponderados mediante la realización de la matriz QFD mostrada en el Anexo 2.

Tabla 5. Requerimientos del técnico

Requerimientos	Descripción
Alto nivel	Debe tener un lenguaje de programación de alto nivel
Procesamiento	Capaz de procesar rápido
Calidad de imagen	Debe ser capaz de adquirir imágenes con buena definición
Luminosidad	Debe ser capaz iluminar el área de muestreo
Interfaz de usuario	Debe tener una interfaz de usuario
Material	Deber ser de un material ligero

4.1.3 Planteamiento de alternativas de solución

Una vez que se determinaron los requerimientos del usuario y se tomó en cuenta el cumplimiento del desarrollo del sistema, se propusieron las siguientes alternativas de solución que satisfacen las necesidades establecidas.

Alternativa 1: Se propuso la primera solución a los requerimientos obtenidos implementando un sistema de visión artificial en el lenguaje Python. Los componentes para lograr la funcionalidad de esta alternativa se presentan en la Tabla 6.

Tabla 6. Solución alternativa 1

Módulo	Criterio	Elemento
Software	Alto nivel	Python
Sistema embebido	Procesamiento	Raspberry
Cámara	Calidad de imagen	Raspberry
Iluminación	Luminosidad	Led
Interfaz de usuario	Interfaz gráfica	PyQt5
Impresión 3D	Material	PLA

Alternativa 2: Se propuso la segunda solución a los requerimientos obtenidos implementando un sistema de visión artificial en Matlab. Los elementos que representan el desarrollo del sistema se los detalla en la Tabla 7.

Tabla 7. Solución alternativa 2

Módulo	Criterio	Elemento
Software	Alto nivel	Matlab
Sistema embebido	Procesamiento	Arduino
Cámara	Calidad de imagen	Celular
Iluminación	Luminosidad	Incandescente
Interfaz de usuario	Interfaz gráfica	Matlab
Impresión 3D	Material	PETG

4.1.4 Ponderación de criterios

Para realizar la ponderación de criterios se determinaron los criterios más relevantes por parte del usuario y el técnico, los cuales se encuentran detallados en la Tabla 8 partiendo de la Tabla 5.

Tabla 8. Nomenclatura de criterios

Nomenclatura	Criterio
C1	Alto nivel
C2	Procesamiento
C3	Calidad de imagen
C4	Luminosidad
C5	Interfaz gráfica
C6	Material

A continuación, se realizó la matriz de criterios la cual muestra el grado de importancia que tiene cada criterio respecto a los requerimientos del sistema. Primero se realizó la comparación de criterios y así se determinó el orden de importancia como se muestra en la Tabla 9.

Tabla 9. Comparación de criterios

	C1	C2	C3	C4	C5	C6	$\sum +1$	Ponderación
C1	-	1	1	1	1	1	6	0.29
C2	0	-	1	1	1	1	5	0.24
C3	0	0	-	0.5	0	0.5	2	0.10
C4	0	0	0.5	-	0	0.5	2	0.10
C5	0	0	1	1	-	1	4	0.19
C6	0	0	0.5	0.5	0	-	2	0.10
	Total						21	1

4.1.5 Ponderación de alternativas con cada criterio

En las siguientes tablas se realizó la comparación de las alternativas de solución con respecto a cada criterio. Se presentó en la Tabla 10 el criterio para desarrollar un sistema con un lenguaje de alto nivel.

Tabla 10. Comparación criterio de alto nivel

Alto nivel	Alternativa 1	Alternativa 2	$\sum +1$	Ponderación
Alternativa 1	-	0.5	1.5	0.50
Alternativa 2	0.5	-	1.5	0.50
	Total		3	1

En la Tabla 11, se realizó la comparación de las alternativas con el criterio para tener un mejor procesamiento del sistema.

Tabla 11. Comparación criterio de procesamiento

Procesamiento	Alternativa 1	Alternativa 2	$\sum +1$	Ponderación
---------------	---------------	---------------	-----------	-------------

Alternativa 1	-	0,5	2	0.50
Alternativa 2	0,5	-	1	0.50
Total			3	1

En la Tabla 12, se realizó la comparación de las alternativas con el criterio para obtener una buena calidad de imagen.

Tabla 12. Comparación criterio de calidad de imagen

Calidad de imagen	Alternativa 1	Alternativa 2	$\sum +1$	Ponderación
Alternativa 1	-	0	2	0.34
Alternativa 2	1	-	1	0.66
Total			3	1

En la Tabla 13, se realizó la comparación de las alternativas con el criterio para obtener una buena luminosidad.

Tabla 13. Comparación criterio de luminosidad

Luminosidad	Alternativa 1	Alternativa 2	$\sum +1$	Ponderación
Alternativa 1	-	1	1.5	0.66
Alternativa 2	0	-	1.5	0.34
Total			3	1

En la Tabla 14, se realizó la comparación de las alternativas con el criterio para tener una interfaz de usuario de fácil uso.

Tabla 14. Comparación criterio de interfaz de usuario

Interfaz de usuario	Alternativa 1	Alternativa 2	$\sum +1$	Ponderación
Alternativa 1	-	1	1.5	0.5
Alternativa 2	1	-	1.5	0.5
Total			3	1

En la Tabla 15, se realizó la comparación de las alternativas con el criterio para obtener un mejor material.

Tabla 15. Comparación criterio de material

Material	Alternativa 1	Alternativa 2	$\sum +1$	Ponderación
Alternativa 1	-	1	1.5	0.66
Alternativa 2	0	-	1.5	0.34

Total	3	1
-------	---	---

Realizadas las matrices de criterios, se procedió al análisis de cada ponderación en cada criterio según el análisis realizado con cada alternativa. Se estableció en la Tabla 16, la multiplicación de la ponderación de cada criterio con la ponderación de cada alternativa dando como resultado la ponderación de cada alternativa.

Tabla 16. Resultados de la ponderación de criterios

	C1	C2	C3	C4	C5	C6	Ponderación
Alternativa 1	0.29*.05	0.24*.5	0.10*0.34	0.10*0.66	0.19*0.5	0.10*0.66	0,52
Alternativa 2	0.29*0.5	0.24*.5	0.10*0.66	0.10*0.34	0.19*0.5	0.10*0.34	0,48

El resultado del análisis dio como mejor solución la alternativa 1, la cual representó en un 52% el cumplimiento de los requerimientos determinados en las diferentes ponderaciones.

4.1.6 Análisis de componentes

En la Tabla 17, se realizó la lista de las ventajas y desventajas de utilizar los componentes de la alternativa 1. Se describió el lenguaje de programación, sistema embebido, cámara, luces, interfaz de usuario y material para impresión 3D.

Tabla 17. Descripción de componentes

Componente	Ventajas	Desventajas
Lenguaje de programación Python	<ul style="list-style-type: none"> • Simplificado y rápido • Elegante y flexible • Programación sana y productiva • Ordenado y limpio • Portable • Comunidad 	<ul style="list-style-type: none"> • Curva de aprendizaje • Hosting • Librerías incluidas
Sistema embebido Raspberry	<ul style="list-style-type: none"> • Micro USB de 5V y 1,2 A • CPU • HDMI • Wifi • Ethernet • Sistema operativo • Tamaño pequeño • Pines de E/S • Entradas para periféricos 	<ul style="list-style-type: none"> • Capacidad de cómputo reducido • Consumo de energía • Compleja instalación
Cámara Raspberry	<ul style="list-style-type: none"> • 5 megapíxeles • 1080 p y 30 fps • Reducido tamaño • Robustez 	<ul style="list-style-type: none"> • Conexión por puerto CSI

Leds	<ul style="list-style-type: none"> • Ahorro de energía • Mayor tiempo útil • Cantidad de luz y distancia • Forma direccionada • Se puede integrar a controladores 	<ul style="list-style-type: none"> • Mayor precio • Mayor temperatura • Capacidad de regulación complicada
Interfaz gráfica PyQt5	<ul style="list-style-type: none"> • Flexibilidad de codificación • Amplia gama de APIs • Varios widgets • Varios recursos de documentación • Fácil de dominar 	<ul style="list-style-type: none"> • Falta de documentación específica de Python para las clases de PyQt5 • Curva de aprendizaje empinada
Filamento PLA	<ul style="list-style-type: none"> • Dureza: Alta • Flexibilidad: Baja • Durabilidad: Media • Dificultad de uso: Baja • Temperatura de impresión: 180°C – 230°C • Temperatura de la cama de impresión: 20°C – 60°C • Contracción/deformación: Mínima • Soluble: No 	<ul style="list-style-type: none"> • Quebradizo • Propiedades mecánicas bajas

4.2 Diseño del sistema

Para el desarrollo del sistema se tomaron en cuenta todos los elementos que hacen que el sistema de visión artificial cumpla con la solución de los requerimientos desde la parte física de diseño hasta la parte del algoritmo. En la Figura 16 se muestran los elementos que componen el sistema de visión artificial para la identificación de colores en tejidos con hilo.

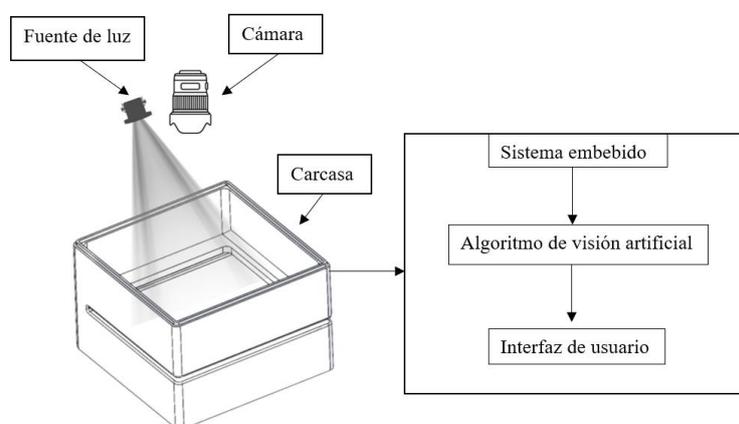


Figura 16. Sistema de visión artificial

4.2.1 Desarrollo de algoritmo de visión artificial

El desarrollo de un algoritmo de visión artificial tiene distintas maneras de ser programado y se suelen utilizar diferentes herramientas, lenguajes y librerías para lograr la mejor programación. Este sistema de visión artificial utilizó el lenguaje de programación Python ya que es un lenguaje de alto nivel, de fácil lectura y por tanto de diseño sencillo y rápido. En la Figura 17 se muestra el diagrama de flujo del algoritmo.

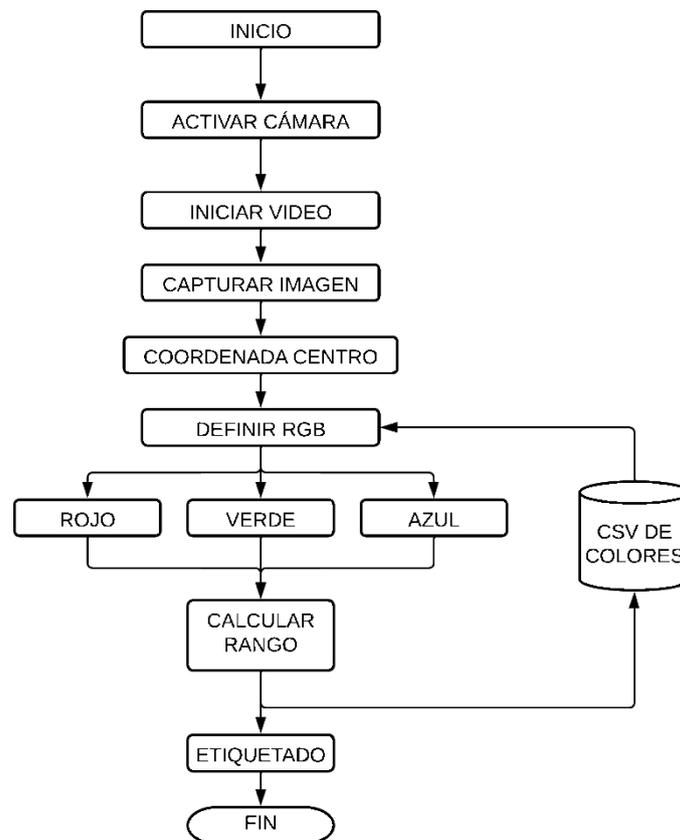


Figura 17. Diagrama de flujo

Fue necesario instalar el editor de código Visual Studio Code, este editor se puede ejecutar en Linux, el cual es el sistema operativo que tiene el sistema embebido Raspberry, es compatible con Python y todas las librerías de visión artificial. Definido el lenguaje de programación, editor de código y librerías de visión artificial se procedió a desarrollar el algoritmo de visión artificial que cumpla con identificar el color en tejidos de hilos. Para lo cual se dividió el código en procesos.

- Adquisición de la imagen
 - a) Selección de cámara
 - b) Lectura de video en tiempo real
 - c) Captura de imagen del video
 - d) Almacenamiento de la imagen capturada

- Lectura del archivo de colores .csv
 - a) Definir los encabezados de las columnas del archivo
 - b) Calcular distancia mínima de todos los colores RGB
 - c) Obtener los valores absolutos de RGB del archivo
 - d) Obtener el color más coincidente en RGB

- Obtener las coordenadas RGB de la imagen
 - a) Definir las coordenadas en la imagen
 - b) Definir una entrada de acción (mouse)
 - c) Crear variables RGB
 - d) Obtener el RGB de la coordenada de la imagen

- Etiquetar el código de color y el nombre
 - a) Crear un rectángulo de fondo
 - b) Crear el texto del nombre y código
 - c) Definir la fuente y escala
 - d) Mostrar el texto con color negro o blanco

Descritos los procesos del algoritmo de identificación de colores como se muestra en la Figura 18, se procedió a realizar una prueba del funcionamiento del algoritmo con las siguientes condiciones:

- Iluminación de un foco incandescente de 100W

- Ambiente abierto
- Cámara de laptop Asus
- Cono de hilo con medida de 15 cm x 5 cm color burdeos.
- Punto de interés seleccionado con un clic

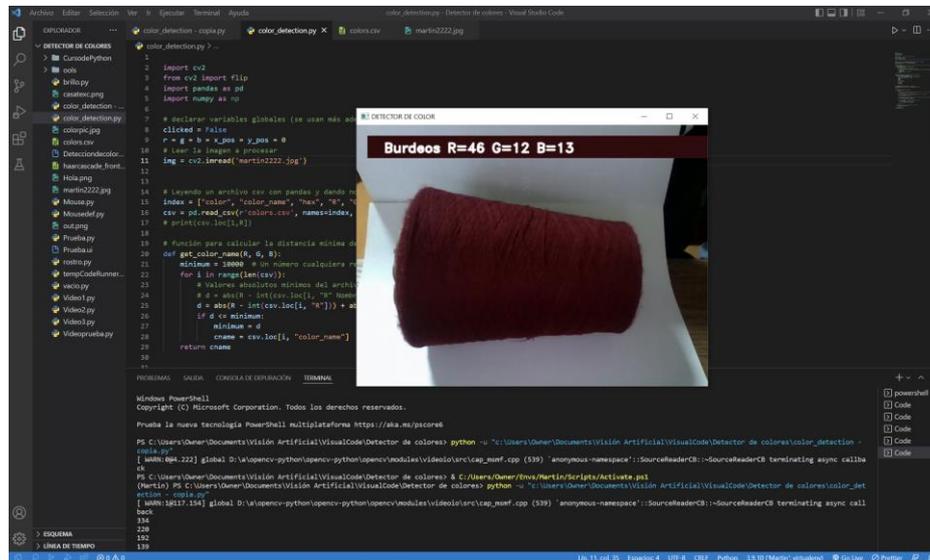


Figura 18. Ejemplo de algoritmo sin condiciones de ambiente controlado

El algoritmo cumplió la función de identificar el color del hilo en un punto de interés. En esta prueba a pesar de identificar el color y a su vez realizar el membretado con el nombre del color y el código RGB, se logró ver las deficiencias que trae realizar un algoritmo de visión artificial sin una buena adquisición de imagen, iluminación, cámara y ambiente. El algoritmo aun cuando sea el mejor es complementado por las características mencionadas en la solución del sistema Figura, para tener un mejor resultado.

4.2.2 Diseño de carcasa

La estructura del sistema fue una carcasa, en donde se pudo implementar y proteger los sistemas de adquisición de imagen, iluminación y el sistema operativo. En la Figura 19, se muestra la representación del modelado de la carcasa.

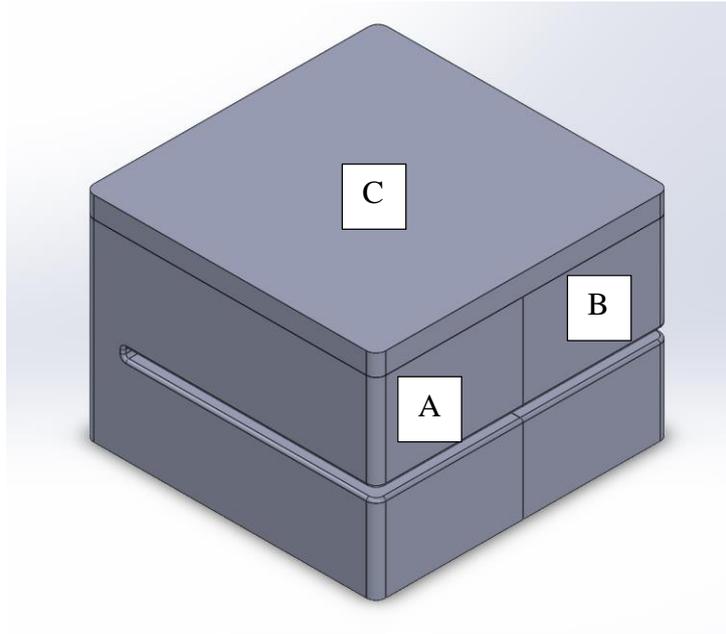


Figura 19. Estructura de carcasa

La carcasa contó con tres piezas, las cuales fueron seccionadas estratégicamente para una fácil impresión en 3D, tomando en cuenta voladizos, agujeros y dimensiones que son características importantes al momento de pasar un CAD a un STL para su impresión en 3D. A continuación, se presentaron las descripciones de cada una de las piezas.

Pieza A. – La principal característica de esta pieza fueron las extrusiones laterales y superiores para el ensamble con las demás piezas como se muestra en la Figura 20.

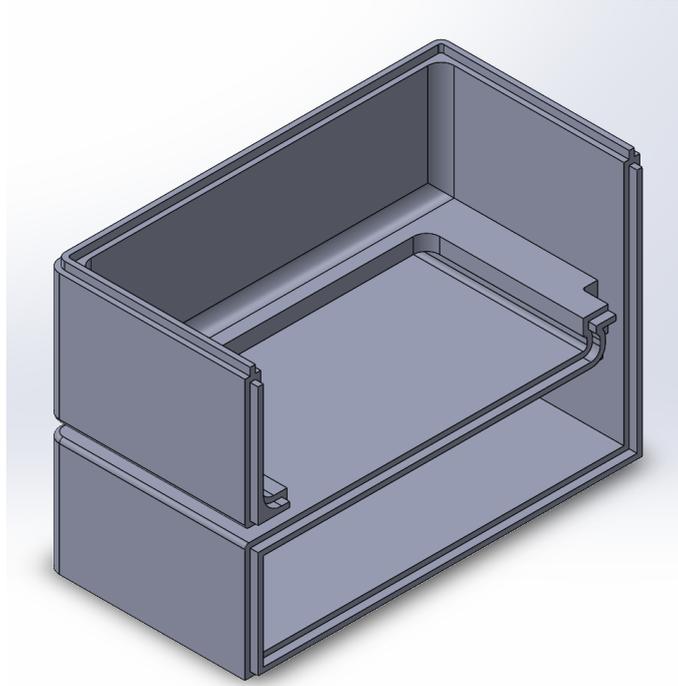


Figura 20. Modelado pieza A

Pieza B. – Esta pieza contó con una extrusión superior igual a la anterior Figura 20. Por medio de las extrusiones de corte en las paredes laterales se accederá a los periféricos del sistema embebido como se ve en Figura 21.

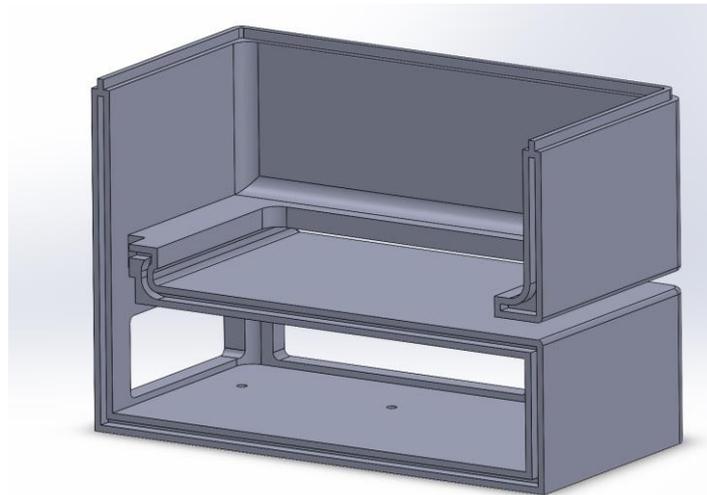


Figura 21. Modelado pieza B

Pieza C. – La tapa de la carcasa tiene extrusiones de corte en donde se implementó la cámara para la toma de imágenes y las luces para la iluminación del área de trabajo como se observa en la Figura 22.

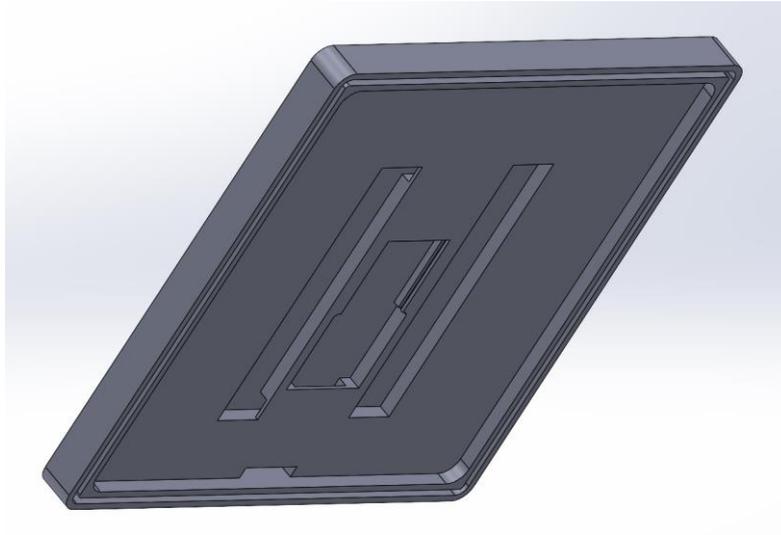


Figura 22. Modelado pieza C

El sistema de carcasa final protege a todos los periféricos implementados como lo es el sistema embebido para el cual se utilizó un STL de la carcasa protectora de la raspberry pi 4. Además, se implementó un agujero para la inserción de la fuente de alimentación del sistema de luminosidad como se muestra en la Figura 23.

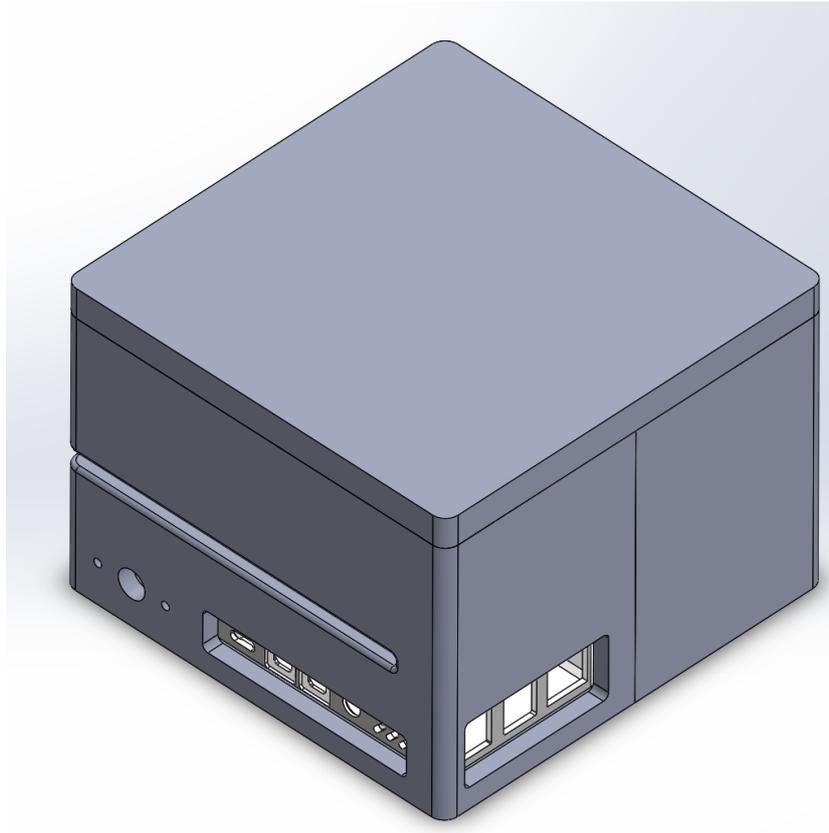


Figura 23. Carcasa de Raspberry pi4

4.2.3 Impresión 3D de carcasa

La estructura de la carcasa fue impresa en 3D, la cual protegerá y soportará a los componentes del sistema como son la cámara, luces led, sistema embebido y la fuente de alimentación. Además, la estructura contiene la entrada donde se introducirán los tejidos con hilo para la detección del color. Para la selección del filamento se tomó en cuenta los siguientes parámetros de la Tabla 18.

Tabla 18. Características de filamentos

Filamento	Dureza	Flexibilidad	Temperatura	Soluble
ABS	Alta	Media	210 °C – 250 °C	Ésteres, cetonas y acetona
PLA	Alta	Baja	180 °C – 230 °C	No
PET	Alta	Media	220 °C – 250 °C	No
PC	Muy alta	Media	270 °C – 310 °C	No

A continuación, se presentó la ficha técnica del PLA, realizada mediante diversos métodos de ensayo como se observa en la Tabla 19. Con las propiedades del PLA se pueden determinar los parámetros de impresión, los cuales se establecieron en el software de PRUSA.

Tabla 19. Propiedades mecánicas del PLA

Propiedades mecánicas	Filamento PLA	Método
Módulo de elasticidad a la tracción	2346,5 MPa	ISO 527
Esfuerzo de tracción a la deformación	49,5 MPa	ISO 527
Esfuerzo de tracción a la rotura	45,6 MPa	ISO 527
Alargamiento a la deformación	3,3 %	ISO 527
Alargamiento a la rotura	5,2 %	ISO 527
Resistencia a la flexión	103,0 MPa	ISO 178
Módulo de flexión	3150,0 MPa	ISO 178
Resistencia a la prueba de impacto Izod, con mella (a 23 °C)	5,1 kJ/m ²	ISO 178
Dureza	83 (Shore D)	Durómetro

Fuente: (Ultimaker, 2017)

Las impresiones fueron realizadas con un grosor de boquilla de 0,20 mm, altura de capa de 0,20 mm, relleno de 50% con patrón triangular y una balsa de 6 mm por las dimensiones grandes de las piezas. Además de activar el escudo de protección para que la falda sea tan grande como el objeto impreso.

- La pieza A tardo en imprimirse 23h58m, usando 927,56 gramos de filamento representando 93,40 metros como se muestra en la Figura 24.

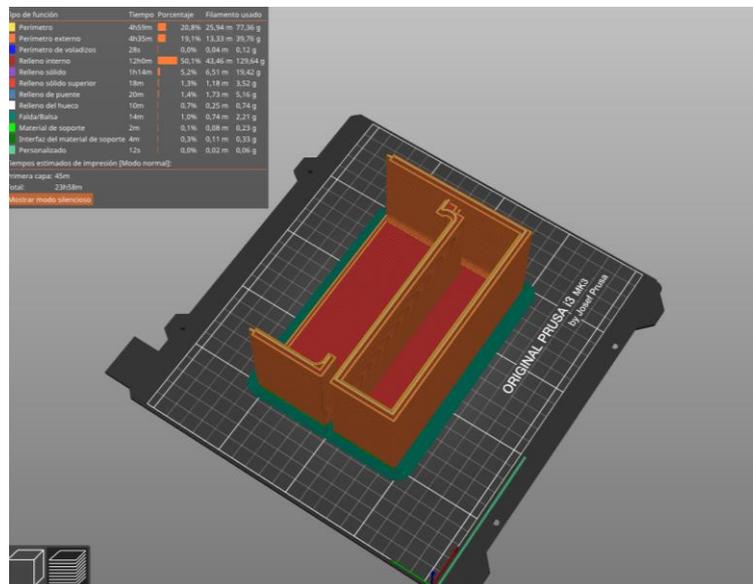


Figura 24. Impresión 3D pieza A

- La pieza B tardo en imprimirse 23h11m, usando 963,37 gramos de filamento representando 93,40 metros como se ve en la Figura 25.

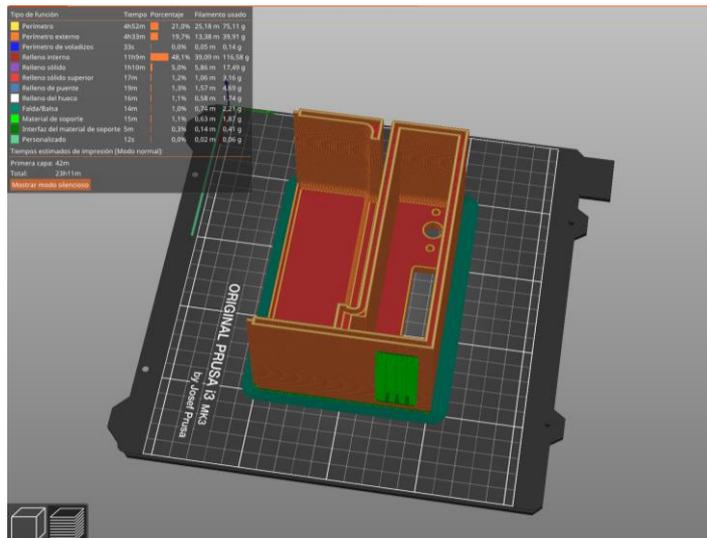


Figura 25. Modelado pieza B

- La pieza C tardo en imprimirse 7h40m, usando 131,94 gramos de filamento representando 44,24 metros como se ve en la Figura 26.

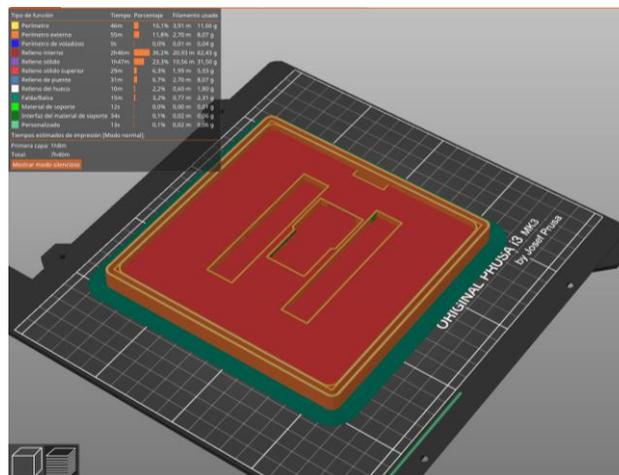


Figura 26. Modelado pieza C

En total la impresión de las piezas tardó 54h49m y se gastó aproximadamente 2kg de filamento que representa dos rollos de filamento PLA. En la Figura 27 se muestra la impresión en 3D del diseño de la carcasa del sistema.

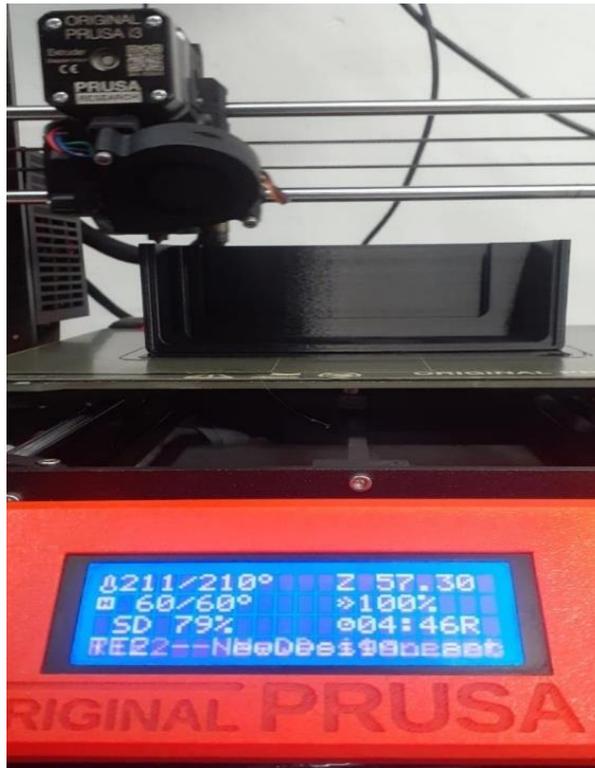


Figura 27. Impresora Prusa

4.2.4 Implementación del sistema embebido

Se implementó el sistema embebido raspberry pi 4 ya que es un ordenador de placa reducida y a comparación de otros ordenadores es de bajo coste. La conexión de los componentes se muestra en la Figura 28.

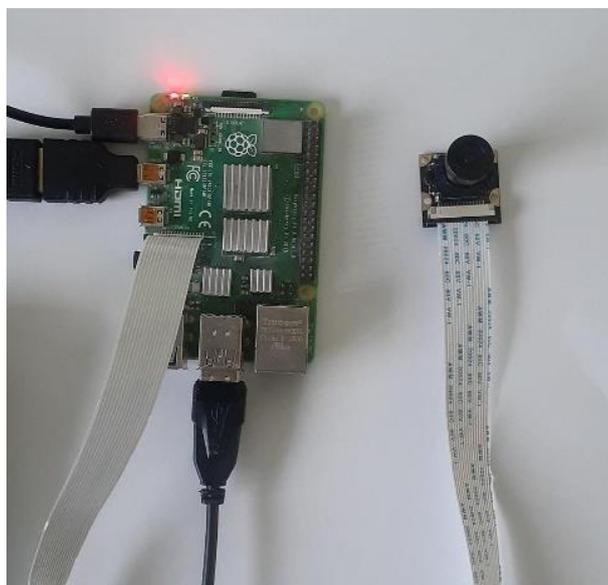


Figura 28. Sistema embebido

Los componentes conectados fueron:

- Fuente de alimentación 5V – 3A
- Tarjeta SD
- HDMI
- Cámara
- Teclado
- Mouse

4.2.5 Conexión de cámara

La implementación de la cámara en el sistema de visión artificial se realizó tomando en cuenta la compatibilidad de esta con el sistema embebido utilizado. Para ello se tomó en cuenta las siguientes características descritas en la Tabla 20.

Tabla 20. Características de la cámara

Característica	Especificación
Módulo de placa	Omnivision 5647 de 5 MP
Lente	Arducam M12
Enfoque	Manual
Conexión	Bus CSI
Compatibilidad	Raspberry Pi modelo B / B + / A + / 3/2/1
Resolución de imágenes fijas	2592x1944
Ángulo de visión	70 °
Velocidad de cuadro	30 fps@1080p
Distancia focal	2.93 cm (enfoque ajustable)
Apertura	1.8 F
Dimensiones	3.58 x 3.58 cm

La óptica de la cámara fue determinada por el lente, este transmite la luz al sensor de la cámara de una manera controlada para obtener una imagen enfocada. Se tomó en cuenta los siguientes parámetros para definir si el lente cumple con una buena distancia focal:

- a) Tamaño del sensor de la cámara
- b) Distancia de la cámara con el objeto
- c) Campo de visión

Una vez conocido los parámetros se aplicó la fórmula para la selección correcta de la distancia focal.

$$f = \frac{b * D}{B} * c$$

Donde:

b = Tamaño del sensor 1/3" (8,5mm)

B = Anchura del objeto 20 mm

f = Distancia focal

D = Distancia de trabajo

c= Factor de conversión del tamaño del sensor = 0.72

$$f = \frac{8.5 * 112}{30} * 0.72 = 22.84 \text{ mm}$$

Realizado el análisis de la distancia focal y de acuerdo con las características descritas en la **¡Error! No se encuentra el origen de la referencia.**, la cámara cumple con los requisitos necesarios para este trabajo.

4.2.6 Implementación del sistema de luminosidad

Para la implementación de la luz se tomó en cuenta el estudio de Albert & Antiga (2012), el cual menciona que existen diferentes técnicas para resaltar aspectos importantes de los objetos. Una de ellas es la iluminación frontal, consiste en poner la cámara en la misma dirección que la luz como se muestra en la Figura 29. Esta técnica suaviza las texturas, reduce sombras y minimiza la generación de rayas.

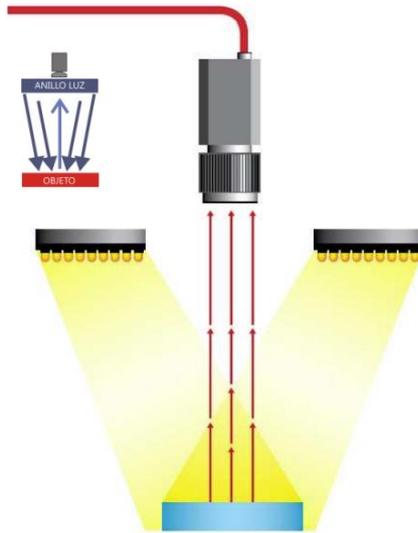


Figura 29. Técnica frontal de iluminación

Fuente: (Albert & Antiga, 2012a)

Aplicación: Las luces se ubicaron alrededor de la cámara en la parte superior de la carcasa de manera que tengan la misma dirección hacia el objeto. Este sistema de obtención de imagen es apta para superficies como la tela o los hilos.

Ventajas: Esta técnica ayuda para la detección de diferentes colores y caracteres.

Desventajas: Se obtienen reflejos sobre superficies reflectantes.

Se tomó en cuenta la preparación de las condiciones de luz del estudio de Pasuy (2019), se muestra en la Figura 30, el sistema de iluminación desarrollado en donde se observa una cinta de luces led de color blanco y difusores en cada led para la distribución homogénea de la luz, además de los cables de conexión para un voltaje de 3V y 1A de corriente.

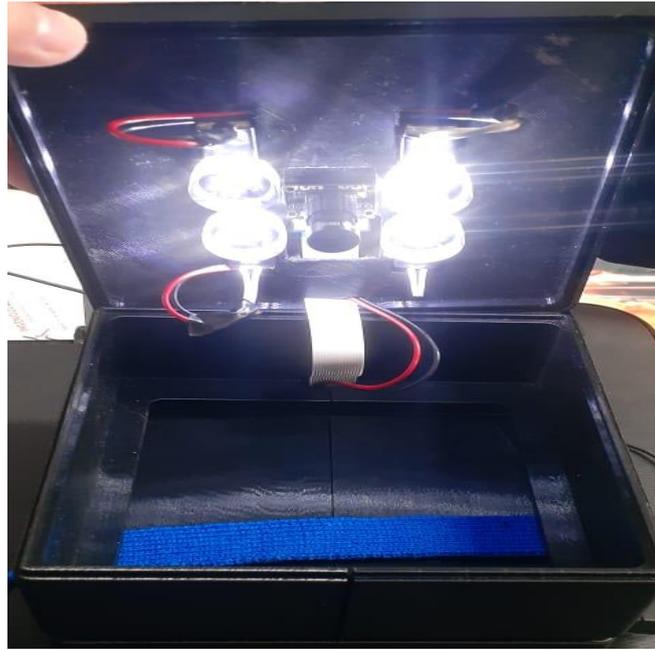


Figura 30. Sistema de iluminación

4.3 Validación de la funcionalidad del sistema

En esta sección se validó el funcionamiento del sistema de visión artificial de identificación de colores en tejidos con hilo, se muestra en la Figura 31 el proceso de funcionamiento del sistema. Se introdujo en la carcasa la muestra del tejido con hilo y se procedió a la identificación del color por medio de una interfaz de usuario.



Figura 31. Funcionamiento del sistema

La interfaz de usuario que se realizó se muestra en la Figura 32, está interfaz es el medio de comunicación entre las personas que usarán el sistema y el sistema de visión artificial sin necesidad de profundizar en la ejecución del algoritmo.



Figura 32. Interfaz de usuario

La interfaz de usuario se mostró automáticamente al encender la Raspberry como se muestra en la Figura 33 mediante la configuración de AutoStart de Raspberry. Esta configuración permitió ejecutar el código de programación de una manera automática sin la necesidad de ingresar al editor de código.

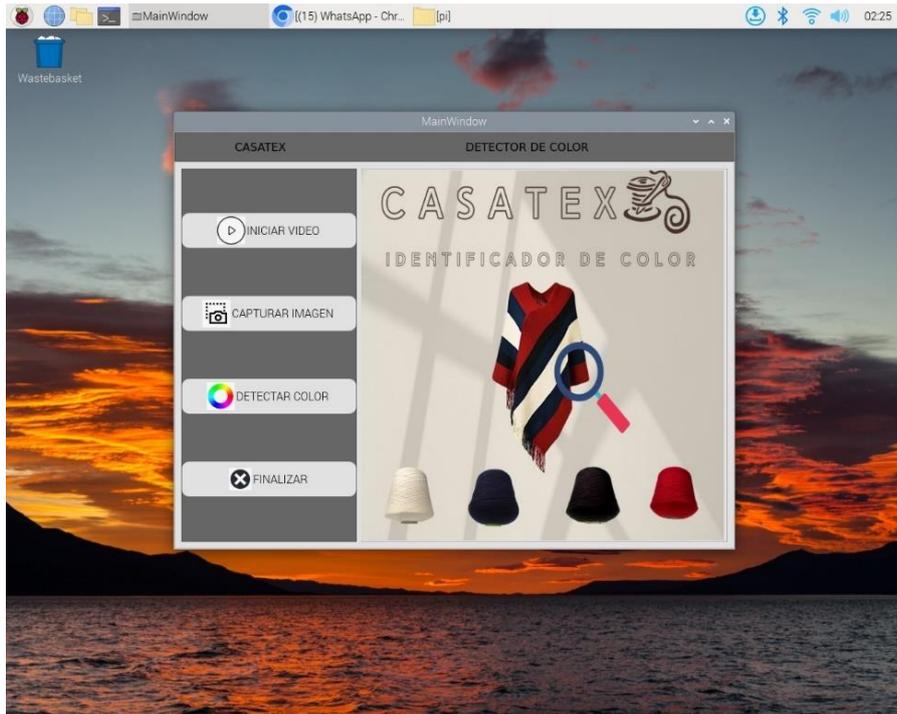


Figura 33. Configuración AutoStart de Raspberry

La finalidad de la interfaz de usuario es mostrar de una manera fácil el código de programación y que pueda ser usada por todos los usuarios. El diseño contó con botones de **inicio de video**, **captura de imagen**, **detección de color** y **finalizar**. Además de un marco de trabajo para mostrar las imágenes de la cámara y una ventana emergente para la detección del color.

- **Botón iniciar video.** – Los botones fueron etiquetados con nombres similares a las funciones que cumplen. Se muestra en la Figura 34 el botón de encendido de la cámara y mostró en tiempo real el video del área de detección.

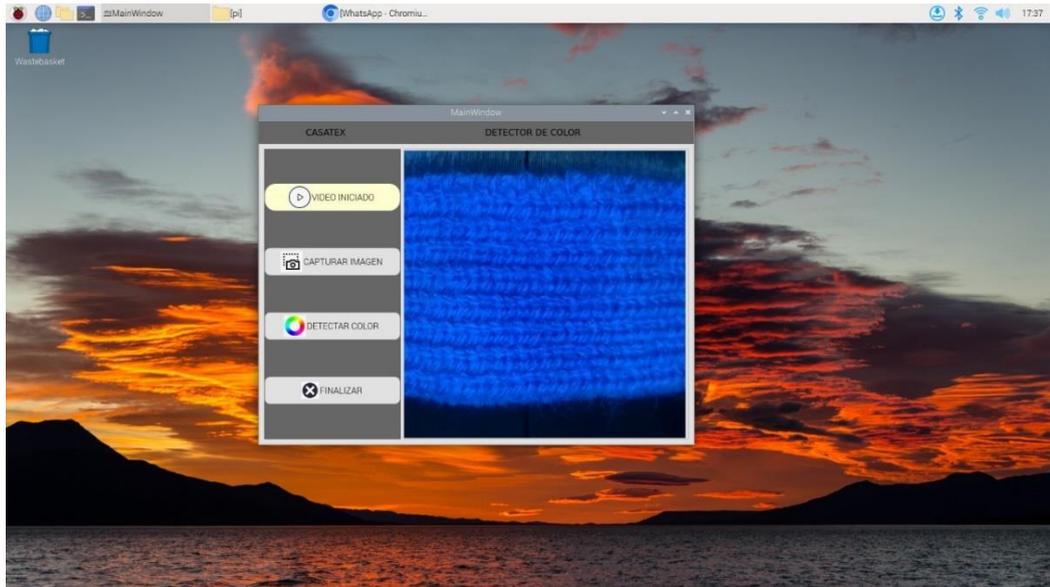


Figura 34. Botón iniciar video

- **Botón capturar imagen.** – Este botón capturó un fotograma del video transmitido por la cámara como se muestra en la Figura 35. Esta captura fue guardada con el nombre de imagen analizada y la fecha en que se realizó la captura para un registro de muestras analizadas.

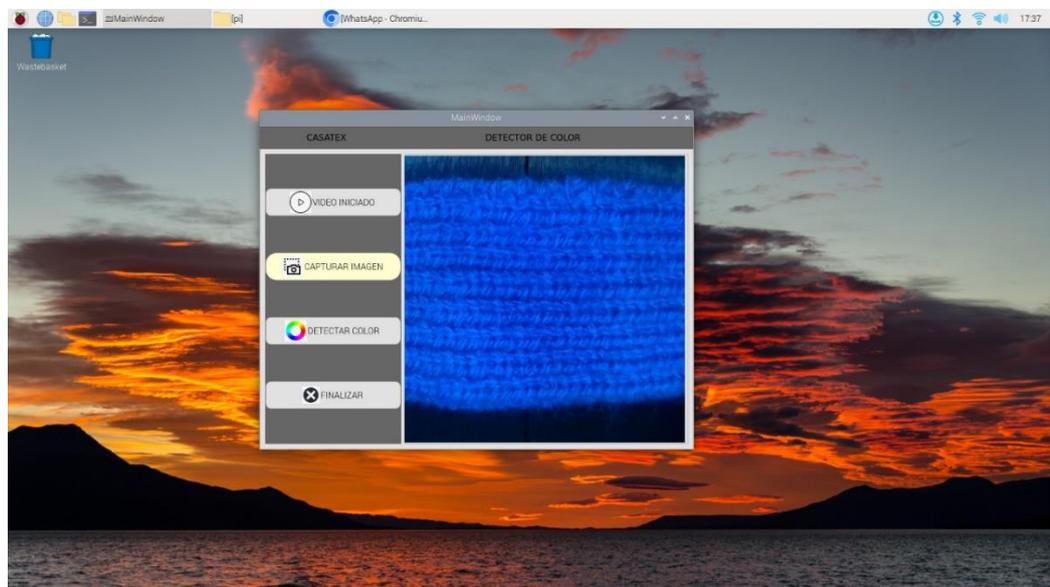


Figura 35. Botón capturar imagen

- **Botón detectar color.** – Una vez se guardó la captura del fotograma, se procedió a realizar la detección de color del área de interés. Este botón abrió una nueva ventana

en donde se pudo realizar la detección de color mediante un clic en la zona de interés como se ve en la Figura 36.

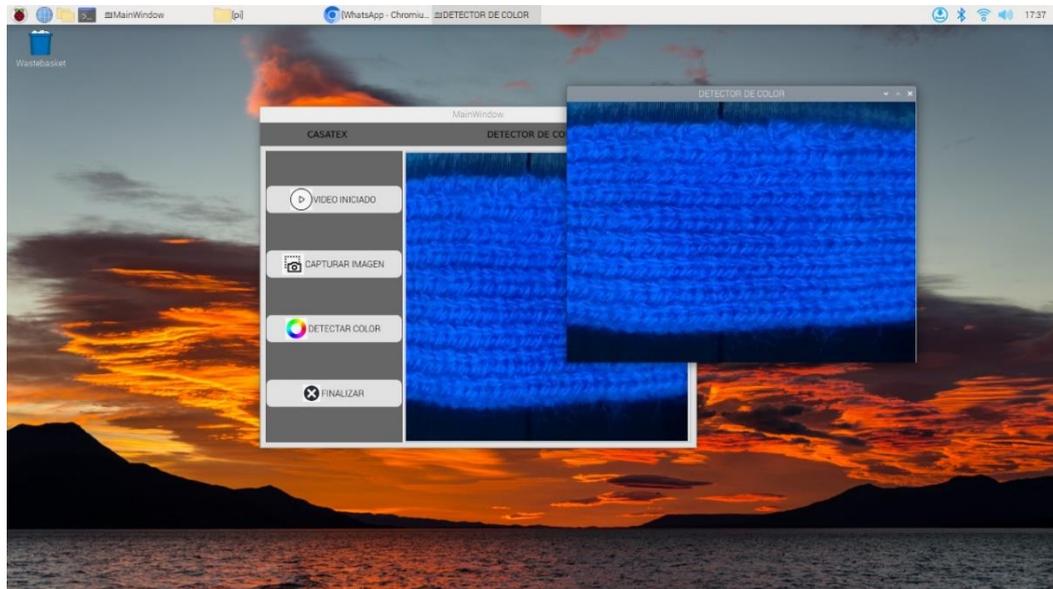


Figura 36. Botón detectar color

- **Ventana Detector de color.** – En esta ventana se pudo realizar la detección del color mediante un clic en la zona de interés. En la ventana se mostró el texto resultado en una franja rectangular de color blanco como se observa en la Figura 37.

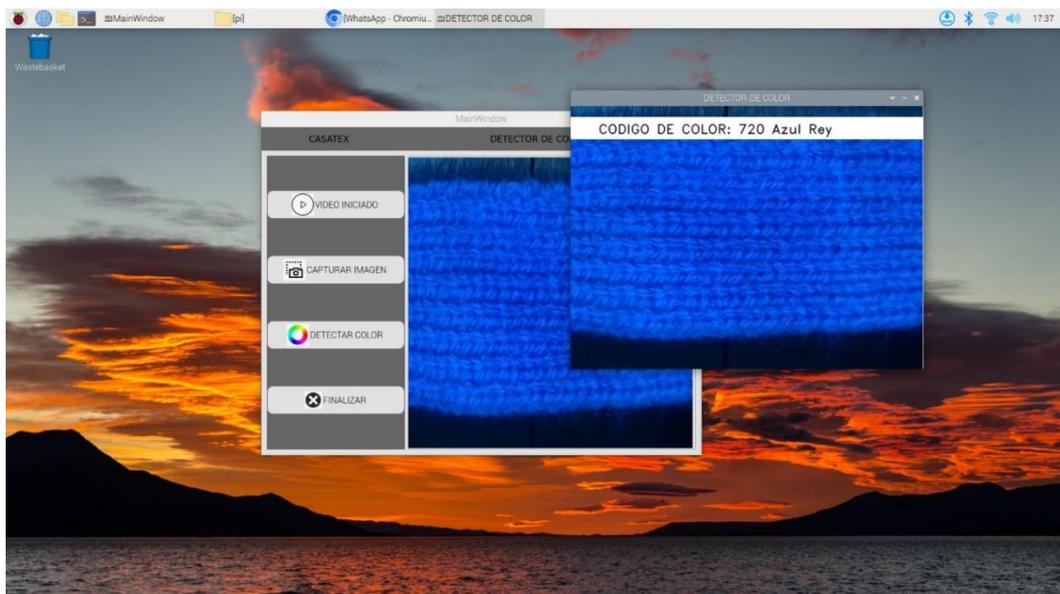


Figura 37. Ventana Detector de color

- Botón finalizar. – En la anterior Figura 37, **¡Error! No se encuentra el origen de la referencia.** se observa el botón finalizar que cierra la interfaz de usuario totalmente.

4.3.1 Matriz de confusión

Este trabajo fue validado realizando el análisis de una matriz de confusión multiclase, donde se asignó un color a cada clase como se muestra en la Figura 38. Las clases definidas fueron: **azul, negro, rojo, rosado, verde, blanco, amarillo.** La frecuencia de observaciones fue de 10 por cada color en total se realizó 70 observaciones. En la matriz se presentan los valores reales en las filas y valores predictivos en las columnas. A continuación, se describen los porcentajes de las observaciones de cada clase.

- **Azul:** Se mostró al sistema 10 observaciones de valor real azul y detectó 10 observaciones del valor predictivo azul, pero existió un valor predictivo azul detectado como un valor real verde. Esto significa que en un 100% se detectó correctamente el color azul, pero existió un 10% de colores que fueron detectados erróneamente como color azul.
- **Negro:** Se mostró al sistema 10 observaciones de valor real negro y detectó 10 observaciones del valor predictivo negro. Esto significa que se identificó en un 100% correctamente el color negro y en un 0% que otro color sea identificado como negro.
- **Rojo:** Se mostró al sistema 10 observaciones de valor real rojo y detectó 10 observaciones del valor predictivo rojo, pero existió un valor predictivo rojo detectado como un valor real rosado. Esto quiere decir que se identificó correctamente en un 100% el color rojo, pero en un 10% otro color se identificó como rojo.
- **Rosado:** Se mostró al sistema 10 observaciones de valor real rosado y detectó 8 observaciones del valor predictivo rosado, pero existieron 2 valores reales rosado que fueron detectados como valores predictivos rojo y blanco. Esto quiere decir que hubo un acierto del 80% del color rosado pero un 20% del color rosado se identificó como otro color.
- **Verde:** Se mostró al sistema 10 observaciones de valor real verde y detectó 6 observaciones del valor predictivo verde, pero existieron 4 valores reales verdes que fueron detectados como valores predictivos azul y blanco. Además, se presentó la detección de un valor real amarillo como valor predictivo verde. Esto significa que en

un 60% se identificó correctamente el color verde, en un 40% el color verde fue identificado como otro color y que un 10% de otro color fue identificado erróneamente color rosado.

- **Blanco:** Se mostró al sistema 10 observaciones de valor real blanco y detectó 10 observaciones del valor predictivo blanco, pero existieron 6 valores predictivos detectados como blanco que fueron valores reales verde, amarillo y rosado. Esto significa que se identificó correctamente el color blanco en un 100% y que un 60% de otros colores fueron identificados erróneamente como color blanco.

- **Amarillo:** Se mostró al sistema 10 observaciones de valor real amarillo y detectó 7 observaciones del valor predictivo amarillo, pero existieron 3 valores reales amarillo que fueron detectados como valores predictivos blanco y verde. Esto significa que se identificó correctamente el color amarillo en un 70% y que en un 30% otro color se identificó erróneamente como color amarillo.

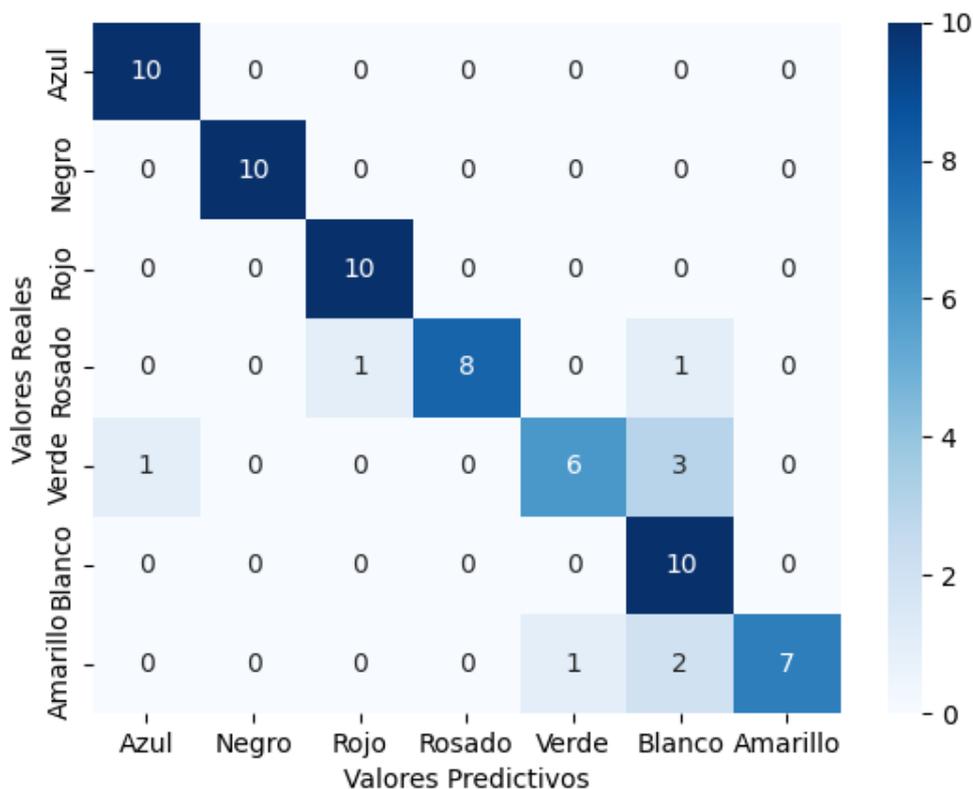


Figura 38. Matriz de confusión

La matriz resultante fue de 7x7 celdas, en la cual se realizó el análisis de los parámetros de verdadero positivo (TP), verdadero negativo (TN), falso positivo (FP) y falso negativo (FN).

En donde TP significó que el valor real y el valor pronosticado son positivos, TN significó que el valor real es positivo pero el valor pronosticado por el modelo es negativo, FP significó que el valor real es negativo pero el valor pronosticado por el modelo es positivo y, finalmente, FN significó ambos valores real y pronosticado son negativos como se ve en la matriz de observación en la Tabla 21.

Tabla 21. Matriz de observación

CLASES	Medidas				
	TP	TN	FP	FN	
Azul	10	59	1	0	
Negro	10	60	0	0	
Rojo	10	59	1	0	
Rosado	8	60	0	2	
Verde	6	59	1	4	
Blanco	10	54	6	0	
Amarillo	7	60	0	3	

De la matriz de confusión se pudo recolectar información sobre la frecuencia con la que se detecta correctamente un determinado comportamiento y la frecuencia con la que se clasifica como otro comportamiento. La exactitud de la clasificación suele resumirse mediante indicadores de rendimiento como la precisión, sensibilidad, especificidad (Ruuska et al., 2018).

- **La sensibilidad** es la capacidad de un clasificador para seleccionar todos los casos que necesitan ser seleccionados. Un clasificador perfecto no tendrá FN. La sensibilidad se expresa como una relación (o porcentaje) calculada de la siguiente manera:

$$Sensibilidad = \frac{TP}{TP + FN}$$

- **La especificidad** es la capacidad de un clasificador para rechazar todos los casos que necesitan ser rechazados. Un clasificador perfecto no tendrá FP. La especificidad se expresa como una relación (o porcentaje) calculada como:

$$Especificidad = \frac{TN}{TN + FP}$$

- **La precisión** se define como la proporción de casos encontrados que eran realmente relevantes. Por lo tanto, la precisión es:

$$Precisión = \frac{TP}{TP + FP}$$

Se realizó el análisis cuantitativo de la **precisión, sensibilidad, especificidad** para validar el sistema de visión artificial para la detección de colores. Los resultados se presentaron en la Tabla 22, se observó que 6 de los colores detectados contaron con medidas de precisión mayores a 0,86 y solo un color obtuvo una precisión de 0,62. En el análisis cuantitativo de sensibilidad se obtuvo 5 colores detectados con sensibilidad mayor a 0,80 y 2 colores con sensibilidad menor o igual a 0,70. Y por último, se obtuvo el análisis de la especificidad dando como resultado un valor mayor al 0,90 en todos los colores detectados.

Tabla 22. Medidas de clasificación

CLASES	Medidas			Observaciones
	Precisión	Sensibilidad	Especificidad	
Azul	0,91	1,00	0,98	10
Negro	1,00	1,00	1,00	10
Rojo	0,91	1,00	0,98	10
Rosado	1,00	0,80	1,00	10
Verde	0,86	0,60	0,98	10
Blanco	0,62	1,00	0,90	10
Amarillo	1,00	0,70	1,00	10
TOTAL	0,90	0,87	0,90	70

Por último, se realizó en el Anexo 4, el manual de usuario donde consta la descripción, la instalación y el funcionamiento del sistema detector de colores en tejidos con hilo por medio de visión artificial.

CAPÍTULO V

5.1 Conclusiones

- Derivado del análisis de los requerimientos y necesidades presentadas en la distribuidora de hilos, se determinaron los criterios más importantes del sistema de visión artificial, así como la alternativa de solución al sistema que comprendió el uso del lenguaje de programación Python, librerías de visión artificial como OpenCV, análisis de datos Numpy y gráficos Matplotlib.
- Basado en la descripción de los componentes que conforman el diseño del sistema de visión artificial se estableció como punto de mayor investigación la compatibilidad de los elementos conectados al sistema embebido Raspberry, con librerías de visión artificial, lectura de datos, desarrollador de interfaz de usuario y compatibilidad con la cámara.
- El sistema de visión artificial implementado permitió detectar los colores de los hilos mediante el procesamiento de imágenes de diferentes muestras de tejidos de hilos; dando como resultado el nombre y código del color de la empresa según el catálogo de colores. Adicionalmente, se logró el diseño de un sistema ligero, pequeño y de fácil movilidad.
- Finalmente, para validar el sistema se analizó cuantitativamente la precisión, sensibilidad y especificidad del sistema mediante una matriz de confusión. La cual dio como resultado que el sistema de visión artificial es mayormente establecido en el parámetro de especificidad con un valor de 0,97, además tiene una sensibilidad de 0,87 y precisión de 0,90 en los colores propuestos.

5.2 Recomendaciones

- En la implementación de la cámara se recomienda realizar la calibración de los extremos de la cámara ya que sin una calibración se puede producir el efecto angular ojo de pez el cual perjudica a la detección de colores en los extremos de las muestras.
- En la implementación de la iluminación se recomienda realizar un controlador de luminosidad del ambiente ya que mantener una misma medida de lúmenes en las leds puede mejorar la calibración del sistema.
- En la detección de color del píxel se recomienda redimensionar la imagen a una imagen de menor cantidad de píxeles para segmentar de mejor manera los colores que se encuentran en la imagen.

REFERENCIAS

- Albert, N., & Antiga, J. (2012a). *Aplicación práctica de la visión artificial en el control de procesos industriales*. 0(0), 1–28.
- Albert, N., & Antiga, J. (2012b). *Aplicación práctica de la visión artificial en el control de procesos industriales Conocimientos básicos de visión artificial*. 0(0), 1–28. <http://visionartificial.fpcat.cat/wp-content/uploads/Conocimientos.pdf>
- Al-jabery, K. K., Obafemi-Ajayi, T., Olbricht, G. R., & Wunsch II, D. C. (2020). Data analysis and machine learning tools in MATLAB and Python. In *Computational Learning Approaches to Data Analytics in Biomedical Applications* (Issue MI, pp. 231–290). Elsevier. <https://doi.org/10.1016/B978-0-12-814482-4.00009-7>
- Al-Osaimi, F., Bennamoun, M., & Mian, A. (2011). Illumination normalization of facial images by reversing the process of image formation. *Machine Vision and Applications*, 22(6), 899–911. <https://doi.org/10.1007/s00138-010-0309-5>
- Bennabhaktula, G. S., Alegre, E., Karastoyanova, D., & Azzopardi, G. (2022). Camera model identification based on forensic traces extracted from homogeneous patches. *Expert Systems with Applications*, 206(June), 117769. <https://doi.org/10.1016/j.eswa.2022.117769>
- Bradski, G., Mallick, S., Petrovicheva, A., Serebryakov, G., & Khanova, T. (2000). *OpenCV*. OpenCV Resources.
- Bulanon, D., Kataoka, T., Ota, Y., & Hiroma, T. (2002). A segmentation algorithm for the automatic recognition of Fuji apples at harvest. *Biosystems Engineering*, 83(4), 405–412. <https://doi.org/10.1006/bioe.2002.0132>
- Cubero, S., Aleixos, N., Moltó, E., Gómez-Sanchis, J., & Blasco, J. (2011). Advances in Machine Vision Applications for Automatic Inspection and Quality Evaluation of Fruits and Vegetables. *Food and Bioprocess Technology*, 4(4), 487–504. <https://doi.org/10.1007/s11947-010-0411-8>
- Dorj, U., Lee, M., & Yun, S. (2017). An yield estimation in citrus orchards via fruit detection and counting using image processing. *Computers and Electronics in Agriculture*, 140, 103–112. <https://doi.org/10.1016/j.compag.2017.05.019>
- Du, W., Liando, J., & Li, M. (2017). Soft Hint Enabled Adaptive Visible Light Communication over Screen-Camera Links. *IEEE Transactions on Mobile Computing*, 16(2), 527–537. <https://doi.org/10.1109/TMC.2016.2551750>

- Fuentes, J., Ruiz, J., & Rendón, J. M. (2015). Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, 43(1), 55–81. <https://doi.org/10.1007/s10462-012-9365-8>
- Guida, P., Casaburi, A., Busiello, T., Lamberti, D., Sorrentino, A., Iuppariello, L., D'Albore, M., Colella, F., & Clemente, F. (2019). An alternative to plaster cast treatment in a pediatric trauma center using the CAD/CAM technology to manufacture customized three-dimensional-printed orthoses in a totally hospital context: a feasibility study. *Journal of Pediatric Orthopaedics B*, 28(3), 248–255. <https://doi.org/10.1097/BPB.0000000000000589>
- Heimberger, M., Horgan, J., Hughes, C., McDonald, J., & Yogamani, S. (2017). Computer vision in automated parking systems: Design, implementation and challenges. *Image and Vision Computing*, 68, 88–101. <https://doi.org/10.1016/j.imavis.2017.07.002>
- Hunter, J., Dale, D., Firing, E., & Droettboom, M. (2014). *Matplotlib Release 1.4.2*.
- Ji, B., Zhu, W., Liu, B., Ma, C., & Li, X. (2009). Review of recent machine-vision technologies in agriculture. *2009 2nd International Symposium on Knowledge Acquisition and Modeling, KAM 2009*, 3, 330–334. <https://doi.org/10.1109/KAM.2009.231>
- Kakani, V., Nguyen, V. H., Kumar, B., Kim, H., & Pasupuleti, V. (2020). A critical review on computer vision and artificial intelligence in food industry. *Journal of Agriculture and Food Research*, 2(February), 100033. <https://doi.org/10.1016/j.jafr.2020.100033>
- Kulappurath, S., & Shamey, R. (2021). The effect of luminance on the perception of small color differences. *Color Research & Application*, 46(5), 929–942. <https://doi.org/10.1002/col.22637>
- Losson, O., Macaire, L., & Yang, Y. (2010). Comparison of Color Demosaicing Methods. In *Advances in Imaging and Electron Physics* (1st ed., Vol. 162, Issue C, pp. 173–265). Elsevier Inc. [https://doi.org/10.1016/S1076-5670\(10\)62005-8](https://doi.org/10.1016/S1076-5670(10)62005-8)
- McGuinness, B., Duke, M., Au, C. K., & Lim, S. H. (2021). Measuring radiata pine seedling morphological features using a machine vision system. *Computers and Electronics in Agriculture*, 189(August), 106355. <https://doi.org/10.1016/j.compag.2021.106355>
- Mucho Mejor Ecuador. (2021). *El sector textil y de confecciones y su importancia para Ecuador*. Muchomejorecuador.Org.Ec. <https://www.muchomejorecuador.org/ec/el-sector-textil-y-de-confecciones-y-su-importancia-para-ecuador/>
- NumPy. (2008). *NumPy user guide*. NumPy.

- Palacios, O. P. (2017). Descripción de los colores de una imagen mediante técnicas de Deep learning. *TFG EN INGENIERIA INFORMÁTICA, ESCOLA D'ENGINYERIA (EE), UNIVERSITAT AUTÓNOMA DE BARCELONA (UAB)*.
- Pandas. (2015). *Python Data Analysis Library*. Python Data Analysis Library/pandas.
- Pasuy, P. (2019). *Clasificación de aguacates basado en visión por computador* [Universidad Técnica del Norte]. <http://repositorio.utn.edu.ec/handle/123456789/9364>
- Patel, P. S., Marcum, D. L., & Remotigue, M. G. (2006). Automatic CAD model topology generation. *International Journal for Numerical Methods in Fluids*, 52(8), 823–841. <https://doi.org/10.1002/fld.1166>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., & Thirion, B. (2012). Scikit-learn: Machine Learning in Python Fabian. *Journal of Machine Learning Research*, 12(9), 2825–2830.
- Rock, S., & Wozny, M. (1992). Generating Topological Information from a “Bucket of Facets.” *Appears in Solid Freeform Fabrication Symposium Proceedings*, 251–259.
- Ruuska, S., Hämäläinen, W., Kajava, S., Mughal, M., Matilainen, P., & Mononen, J. (2018). Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle. *Behavioural Processes*, 148, 56–62. <https://doi.org/10.1016/j.beproc.2018.01.004>
- Sabzi, S., Abbaspour, Y., García, G., Ruiz, A., & Molina, J. (2018). Segmentation of Apples in Aerial Images under Sixteen Different Lighting Conditions Using Color and Texture for Optimal Irrigation. *Water*, 10(11), 1634. <https://doi.org/10.3390/w10111634>
- Shafer, S. (1985). Using color to separate reflection components. *Color Research & Application*, 10(4), 210–218. <https://doi.org/10.1002/col.5080100409>
- Slaughter, D. C., Giles, D. K., & Downey, D. (2008). Autonomous robotic weed control systems: A review. *Computers and Electronics in Agriculture*, 61(1), 63–78. <https://doi.org/10.1016/j.compag.2007.05.008>
- Søgaard, H., & Olsen, H. (2003). Determination of crop rows by image analysis without segmentation. *Computers and Electronics in Agriculture*, 38(2), 141–158. [https://doi.org/10.1016/S0168-1699\(02\)00140-0](https://doi.org/10.1016/S0168-1699(02)00140-0)
- Sun, K., Artan, N. S., & Dong, Z. (2021). Calibration for Ambient Light Correction in Screen-to-Camera Visible Light Communication. *Results in Optics*, 5, 100122. <https://doi.org/10.1016/j.rio.2021.100122>
- Ultimaker. (2017). *Ficha de datos técnicos PLA*. <https://ultimaker.com/download/67583/TDS-PLA-v3.011-spa-ES.pdf>

- Wu, X., & Zhang, N. (2004). Primary-Consistent Soft-Decision Color Demosaicking for Digital Cameras (Patent Pending). *IEEE Transactions on Image Processing*, *13*(9), 1263–1274. <https://doi.org/10.1109/TIP.2004.832920>
- Zandonadi, R., Pinto, F., Sena, D., Queiroz, D., Viana, P., & Mantovani, E. (2005). Identification of lesser cornstalk borer-attacked maize plants using infrared images. *Biosystems Engineering*, *91*(4), 433–439. <https://doi.org/10.1016/j.biosystemseng.2005.05.002>

ANEXOS

Anexo 1. Entrevistas a los usuarios

	Entrevistado 1 Abel	Entrevistado 2 Fernando	Entrevistado 3 Jordy	Entrevistado 4 Luis	Entrevistado 5 Wilson
¿Cuáles son los problemas que los clientes tienen para identificar los colores de los hilos?	Los clientes no saben los códigos o los nombres de cada color	Existen variedad de colores y tonos de colores, pero las personas no saben cual es el tono del color	Los clientes conocen con otros nombres a los colores que la empresa dispone	Hay muchos colores por lo que es fácil equivocarse con la elección del color por lo general los clientes traen sus muestras	No se comparte los mismos nombres del color el cliente tiene su catálogo y la empresa su propio catalogo
¿Cuáles son los problemas que los trabajadores tienen para identificar los colores de los hilos?	Los clientes llegan con muestras de colores que la empresa no tiene y se dificulta el buscar en bodegas y que el color coincida	La dificultad es la variedad de colores ya que incluso los clientes traen diferentes gamas de colores a los que la empresa tiene en bodega	Los clientes nos dicen otros nombres diferentes a como disponemos en la carta de colores de la empresa	Cuando los trabajadores son nuevos es muy difícil reconocer los colores solicitados por los clientes. Después la identificación del color del hilo se lo hace por experiencia	Cuando hay problemas de los productores de los hilos y no viene el color con los nombres ya etiquetados según el catálogo de colores proporcionado
¿Cómo se realiza la comparación de colores?	Se realiza con la carta de colores por medio visual comparando el hilo y la muestra, afuera en el sol para mayor visibilidad	El cliente trae la muestra física del color, se busca el color en bodega comparando con la carta de colores	La comparación de los colores se lo realiza de manera visual dependiendo de la hora del día afuera en el sol o adentro de la empresa con el catálogo de colores	Por medio del código y nombres del color del hilo disponibles de una carta de colores	El color siempre tiende a cambiar con la diferencia de luz del ambiente dependiendo de la hora
¿Cómo el cliente valida que el color solicitado sea el correcto?	El cliente trae su muestra, o trae su tejido y se lo compara lo más parecido que sea de manera visual.	Se indica el color del hilo y la muestra, se pide la aprobación del cliente por medio visual	El cliente mira su muestra, el hilo proporcionado y la carta de colores y valida si le parece igual el color	Se lo realiza por medio visual ya sea dentro de la empresa o fuera de la empresa con el sol para una mejor visualización	En función de su experiencia, los clientes ya conocen los colores por medio visual
Si se diseñara un sistema para identificar los colores ¿Qué	Que salga el nombre del color y más el código de color que la empresa utiliza	Que se pueda ingresar el código, el nombre del color y que se ejecute el programa mediante esos datos.	Debe mostrar el nombre del color o el código de la empresa que posee	Que nos ayudara con el código y nombre del color	Mientras más información la máquina proporcione del color sería mucho

requerimientos debería tener?	para saber que tonos es	Que el cliente traiga su muestra, yo meto la muestra a la máquina y la máquina me diga el código y el color que necesito			mejor. Como el código, nombre y empresa
	Debe ser portátil por fija traería muchos problemas para identificar muestras que este en otro lugar. Deber ser ligera para transportarla.	Sea portátil o que se pueda movilizar de una manera sencilla por la empresa. La característica debería ser como un portafolio o como una laptop. De unos 10 kg de peso más o menos ligera	Sea portátil para poder verificar los colores en las bodegas. Debe ser de tamaño pequeño para movilizarla por los espacios de las bodegas	Que sea portátil para movilizarlo en todo el espacio de la empresa. La máquina debería ser pequeña y comparada a una computadora	Debería ser portable para llevarlo a los diferentes pisos de la empresa. Interfaz directamente con la computadora para verificar la existencia del producto

Anexo 3. Algoritmo de visión artificial

```
import cv2
import pandas as pd

# declarar variables globales (se usan más adelante)
clicked = False
r = g = b = x_pos = y_pos = 0

##Lectura del video
cap = cv2.VideoCapture(0)

## obtenemos el ancho y alto del video

#print(size)
while(cap.isOpened()):

    ret,frame = cap.read()
    img2=frame
    if ret:

        cv2.imshow('DETECTOR DE COLOR', img2)

        if cv2.waitKey(1) & 0xFF==ord("a"):
            cv2.imwrite("martin2222.jpg",img2)

        if cv2.waitKey(20) & 0xFF == 27:
            break

    else:
        break
```

```

cap.release()
cv2.destroyAllWindows()

#Leer la imagen a procesar
img = cv2.imread ('martin2222.jpg')

# Leyendo un archivo csv con pandas y dando nombres a cada columna (encabezado)
index = ["color", "color_name", "hex", "R", "G", "B"]
csv = pd.read_csv('colors.csv', names=index, header=None)
#print(csv.loc[1,R])

# función para calcular la distancia mínima de todos los colores y obtener el color más
coincidente
def get_color_name(R, G, B):
    minimum = 10000 #Un número cualquiera referencial
    for i in range(len(csv)):
        #Valores absolutos mínimos del archivo csv
        # d = abs(R - int(csv.loc[i, "R" Nombre de la columna]))
        d = abs(R - int(csv.loc[i, "R"])) + abs(G - int(csv.loc[i, "G"])) + abs(B - int(csv.loc[i,
"B"]))

        if d <= minimum:
            minimum = d
            cname = csv.loc[i, "color_name"]
    return cname

# función para obtener las coordenadas x, y del mouse, haga doble clic
def draw_function(event, x, y, flags, param):
    if event == cv2.EVENT_RBUTTONDOWN: #Escoger la acción del mouse
        global b, g, r, x_pos, y_pos, clicked #Se crea variables para que solo funcione en esta
función

```

```

clicked = True
x_pos = x
y_pos = y
#print(x,y)
b, g, r = img2[y, x]
b = int(b)
g = int(g)
r = int(r)
#print(b,g,r)

cv2.namedWindow('DETECTOR DE COLOR')
cv2.setMouseCallback('DETECTOR DE COLOR', draw_function)

while True:

    cv2.imshow("DETECTOR DE COLOR", img2)
    if clicked:

        # cv2.rectangle (imagen, punto inicial, punto final, color, grosor) -1 rellena el rectángulo
        # completo
        cv2.rectangle(img2, (20, 20), (750, 60), (b, g, r), -1)

        # Creación de una cadena de texto para mostrar (nombre del color y valores RGB)
        text = get_color_name(r, g, b) + ' R=' + str(r) + ' G=' + str(g) + ' B=' + str(b)

        # cv2.putText (img, texto, inicio, fuente (0-7), escala de fuente, color, grosor, tipo de
        # línea)
        cv2.putText(img2, text, (50, 50), 2, 0.8, (255, 255, 255), 2, cv2.LINE_AA)

        # Para colores muy claros, mostraremos el texto en color negro.
        if r + g + b >= 600:
            cv2.putText(img2, text, (50, 50), 2, 0.8, (0, 0, 0), 2, cv2.LINE_AA)

    clicked = False

```

```
# Romper el ciclo cuando el usuario presione la tecla 'esc'  
if cv2.waitKey(20) & 0xFF == 27:  
    break  
  
cv2.destroyAllWindows()
```

Anexo 4. Manual de usuario

MANUAL DE USUARIO

SISTEMA PARA DETECTAR COLORES EN TEJIDOS CON HILO



Descripción

Instalación

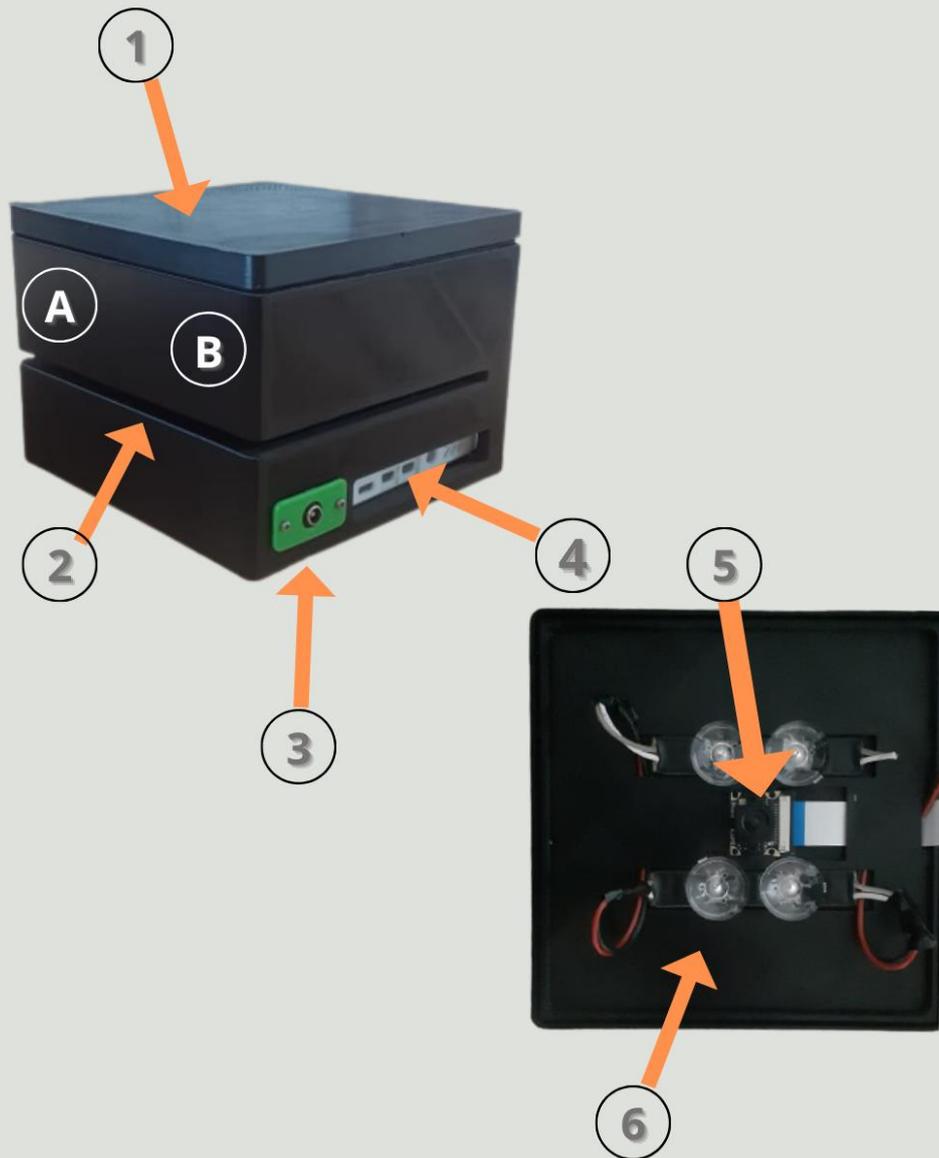
Funcionamiento

DESCRIPCIÓN DEL SISTEMA

La carcasa base tiene dos particiones:

A. Partición izquierda

B. Partición derecha



1.Tapa de la carcasa

2.Carcasa base e ingreso para la muestra

3.Fuente de alimentación de luces

4.Sistema embebido raspberry

6.Cámara raspberry

7.Luces led

INSTALACIÓN DEL SISTEMA

PASO 1

Para ensamblar el sistema primero se debe implementar el sistema embebido. Se identifica la carcasa de la raspberry y se introduce en la partición B de la carcasa. Junto a la entrada tipo jack del sistema de luces



PASO 2

A continuación, se implementa la cámara y las luces led en la tapa superior de la carcasa.



PASO 3

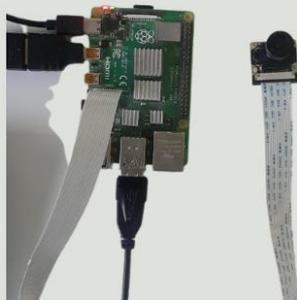
Completados los pasos anteriores, asegurese de que el sistema se vea ensamblado de la siguiente manera. Juntando las particiones A y B con la tapa superior de la carcasa.



INSTALACIÓN DEL SISTEMA

PASO 4

Al sistema embebido se debe conectar la fuente de alimentación, la cámara, hdmi y mouse. La conexión debe ser tal cual se muestra a continuación una vez que ya se haya ensamblado todas las partes de la carcasa.

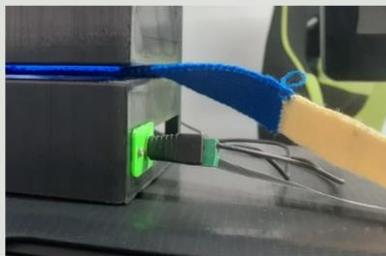


IMPORTANTE

La fuente de alimentación de la raspberry debe ser de 5V y 3A para su correcto funcionamiento.

PASO 6

El sistema de luces tiene una fuente de alimentación propia que debe ser de 6V y 1A para una correcta iluminación.



IMPORTANTE

La fuente de alimentación del sistema de las luces debe ser de 6V y 1A.

PASO 7

Una vez ejecutados todos los pasos mencionados y debidamente energizado el sistema. El hdmi debe ser conectado a cualquier monitor para visualizar la interfaz de usuario. Una vez conectado el monitor se puede hacer uso del sistema.



FUNCIONAMIENTO DEL SISTEMA

Este es el sistema de visión artificial para detectar colores en tejidos con hilo para lo cual se debe introducir la muestra de tejido que se desea identificar el color.



Una vez introducida la muestra se procede a realizar la identificación del color desde la interfaz de usuario. El primer paso es pulsar el botón de **INICIAR VIDEO**



Los siguientes pasos son:

Pulsar el botón **CAPTURAR IMAGEN**

Pulsar el botón **DETECTAR COLOR**



Se muestra una ventana emergente en donde se debe hacer click en la zona de interés. Y se muestra el código de color perteneciente a la muestra de tejido de hilo identificada.

Para realizar otra identificación de color se debe cerrar la ventana emergente y repetir los pasos desde la captura de imagen.

Si desea finalizar la identificación se debe pulsar **FINALIZAR**.