

## RAE

1. TIPO DE DOCUMENTO: Trabajo de grado para optar por el título de INGENIERO MECATRÓNICO.
2. TÍTULO: RECONOCIMIENTO CONTINUO DE OBJETOS POR MEDIO DEL RECONOCIMIENTO ESTADÍSTICO DE PATRONES (REP) CON CLASIFICACIÓN SUPERVISADA.
3. AUTOR: Juan Esteban Valencia Zuleta
4. LUGAR: Bogotá, D.C
5. FECHA: julio de 2017
6. PALABRAS CLAVE: Tiempo Real, Visión Artificial, Reconocimiento Estadístico de Patrones (REP), Clasificación Supervisada, Reconocimiento de Objetos, Extracción de Características.
7. DESCRIPCIÓN DEL TRABAJO: Se presenta la implementación, construcción, algoritmos y desarrollo de un sistema identificador de piezas mediante control visual, en el que se extraen características morfológicas de la imagen y con éstas se determina estadísticamente cuál es el objeto más probable. Como clasificador se entrenó una matriz de resultados por medio de clasificación supervisada. El Software usado es Simulink y el hardware usado es Raspberry PI con sus accesorios.
8. LÍNEAS DE INVESTIGACIÓN: Análisis y procesamiento de señales
9. METODOLOGÍA: Es de carácter empírico analítico, con un enfoque metodológico con base en el estudio.
10. CONCLUSIONES: Se logró obtener un reconocimiento con una precisión de 81.25%. La banda transportadora encargada de ubicar los objetos en la posición correcta se controló mediante un control que regula la velocidad de la banda, tomando como señal de realimentación la posición actual del objeto obtenida con una cámara. La respuesta del sistema es lenta. Sin embargo, así se logró un buen control con margen de error de 30 px.

RECONOCIMIENTO CONTINUO DE OBJETOS POR MEDIO DEL  
RECONOCIMIENTO ESTADÍSTICO DE PATRONES (REP) CON CLASIFICACIÓN  
SUPERVISADA

JUAN ESTEBAN VALENCIA ZULETA  
jevalencia@academia.usbbog.edu.co

UNIVERSIDAD DE SAN BUENAVENTURA SEDE BOGOTÁ  
FACULTAD DE INGENIERÍA  
PROGRAMA DE INGENIERÍA MECATRÓNICA  
BOGOTÁ

21 DE JULIO DE 2017

RECONOCIMIENTO CONTINUO DE OBJETOS POR MEDIO DEL  
RECONOCIMIENTO ESTADÍSTICO DE PATRONES (REP) CON CLASIFICACIÓN  
SUPERVISADA

JUAN ESTEBAN VALENCIA ZULETA

Proyecto de Grado para optar al título de  
Ingeniero Mecatrónico

Tutor  
Ing. Luis Barrero Páez

UNIVERSIDAD DE SAN BUENAVENTURA SEDE BOGOTÁ  
FACULTAD DE INGENIERÍA  
PROGRAMA DE INGENIERÍA MECATRÓNICA  
BOGOTÁ

21 DE JULIO DE 2017

NOTA DE ACEPTACIÓN

---

---

---

---

---

---

---

FIRMA DEL PRESIDENTE DEL JURADO

---

FIRMA DEL JURADO

---

FIRMA DEL JURADO

BOGOTÁ D.C. JULIO DE 2017

## AGRADECIMIENTOS

Primero que todo, agradecimientos a mi familia, mis padres Juan Diego Valencia y María Liliana Zuleta. A mi hermana Claudia Milena. A ellos por guiarme siempre durante toda mi vida.

Agradezco a mis amigos, Kmilox y Juanda, que tuve durante la carrera por tantas victorias y derrotas compartidas, sin importar la situación siempre adelante.

Un abrazo y agradecimiento muy especial a Chevio por ser más que un amigo, un tutor y socio de trabajo. Por enseñarme tanto de programación y siempre motivarme adelante.

El mayor de todos los agradecimientos es para mi tutor Luis Efrén, por su enseñanza, increíble paciencia, literalmente aguantarme, y no dejarme desmotivar con este proyecto. Siempre motivándome y apoyándome.

Por último, también debo agradecer a quienes me acompañaron de principio a fin, desde el colegio hasta el final de mi carrera, a Google y Wikipedia porque sin ellos jamás lo hubiera logrado.

A todos mil gracias

## CONTENIDO

<b>RAE .....</b>	<b>1</b>
<b>CONTENIDO .....</b>	<b>6</b>
<b>LISTA DE TABLAS .....</b>	<b>9</b>
<b>LISTA DE FIGURAS .....</b>	<b>10</b>
<b>LISTA DE ANEXOS .....</b>	<b>13</b>
<b>RESUMEN .....</b>	<b>14</b>
<b>INTRODUCCIÓN .....</b>	<b>15</b>
<b>1. PLANTEAMIENTO DEL PROBLEMA .....</b>	<b>16</b>
1.1 DESCRIPCIÓN DEL PROBLEMA.....	16
1.2 JUSTIFICACIÓN .....	17
1.3 ANTECEDENTES .....	17
1.4 OBJETIVOS .....	19
1.4.1 <i>Objetivo General</i> .....	19
1.4.2 <i>Objetivos Específicos</i> .....	19
1.5 ALCANCES Y LIMITACIONES DEL PROYECTO .....	19
1.5.1 <i>Alcances</i> .....	19
1.5.2 <i>Limitaciones</i> .....	21
<b>2 MARCO TEÓRICO .....</b>	<b>22</b>
2.1 VISIÓN ARTIFICIAL .....	22
2.1.1 <i>Introducción</i> .....	22
2.1.2 <i>Tipos de Luz</i> .....	24
2.1.3 <i>Contribución de la luz ambiental</i> .....	25
2.1.4 <i>Interacción de la luz con objetos</i> .....	27
2.1.5 <i>El Color</i> .....	28
2.1.6 <i>Técnicas de Iluminación</i> .....	29
2.2 PROCESAMIENTO DE LA IMAGEN.....	31
2.2.1 <i>Conceptos de la imagen</i> .....	31
2.2.2 <i>Filtrado</i> .....	33
2.2.3 <i>Reducción de ruido con filtros</i> .....	33
2.2.4 <i>Umbralización</i> .....	35
2.2.5 <i>Erosión</i> .....	36
2.2.6 <i>Dilatación</i> .....	37

2.3	CARACTERIZACIÓN DE OBJETOS .....	38
2.3.1	Regiones .....	38
2.3.2	Caracterización .....	38
2.4	FUNDAMENTOS DE PROBABILIDAD Y ESTADÍSTICA .....	40
2.4.1	Conceptos .....	40
2.4.2	Tipos de distribuciones.....	41
2.4.3	Regla 68 – 95 – 99.7.....	44
2.5	RECONOCIMIENTO ESTADÍSTICO DE PATRONES (REP).....	45
2.5.1	Introducción.....	45
2.5.2	Clasificación supervisada versus no supervisada .....	45
2.5.3	Clasificador bayesiano Ingenuo .....	46
<b>3</b>	<b>DISEÑO INGENIERIL .....</b>	<b>50</b>
3.1	REQUERIMIENTOS TÉCNICOS .....	50
3.1.1	Geometría .....	50
3.1.2	Banda transportadora.....	52
3.1.3	Iluminación .....	55
3.2	HARDWARE UTILIZADO.....	59
3.2.1	Raspberry Pi 3B (RPI).....	59
3.2.2	RaspiCam.....	62
3.2.3	Fuente de energía oficial Raspberry PI.....	62
3.2.4	Controlador de Motores L298N.....	63
3.2.5	Otros.....	64
3.3	SOFTWARE UTILIZADO .....	67
3.3.1	Matlab y Simulink .....	67
3.3.2	Rhinoceros .....	71
3.3.3	PuTTY .....	72
3.3.4	SocketTest v3.0.0.....	72
3.3.5	LiveCode .....	74
3.3.6	Fritzing.....	75
3.3.7	Otros.....	76
3.4	CONSTRUCCIÓN.....	76
3.4.1	Modelado 3D .....	76
3.4.2	Proceso de manufactura .....	77
3.4.3	Conexiones electrónicas .....	80
3.5	PROGRAMACIÓN .....	82
3.5.1	Primeros pasos .....	82
3.5.2	Localización de objetos y selección de características.....	84

3.5.3	<i>Algoritmo de procesamiento de imagen.....</i>	86
3.5.4	<i>Algoritmo de toma de muestras .....</i>	89
3.5.5	<i>Algoritmo constructor de las clases .....</i>	91
3.5.6	<i>Algoritmo modelador estadístico de datos .....</i>	93
3.5.7	<i>Algoritmo clasificador .....</i>	95
3.5.8	<i>Algoritmo estimador de la precisión .....</i>	98
3.5.9	<i>Control de posición de la banda transportadora. ....</i>	100
3.5.10	<i>Adicional – Interfaz Máquina Humano. ....</i>	101
<b>4</b>	<b>RESULTADOS.....</b>	<b>103</b>
4.1	OBSERVACIONES GENERALES .....	103
4.2	OBSERVACIONES SOBRE LA TOMA DE MUESTRAS .....	104
4.3	ANÁLISIS ESTADÍSTICO DE LAS MUESTRAS.....	105
4.4	EFFECTIVIDAD DE LAS CARACTERÍSTICAS .....	113
4.5	PRECISIÓN DE ACIERTOS .....	114
4.6	ERROR DE POSICIÓN .....	116
4.7	COMPARACIÓN DE VARIOS CLASIFICADORES .....	119
	<b>CONCLUSIONES .....</b>	<b>121</b>
	<b>BIBLIOGRAFÍA .....</b>	<b>122</b>
	<b>ANEXOS A – ESPECIFICACIONES TÉCNICAS DE LA RASPICAM .....</b>	<b>126</b>



## LISTA DE TABLAS

Tabla 1 – Características en Matlab .....	39
Tabla 2 – Peso y dimensiones de las herramientas .....	50
Tabla 3 – Valores típicos de luminosidad para habitaciones [35] .....	56
Tabla 4 – Factor de reflexión según el color del material. [36] .....	56
Tabla 5 – Tabla del factor de utilización. [36] .....	57
Tabla 6 – Valores típicos de lúmenes para tipos de iluminación .....	58
Tabla 7 – Esquema y funcionamiento del controlador de motores. ....	64
Tabla 8 – Especificaciones técnicas pantalla LCD .....	66
Tabla 9 – Velocidad de ejecución versus FPS .....	83
Tabla 10 – Matriz de resultados en forma de distribución normal.....	105
Tabla 11 – Matriz de resultados en forma de máximos y mínimos .....	109
Tabla 12 – Precisión del clasificador según características.....	113
Tabla 13 – Pruebas de clasificación de objetos.....	114
Tabla 14 – Resultados error de posición .....	117
Tabla 15 – Comparación de varios tipos de clasificadores.....	119
Tabla 16 – Especificaciones de Hardware Raspicam.....	126
Tabla 17 – Características de Hardware Raspicam.....	127
Tabla 18 – Características de Software Raspicam.....	128

## LISTA DE FIGURAS

Figura 1 – Contraste de buena y mala iluminación [16].....	23
Figura 2 – Luz de día [17].....	25
Figura 3 – Luz blanca Led [17] .....	25
Figura 4 – Lámpara fluorescente [17].....	25
Figura 5 – Lámpara halógena. [17].....	25
Figura 6 – Generación de alto contraste por aplicación de filtros de luz. [18].....	26
Figura 7 – Comparación de tipos de luz. [19] .....	27
Figura 8 – Comparación de contrastes. [20].....	28
Figura 9 – Iluminación directa frontal. [21] .....	29
Figura 10 – Iluminación difusa de campo brillante. [21].....	29
Figura 11 – Iluminación de campo oscuro difuso. [21].....	30
Figura 12 – Iluminación de campo oscuro. [21] .....	30
Figura 13 – Contraluz. [21] .....	30
Figura 14 – Imagen a color, escala de grises y binaria. [22].....	31
Figura 15 – Reducción de ruido. [24].....	34
Figura 16 – Umbralización [25].....	35
Figura 17 – Erosión [26] .....	36
Figura 18 – Dilatación [27].....	37
Figura 19 – Distribución Normal [31] .....	41
Figura 20 – Distribución Binomial [31] .....	42
Figura 21 – Guías para elegir tipo de distribución [31] .....	43
Figura 22 – La regla 68-95-99.7 [32] .....	44
Figura 23 – Esquema general de un clasificador de patrones [33].....	46
Figura 24 – gráfico de dispersión [34].....	47
Figura 25 – Clasificación de un objeto nuevo (Blanco). [34].....	48
Figura 26 – Raspberry Pi 3B. [38].....	60
Figura 27 – Esquemático Raspberry Pi [40] .....	61
Figura 28 – Raspicam [41].....	62
Figura 29 – Cargador oficial Raspberry Pi 5V 2.5A [42] .....	63

Figura 30 – Tarjeta controladora de motores L298N [43] .....	64
Figura 31 – Extensión Forza con las 3 fuentes de energía utilizadas.....	65
Figura 32 – Pantalla LCD en funcionamiento. ....	65
Figura 33 – Led de potencia 3W con su disipador [45].....	66
Figura 34 – MATLAB software. [46].....	68
Figura 35 – Simulink [46] .....	68
Figura 36 – Librería Simulink para el RPI .....	70
Figura 37 – Función de Matlab Interpretada.....	70
Figura 38 – Diseño mecánico en Rhinoceros .....	71
Figura 39 – Pruebas comunicación UDP entre PC y RPI .....	73
Figura 40 – Programa Fritzing .....	75
Figura 41 – Ensamble 3D en Rhinoceros .....	77
Figura 42 – Cortes láser en proceso.....	78
Figura 43 – Ensamblado Final.....	79
Figura 44 – Esquema electrónico .....	81
Figura 45 – Conexiones.....	81
Figura 46 – Características disponibles en el bloque <i>Blob Analysis</i> [47].....	84
Figura 47 – Espacio de trabajo .....	86
Figura 48 – Procesamiento de imagen en Matlab .....	87
Figura 49 – Algoritmo de procesamiento de imagen .....	88
Figura 50 – Jerarquía de clases y fotos de entrenamiento .....	89
Figura 51 – Algoritmo de entrenamiento y construcción de clases.....	90
Figura 52 – Algoritmo constructor de la matriz del clasificador.....	92
Figura 53 – Procesamiento estadístico de datos .....	94
Figura 54 – Parte uno del algoritmo clasificador.....	96
Figura 55 – Parte dos del algoritmo clasificador .....	97
Figura 56 – Algoritmo para calcula la precisión del clasificador .....	99
Figura 57 – Control de lazo cerrado de posición (Simplificado).....	101
Figura 58 – Discontinuidades en la muestra.....	105
Figura 59 – Características modeladas como distribuciones normales .....	107
Figura 60 – Separación de clases 2D– Eje Menor vs Eje Mayor. ....	110

Figura 61 – Separación de clases 2D – Área vs Eje Mayor.....	110
Figura 62 – Separación de clases 2D – Área vs Eje Menor .....	111
Figura 63 – Separación de Clases 3D – Área vs Eje Menor vs Eje Mayor .....	111
Figura 64 – Gráfica de datos paralelos.....	112
Figura 65 – Prueba de clasificación.....	115
Figura 66 – Ejemplo de control de posición y clasificación con banda del 20% .....	118
Figura 67 – Dimensiones Raspicam .....	128

## LISTA DE ANEXOS

Anexos A – Especificaciones técnicas de la RaspiCam .....	126
---	-----

## RESUMEN

En la presente se propone la implementación de un módulo capaz de reconocer objetos de manera continua por medio del *Reconocimiento Estadístico De Patrones (REP)* con clasificación supervisada.

Este proceso de visión artificial se compone de cinco etapas:

- Obtención de la imagen: Una cámara en funcionamiento continuo toma capturas de imágenes cada cierto periodo en un espacio con condiciones controladas como el tipo luz, ubicación de la luz y los objetos. [1]
- Pre-procesamiento: Se le realizan varios procesos a la captura de imagen para reducir el error aleatorio del sistema captador de imágenes. Algunos son: aplicación de filtros, normalización de niveles, realce de características, conversión a diferentes espacios de colores, umbralización. [2]
- Segmentación: Se segmenta la imagen preprocesada para extraer las zonas de interés, es decir, extraer áreas de interés del fondo en base a ciertos criterios como niveles de gris, textura y locación. Con esto se logra la separación del fondo y el primer plano (*Background/Foreground*), y la agrupación (*Clustering*). [3]
- Extracción de características: A la imagen segmentada se le extrae toda la información intrínseca, como las características morfológicas y la descripción del color, para obtener un vector de características. [4]
- Clasificación: Al vector de características se le aplica un análisis de probabilidades en el reconocimiento estadístico de patrones para determinar qué objeto es y clasificarlo en una clase determinada de un conjunto de clases previamente entrenado, es decir, con clasificación supervisada. [5]

Es un innovador proyecto de Ingeniería Mecatrónica como una aplicación sencilla del análisis de imagen y algoritmos de visión artificial que propone la clasificación por visión artificial como una opción altamente reprogramable, versátil y flexible.

## INTRODUCCIÓN

La visión computacional es una técnica de inspección rápida, consistente y objetiva, que se ha expandido en variadas industrias. Su velocidad y precisión satisfacen el crecimiento de las necesidades de cantidad y calidad, por tanto, ayudando en el desarrollo de procesos totalmente automatizados. [6]

El avance tecnológico está gradualmente encontrando aplicaciones en distintas industrias, en respuesta a uno de los más grandes retos: dar respuesta a la necesidad de una población creciente. Se están llevando a cabo esfuerzos para reemplazar la operación humana por los sistemas automatizados, ya que las operaciones humanas son inconsistentes y menos efectivas. La automatización significa que se llevan a cabo calculadas acciones para controlar un proceso con máxima eficiencia. Es aquí donde el uso de sistemas de visión artificial cobra vital importancia para maximizar la eficiencia de un proceso industrial. [7]

La tecnología del análisis de imagen es relativamente joven y sus orígenes se remontan a los 60s. Ésta ha experimentado un tremendo crecimiento en teoría y aplicación. Ha encontrado aplicación en áreas como el diagnóstico médico, la manufacturación automatizada, la supervisión aérea, el sensado remoto y recientemente en la clasificación de productos y materiales en el reciclamiento industrial. “La visión computacional es una novedosa tecnología para la adquisición y el análisis de imágenes de una escena real por medio de computadores y otros dispositivos para obtener información, o controlar máquinas o procesos”. [8]

La manipulación de objetos, por ejemplo, en una línea de ensamble, es una tarea donde una gran cantidad de diferentes tipos de objetos aparecen en desorden y desorientados. En este caso es necesario un **reconocimiento continuo de objetos** invariante a las condiciones de luminosidad del ambiente, la posición y orientación de los objetos en la línea. Por tanto, se requieren técnicas de reconocimiento de objetos confiables que puedan cumplir totalmente los requerimientos de la industria.

## 1. PLANTEAMIENTO DEL PROBLEMA

En la industria es evidente la necesidad de clasificar ciertos objetos de acuerdo a algún proceso específico, los diferentes métodos existentes de clasificación se basan generalmente en la lectura de sensores, los cuales determinan las características del objeto a clasificar.

Recientemente se han desarrollado nuevos algoritmos que aplican visión artificial para la detección de características, los cuáles se están usando en diferentes aplicaciones mediante los entornos de programación más comunes, permitiendo la clasificación de objetos en más de dos clases. [9]

Este trabajo tiene entonces como fin desarrollar una aplicación de visión artificial por medio del reconocimiento estadístico de patrones. El problema es cómo reconocer objetos (herramientas) por medio de la estadística.

### 1.1 DESCRIPCIÓN DEL PROBLEMA

Dentro de los procesos industriales existen diferentes métodos de clasificación, uno de ellos es por medio de la visión artificial. Sin embargo, entra en cuestión la fiabilidad de la visión artificial comparada con la visión humana. Para demostrar que un sistema de visión artificial útil y confiable hay que garantizar un alto porcentaje de aciertos.

A partir de esta problemática surge la siguiente pregunta: ¿Se puede lograr un reconocimiento de objetos por medio de la visión artificial con un 80% de éxitos?



## 1.2 JUSTIFICACIÓN

Con este proyecto se pretende dar inicio a una aplicación de visión artificial, en donde se reconocen objetos desorientados y desordenados, para que posteriormente, esta información obtenida sea utilizada por otro sistema de manipulación de piezas como por ejemplo un brazo robótico para que pueda manipular los objetos detectados. Además, dicho estudio puede dar inicio a la aplicación de sistemas de visión a otras modalidades como por ejemplo la robótica móvil. Si este proyecto no se realiza, no se contribuirá a la investigación y promoción de la Visión Artificial en la Universidad San Buenaventura puesto que hay muy pocos proyectos e investigaciones en este tema.

## 1.3 ANTECEDENTES

La visión artificial comenzó en las universidades que fueron pioneras en la inteligencia artificial a finales de los sesentas. Comenzó con el fin de imitar el sistema de visión humano, como paso previo para dotar a los robots con comportamiento inteligente. En 1966, se creía que esto podría lograrse a través de un proyecto universitario, adjuntando una cámara a un PC y hacer que "describa lo que vea". [10]

Lo que distinguía a la visión artificial del campo predominante de procesamiento de imágenes digitales en ese momento era un deseo de extraer la estructura tridimensional a partir de imágenes con el objetivo de lograr la plena comprensión del escenario. Los estudios realizados en la década de 1970 formaron las primeras bases de muchos de los algoritmos de visión artificial que existen hoy en día, incluyendo la extracción de bordes de las imágenes, el etiquetado de las líneas, el modelado geométrico y no geométrico, la representación de los objetos como las interconexiones de estructuras más pequeñas, flujo óptico, y la estimación de movimiento. [11]

La década siguiente vio estudios basados en análisis matemático más riguroso y aspectos cuantitativos de la visión artificial. Estos incluyen el concepto de escala-espacio, la inferencia de la forma de diversas señales como sombreado, textura y foco, y modelos de contorno conocidos como serpientes. Los investigadores también se dieron cuenta de que muchos de estos conceptos matemáticos podrían ser tratados dentro del mismo marco de optimización como la regularización y campos aleatorios de *Markov*. Por la década de 1990, algunos de los temas de investigación anteriores se volvieron más activos que otros. La investigación en reconstrucciones 3-D proyectadas dio lugar a una mejor comprensión de la calibración de la cámara. Con el advenimiento de los métodos de optimización para la calibración de la cámara, se entendió que muchas de las ideas ya habían explorado en la teoría de ajuste de paquete desde el campo de la fotogrametría. Esto condujo a métodos para reconstrucciones 3-D de escenas de múltiples imágenes. Se avanzó en el problema de correspondencia estéreo densa y otras técnicas estéreo de vista múltiple. Al mismo tiempo, se utilizaron variaciones de corte gráfico para resolver segmentación de imágenes. Esta década también marcó la primera vez que las técnicas de aprendizaje estadístico se utilizaron en la práctica para reconocer caras en las imágenes (*Eigenface*). Hacia el final de la década de 1990, un cambio significativo se produjo con el aumento de la interacción entre los campos de gráficos por computador y visión artificial. Esto incluyó la representación de imágenes, *morphing* de imagen, interpolación de vista, la construcción de imagen panorámica y principios de representación del campo de luz. [12] [13]

Trabajos recientes han visto el resurgimiento de los métodos basados en características, que se utiliza en conjunción con técnicas de aprendizaje automático y marcos complejos de optimización. [14] [15]

## 1.4 OBJETIVOS

### 1.4.1 Objetivo General

Implementar un algoritmo de aplicación utilizando software de análisis de imagen especializado que sea capaz de reconocer 5 objetos: Destornillador, Alicata, Regla, Cinta y Llave Inglesa, con al menos un 80% de aciertos mediante un algoritmo de reconocimiento estadístico de patrones.

### 1.4.2 Objetivos Específicos

1. Diseñar el algoritmo del entrenador de clases.
2. Diseñar el algoritmo del reconocimiento de objetos.
3. Implementar el módulo para el reconocimiento de objetos.
4. Implementar una banda transportadora al módulo con un control de posición que cambié la velocidad.
5. Probar que el reconocimiento funcione con el objeto ubicado en cualquier zona del espacio de trabajo definido.
6. Migrar todo el código desarrollado a un sistema embebido.
7. Comprobar que el reconocimiento de objetos sea superior al 80% de aciertos.

## 1.5 ALCANCES Y LIMITACIONES DEL PROYECTO

### 1.5.1 Alcances

- Los objetos propuestos a clasificar son aquellos mencionados en el objetivo general.

- Este proyecto tiene como alcance desarrollar un módulo encargado de reconocer objetos.
- Los objetos a clasificar deberán ser alimentados al módulo mediante una banda transportadora. El operario deberá poner los objetos en la banda transportadora, separados unos de los otros.
- La cámara puede ser convencional, IP como una Webcam, o integrada de un sistema embebido si es el caso.
- Se construirá un cuarto con iluminación LED para mejorar las condiciones de luz en el sistema.
- La identificación permitirá reconocer varios objetos simultáneamente siempre y cuando estén separados entre sí.
- Se implementará el control de la banda transportadora utilizando la cámara como el sensor de retroalimentación. El control ubicará un objeto en el centro del espacio de trabajo para que la cámara enfoque mejor. El control sólo moverá la banda transportadora en su único eje de desplazamiento. El espacio de trabajo se define en el ítem 3.5.3
- El control de la banda regulará la velocidad de ésta y la detendrá cuando el objeto haya alcanzado la posición ideal.
- Toda la fase de prototipado del código se desarrollará en software especializado en tratamiento de imágenes. Una vez obtenido un prototipo funcional se procederá a migrar todo el código realizado a un sistema embebido.
- Las entregas parciales del proyecto serán en el software especializado en visión artificial. El entregable final del proyecto será en sistema embebido.
- El único sistema embebido prohibido por la Universidad es Arduino. De resto es libre elección.

### 1.5.2 Limitaciones

- No se podrán reconocer objetos transparentes o que reflejen demasiado la luz.
- La velocidad del reconocimiento depende de la combinación hardware/software utilizado. Se hará un reconocimiento cada 1000 ms o menos.
- No se podrán reconocer objetos que no tengan un alto contraste con el fondo. Por ejemplo, si es de color negro no se podrán clasificar elementos que sean en su totalidad negros o tengan un contorno negro que no genere contraste con el fondo.
- No se podrán reconocer objetos que no sean ubicados con su cara frontal perpendicular al eje central de la cámara, es decir, todos los objetos deben estar acostados en el espacio de trabajo.
- Todos los objetos deben estar en su totalidad dentro del espacio de trabajo definido y además haya una separación de 20mm entre ellos. No deben cruzarse, anteponerse ni solaparse. Así mismo deben estar totalmente dentro de la banda transportadora y que no haya salientes.

## 2 MARCO TEÓRICO

### 2.1 VISIÓN ARTIFICIAL

#### 2.1.1 Introducción

Quizás ningún otro aspecto del diseño e implementación del sistema de visión artificial ha causado consistentemente más retrasos, excesos de costos y preocupación en general que la iluminación. Históricamente, la iluminación ha sido el último aspecto especificado, desarrollado, o financiado. Esta manera de abordar el tema no era completamente injustificada, ya que hasta hace poco no había una verdadera iluminación específica para los sistemas de visión artificial en el mercado, esto causó que las soluciones para la iluminación consistieran típicamente de productos incandescentes o fluorescentes estándares disponibles al consumidor, con varias cantidades de contribución ambiental.

El conocimiento de los tipos de iluminación, ventajas y desventajas de su uso, eficiencia cuántica del sensor, cámara de visión artificial y rango espectral, técnicas de iluminación y sus campos de aplicación se relacionan con la rugosidad y la reflectividad de una superficie.

Los 4 pilares de la iluminación de la visión artificial son:

- Geometría—La relación tridimensional entre iluminación, luz y cámara.
- Estructura o Patrón—La forma de la luz proyectada sobre la muestra.
- Longitud de onda o Color—Como la luz es reflejada o absorbida distintamente por la muestra y el fondo inmediato.
- Filtros— Bloqueo diferencial y paso de longitudes de onda y/o direcciones de luz.

Entender cómo modificar y amplificar el contraste de la muestra usando los cuatro pilares es crucial en el cumplimiento de los tres criterios de aceptación para evaluar la calidad y robustez de la iluminación. Efectuar cambios al contraste a través de la

geometría involucra cambiar la posición de la muestra, luz y/o cámara hasta encontrar una configuración apta. Por ejemplo, un anillo coaxial de luz (una montada alrededor de la cámara) podría generar una zona de concentración de luz en una superficie de código de barra semi-reflectiva (ver Figura 1), pero con simplemente mover la luz fuera de axis, el reflejo de luz también se mueve fuera de la visión de la cámara. El contraste cambia a través de la estructura o forma de la luz proyectada sobre la muestra, que generalmente es iluminada con focos—o técnicas de iluminación—específicas. Los cambios de contraste debido al color de la iluminación están relacionados al diferencial de absorción de color, no a la reflectividad. [16]

Cuando se ha acumulado, y analizado la información de estas áreas, con respecto a la muestra en específico y de acuerdo con los requerimientos de la inspección, se logra el objetivo principal del análisis de la iluminación de visión artificial—proveer iluminación apta que cumpla los 3 criterios de aceptación consistentemente:

- Maximizar el contraste en los puntos de interés.
- Minimizar el contraste en el resto de la muestra.
- Proporcionar una medida de robustez.

Cada inspección es diferente así que es posible que, por ejemplo, para soluciones de iluminación que cumplan con los criterios de aceptación uno y dos sean efectivas solo si no hay inconsistencias en tamaño de parte, tamaño, forma, orientación, colocación o variables ambientales como la contribución de la luz ambiental.



Figura 1 – Contraste de buena y mala iluminación [16]

### 2.1.2 Tipos de Luz

Las fuentes de luz más usadas en este momento en la visión artificial son fluorescentes, halógenos de cuarzo, LED, Haluro de metal (mercurio) y Xenón.

Las fuentes fluorescentes, halógenas de cuarzo y LED son los más usados para estaciones de inspección de tamaño pequeño o mediano. Típicamente, el Haluro de metal, Xenón, y Sodio de alta presión son más usados en usos a gran escala o en áreas que requieren una fuente de luz muy brillante. El haluro de metal, también conocido como mercurio, es frecuentemente usado en microscopía porque tiene muchos picos discretos en la longitud de onda, que complementan el uso de filtros para estudios de fluorescencia. Una fuente de Xenón es útil para aplicaciones que requieren una luz estroboscópica muy brillante. La Figura 2, Figura 3, Figura 4 y Figura 5 muestran una comparación de los tipos de iluminación: luz del día, fluorescente, Halógeno de cuarzo y LED. Por ejemplo, mientras que la iluminación LED tiene una mayor longevidad, la iluminación halógena de cuarzo puede ser la elección para una inspección en particular, ya que ofrece una mayor intensidad.

No todas las fuentes de luz blanca tienen la misma sombra de blanco para el espectador. En caso de que el sistema de cámaras sea de color (al igual que si es monocromo), También puede ser observado que las imágenes tienen efectos de color, o que los valores grises cambian considerablemente. [17]

En la Figura 2 la imagen muestra colores normales, parecidos a los que el ojo humano vería. El área blanca de referencia a la izquierda es gris claro. En la Figura 3 el logo azul (esquina izquierda superior) más brillante, Imagen de color más frío efecto del tinte azul. En la Figura 4 el área de referencia blanca se ve amarillo-verdoso, los tonos verdes son un poco más brillantes. En la Figura 5 el área de referencia blanca parece naranja, debido a la gran proporción de longitudes de onda amarillas y rojas. Parte completamente descolorada y solo con bajo contraste de la imagen.





Figura 2 – Luz de día [17]



Figura 4 – Lámpara fluorescente [17]



Figura 3 – Luz blanca Led [17]



Figura 5 – Lámpara halógena. [17]

### 2.1.3 Contribución de la luz ambiental

La presencia de luz ambiental puede tener grandes impactos en la calidad y consistencia de las inspecciones, particularmente cuando se usan fuentes multiespectrales como la luz blanca. Los contribuidores ambientales más comunes son la luz solar y luces de fábricas superiores, pero ocasionalmente iluminación errónea de tareas específicas de la visión artificial de otras estaciones de inspección o hasta otras estaciones en el mismo espacio de trabajo pueden tener un impacto.

Hay tres métodos activos para lidiar con la luz ambiental: (1) producir un efecto estroboscópico de alta potencia con pulsos de corta duración, (2) confinamientos físicos, y (3) filtros de paso. El método que es aplicado se elige en función de varios factores. El efecto estroboscópico de alta potencia simplemente abruma y elimina la contribución ambiental, pero tiene desventajas ergonómicas, costo, y esfuerzo de implementación, además no todas las fuentes, tales como la fluorescente, pueden ser abrumadas por un efecto estroboscópico. Si no se puede usar un efecto estroboscópico, y la aplicación requiere el uso de una cámara de color, la luz blanca multiespectral es necesaria para reproducciones precisas y balanceadas del color. En esta circunstancia, un filtro de paso de longitud de ondas estrecho es ineficaz, ya que bloquearía una gran porción de contribución de luz blanca, así dejando a los confinamientos físicos como la mejor opción. [18]

Sin embargo, hay excepciones a esta regla general. Por ejemplo, un filtro de paso corto de 700nm, también conocido como un bloqueador IR, es estándar en cámaras de color ya que el contenido IR puede alterar la precisión y balance del color, del canal verde particularmente. La Figura 6 ilustra cómo el uso de un filtro de paso puede bloquear la luz ambiental eficazmente, particularmente cuando la luz de interés es de fluorescencia de bajo rendimiento.

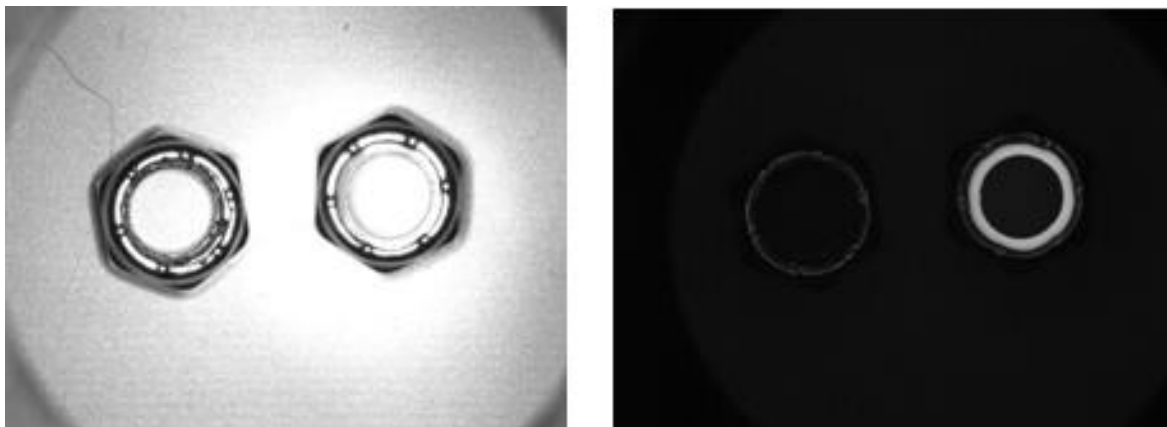


Figura 6 – Generación de alto contraste por aplicación de filtros de luz. [18]

En la Figura 6 la imagen de la izquierda muestra tuercas nyloc con un anillo de luz UV, pero inundado con luz roja “ambiental” de 660 nm. La meta es determinar la

presencia/ausencia de nylon. Dada la gran contribución ambiental, es difícil obtener suficiente contraste de la luz azul fluorescente de rendimiento relativamente bajo de la muestra. La imagen de la derecha tiene la misma iluminación, excepto por un filtro de paso pequeño de 510 nm que fue instalado en el lente de la cámara, efectivamente bloqueando la luz roja “ambiental” y permitiendo que la luz azul de 450 nm pase.

#### 2.1.4 Interacción de la luz con objetos

La manera en que la superficie de una muestra reacciona a la luz ambiental y la creada específicamente para su análisis, está relacionada a muchos factores, incluyendo la forma superficial bruta, la geometría y la reflectividad, así como su composición, topografía y color. Una combinación de todos estos factores determina cuanta luz, y en qué manera, es reflejada a la cámara, y subsecuentemente disponible para adquirir, procesar, y medir. Por ejemplo, una superficie curva y cóncava, así como el fondo de una lata de refresco, refleja una fuente de luz direccional diferentemente de una superficie plana y difusa, así como la del papel. Similarmente, una superficie topográfica, como una tarjeta de circuito impreso poblada, refleja diferentemente de una superficie plana pero finamente texturizada o con hoyuelos, dependiendo del tipo de luz y la geometría. [19]

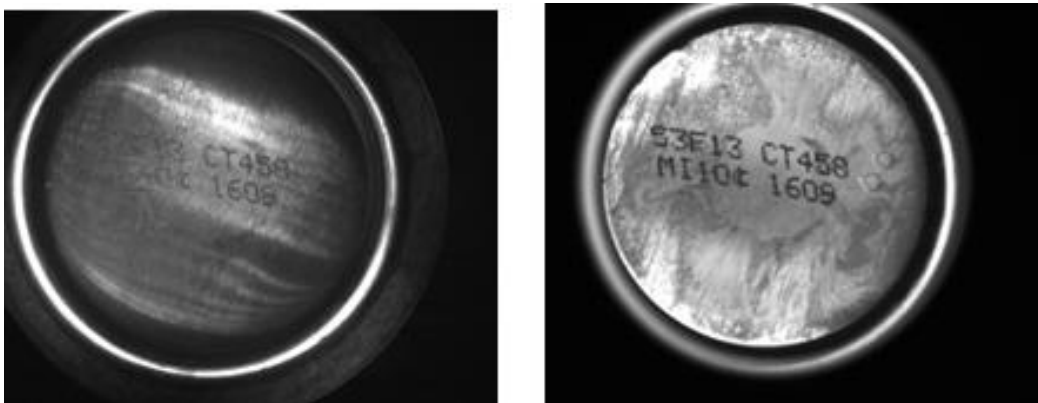


Figura 7 – Comparación de tipos de luz. [19]

En la Figura 7 en la izquierda, la parte de abajo de una lata de refresco es iluminada con un anillo de campo de luz brillante, pero muestra un contraste pobre, iluminación no uniforme y reflexiones cóncavas. En la derecha, la lata de refresco es iluminada con luz difusa, creando un fondo uniforme para que el código pueda ser leído

### 2.1.5 El Color

Los materiales reflejan y/o absorben las distintas longitudes de onda de luz de diferente manera, un efecto que es válido para los espacios de imágenes blanco y negro o de color. Cómo cuando los colores se reflejan las superficies se iluminan; inversamente, Los colores opuestos absorben y las superficies se oscurecen. Usando una simple rueda de colores cálidos contra una rueda de colores, se puede generar un contraste diferencial entre una parte y el fondo, e incluso diferenciar partes de color, dada una paleta limitada y conocida de colores, con una cámara en blanco y negro. [20]

En la Figura 8, se muestran caramelos ilustrando: (a) luz blanca y cámara de color CCD, (b) luz blanca y cámara en blanco y negro, (c) luz roja, iluminando ambos el rojo y el amarillo y oscureciendo el azul, (d) luz verde y roja, produciendo amarillo, iluminando el amarillo más que al rojo, (e) luz verde, iluminando el verde y el azul y oscureciendo el rojo, (f) luz azul, iluminando el azul y oscureciendo los otros.

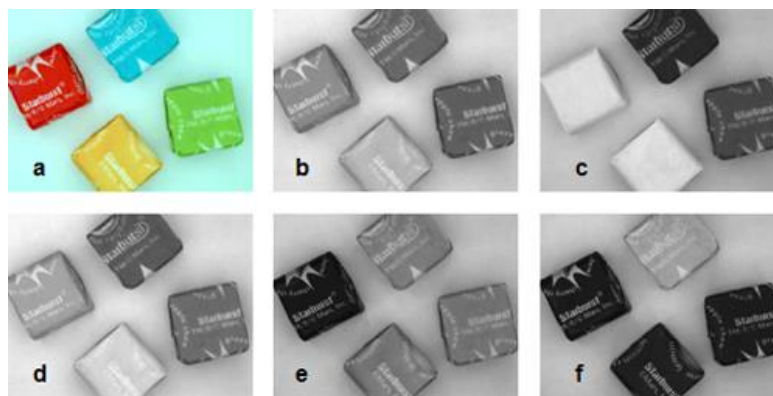


Figura 8 – Comparación de contrastes. [20]

### 2.1.6 Técnicas de Iluminación

El ángulo de incidencia de la luz sobre el objeto también influye en el resultado. Hay muchas técnicas distintas, tales como iluminación frontal o contraluz, iluminación directa o difusa, Iluminación de campo claro o campo oscuro.

Directa Frontal: (un anillo de luz ilumina los objetos directamente, más o menos paralela al eje de la cámara). En la Figura 9 se muestra una imagen no uniforme y moteada.



Figura 9 – Iluminación directa frontal. [21]

Iluminación difusa de campo brillante: En la Figura 10 la imagen se ve más uniforme. Hay un fuerte contraste entre el objeto y el fondo, pero la superficie reflectora del conector “inunda” la cámara, es decir la cámara es “deslumbrada” y no puede detectar algunos detalles. Además, se forman sombras sobre la parte superior del conector.



Figura 10 – Iluminación difusa de campo brillante. [21]

Iluminación de campo oscuro difuso: Luz con un ángulo oblicuo de incidencia de un anillo de luz con un ángulo entre la unidad de iluminación frontal y el objeto. En la Figura 11 se pueden ver más detalles en el conector y no se forman sombras.

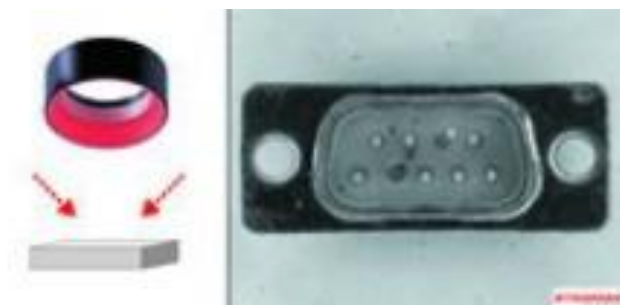


Figura 11 – Iluminación de campo oscuro difuso. [21]

Iluminación de campo oscuro: Ángulo de incidencia poco profundo de la luz en el plano del objeto. En la Figura 12, los bordes de arriba de los pines, el conector y los huecos parecen círculos brillantes y, gracias a esto, pueden ser fácilmente identificados a través del uso de software que analiza imágenes. El pin faltante (no hay círculo brillante) y el pin doblado (posición incorrecta) son visibles con más facilidad que en comparación con la iluminación frontal.

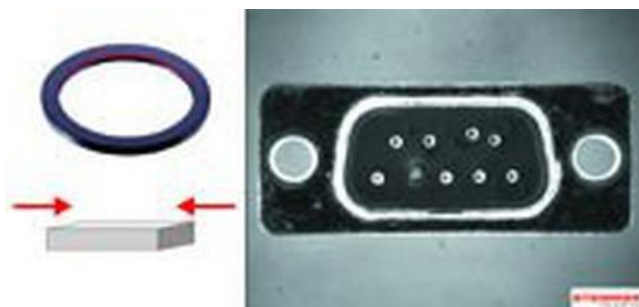


Figura 12 – Iluminación de campo oscuro. [21]

Contraluz: Luz apuntada a la parte trasera del objeto. La luz solamente penetra donde no hay nada para obstruirla. Esto permite que los huecos taladrados en cada lado del conector puedan ser medidos con precisión. Un punto brillante es fácilmente detectable en el espacio de pin faltante. Ver Figura 13.

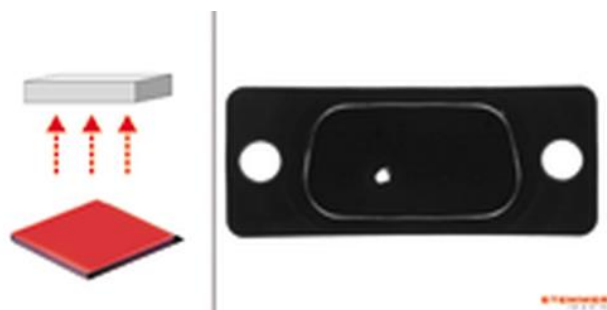


Figura 13 – Contraluz. [21]

## 2.2 PROCESAMIENTO DE LA IMAGEN

### 2.2.1 Conceptos de la imagen

Imagen a color: también se conoce como una imagen RGB. Una verdadera imagen a color es una imagen en la que cada pixel es especificado por tres valores cada uno para los componentes rojo, azul y verde del escalar de píxeles. Arreglo M de-n-por-3 de clase uint8, uint16, single o double cuyos valores de píxeles especifican valores de intensidad. Para matrices simples o dobles, los valores van desde [0, 1]. Para uint8, los valores van desde [0, 255]. Para uint16, los valores van desde [0, 65535].

Escala de Grises: también se conoce como una imagen de intensidad, escala de grises, o nivel de gris. Conjunto de clase uint8, uint16, int16, single o double cuyos valores de píxeles especifican valores de intensidad. Para matrices simples o dobles, los valores van desde [0, 1]. Para uint8, los valores van desde [0,255]. Para uint16, los valores van desde [0, 65535]. Para int16, los valores van desde [-32768, 32767].

Imagen Binaria: Conjunto lógico que contiene sólo 0s y 1s, interpretados como blanco y negro, respectivamente. Ver Figura 14.

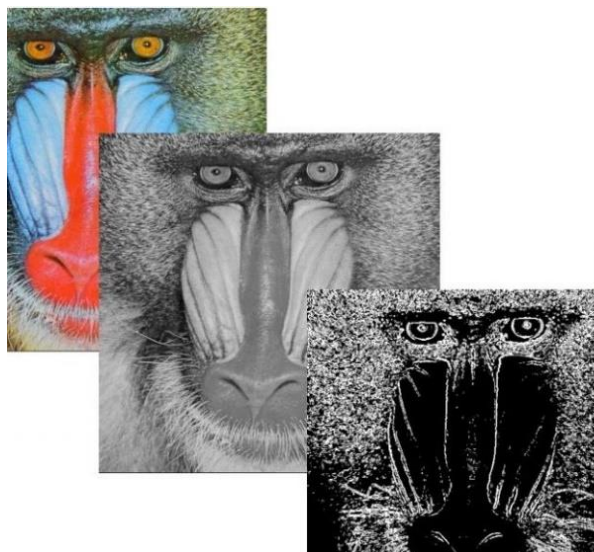


Figura 14 – Imagen a color, escala de grises y binaria. [22]



Resolución: Similarmente a una señal de tiempo unidimensional, el muestreo para imágenes es hecho en el dominio espacial, y la cuantificación es hecha para los valores de brillo. En el proceso de muestreo, el dominio de las imágenes es dividido en filas  $N$  y columnas  $M$ . La región de interacción de una fila y una columna se conoce como píxel. El valor asignado a cada píxel es el brillo promedio de las regiones. La posición de cada píxel es descrita por un par de coordenadas  $(x_i, x_j)$ . La resolución de una señal digital es el número de píxeles presentado en el número de columnas  $\times$  número de filas. Por ejemplo, una imagen con una resolución de  $640 \times 480$  significa que muestra 640 píxeles en cada una de las 480 filas. Otra resolución comúnmente usada es  $800 \times 600$  y  $1024 \times 728$ , entre otras. La resolución es una de las maneras más utilizadas para describir la calidad de imagen de la cámara digital u otro equipo óptico. La resolución de un sistema de visualización o equipo de impresión se expresa a menudo en el número de puntos por pulgada. Por ejemplo, la resolución de un sistema de visualización es de 72 puntos por pulgada (ppp) o puntos por cm.

Niveles de Gris: Representan el número de intervalo de cuantificación en el procesamiento de imágenes en escala de grises. Actualmente, el método más comúnmente usado de almacenamiento es el almacenamiento de 8 bits. Hay 256 niveles de gris en una imagen de escala de grises de 8 bits, y la intensidad de cada píxel puede tener de 0 a 255, con el 0 siendo negro y el 255 siendo blanco. Otro método de almacenamiento comúnmente usado es el almacenamiento de 1 bit. Hay dos niveles de gris, con el 0 siendo negro y el 1 siendo blanco, está siendo referida como una imagen binaria. Como las imágenes binarias son fáciles de operar, imágenes de otro formato de almacenamiento a menudo son convertidas a binarias cuando son usadas para mejorar o detectar bordes. [22]



### 2.2.2 Filtrado

El filtrado es una técnica para modificar o mejorar una imagen. Por ejemplo, Se puede filtrar una imagen para enfatizar ciertas características o remover otras. Las operaciones de procesamiento de imágenes implementadas con filtrado incluyen suavizado, nitidez y mejora de bordes.

El filtrado es una operación de sectores, en la cual el valor de cualquier píxel dado en la imagen de salida es determinado aplicando un algoritmo a los valores de los píxeles en el sector del píxel de entrada correspondiente. El sector de un píxel es un conjunto de píxeles, definidos por sus ubicaciones en relación con ese píxel. El filtrado lineal es un filtrado en el cual el valor del píxel de salida es una combinación lineal de los valores de los píxeles en el sector del píxel de entrada.

Convolución: el filtrado lineal de una imagen es logrado a través de una operación llamada convolución. La convolución es una operación por sectores en la cual cada píxel de salida es la suma ponderada de los píxeles de entrada del sector. La matriz de pesos es llamada núcleo de convolución, también conocido como el filtro. Un núcleo de convolución es un núcleo de correlación que ha sido rotado 180 grados.

Correlación: La operación llamada correlación está altamente relacionada con la convolución. En la correlación, el valor de un píxel de salida es también calculado como una suma ponderada de los píxeles del mismo sector. La diferencia es que la matriz de pesos, en este caso llamada núcleo de correlación, no es rotada durante los cálculos. [23]

### 2.2.3 Reducción de ruido con filtros

Las imágenes digitales son propensas a una variedad de tipos de ruido. El ruido es el resultado de errores en el proceso de adquisición de la imagen que causa que los valores de los píxeles no reflejen la verdadera intensidad de la escena. Hay varias

maneras en las que el ruido puede ser introducido en una imagen, dependiendo de cómo la imagen es creada. Por ejemplo:

Si la imagen es escaneada de una fotografía hecha en cinta, la granularidad de la cinta es una fuente de ruido. El ruido también puede ser el resultado de daños a la cinta, o puede ser introducido por el propio escáner.

Si la imagen es obtenida directamente en un formato digital, El mecanismo para recopilar los datos (tales como un detector CCD) pueden introducir más ruido. La transmisión electrónica de datos de imagen puede introducir ruido.

Se puede usar el filtrado lineal para remover ciertos tipos de ruido. Ciertos filtros, tales como los filtros gaussianos o de promedio, son apropiados para esto. Por ejemplo, un filtro de promedio es útil para eliminar ruido por granularidad de una fotografía. Debido a que cada píxel se establece en el promedio de los píxeles en su sector, las variaciones locales causadas por la granularidad se reducen. Se muestra un ejemplo de ruido en la Figura 15. [24]

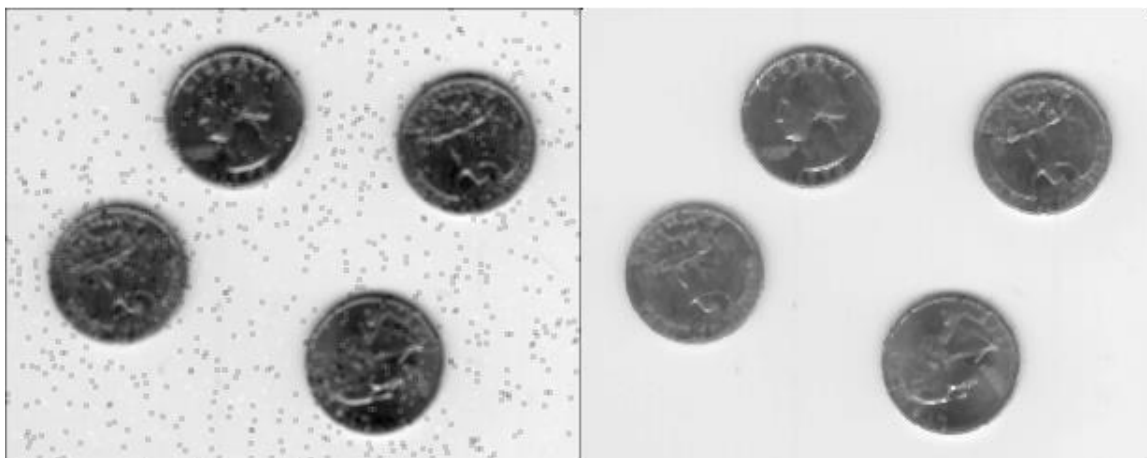


Figura 15 – Reducción de ruido. [24]

### 2.2.4 Umbralización

Es el método de segmentación más simple útil para diferenciar los píxeles que nos interesan del resto (que, eventualmente serán rechazados), Se lleva a cabo una comparación de los valores de intensidad de cada píxel con respecto a un umbral (que se determina de acuerdo al problema que se quiera resolver). [25]

Ejemplo de aplicación: Separar las regiones de una imagen correspondiente a los objetos que queremos analizar. Esta separación se basa en la variación de intensidad entre los píxeles del objeto y los píxeles de fondo.

Una vez se han separado correctamente los píxeles importantes, se pueden marcar con un valor determinado para identificarlos, es decir, se puede asignarles un valor de 0 (negro), 255 (blanco) u otro valor. Se muestra un ejemplo de umbralización en la Figura 16.



Figura 16 – Umbralización [25]

### 2.2.5 Erosión

La erosión es uno de los dos operadores básicos en el área de la morfología matemática, el otro es la dilatación. Típicamente aplicado a imágenes binarias, pero hay versiones que funcionan en imágenes en escala de grises. El efecto básico del operador en una imagen binaria es erosionar los límites de las regiones de píxeles de primer plano (es decir, píxeles blancos típicamente). Por lo tanto, las áreas de los píxeles de primer plano se encojen y los agujeros dentro de esas áreas se hacen más grandes. [26]

El operador de la erosión toma dos piezas de datos como entradas. La primera es la imagen que va a ser erosionada. La segunda es un conjunto de coordenadas (usualmente pequeñas) conocidas como un elemento estructurador (también conocido como núcleo). Es este elemento estructurador es el que determina el efecto preciso de la erosión en la imagen de entrada.

La definición matemática para la erosión de escala de grises es idéntica excepto en la manera en que se deriva el conjunto de coordenadas asociadas con la imagen de entrada. Además, estas coordenadas son 3-D en vez de 2-D.

Como un ejemplo de la erosión binaria, supóngase que el elemento estructurador es un cuadro de 3×3. Nótese que en este y subsecuentes diagramas, los píxeles de primer plano son representados por 1's y el fondo por 0's. Ver Figura 17.

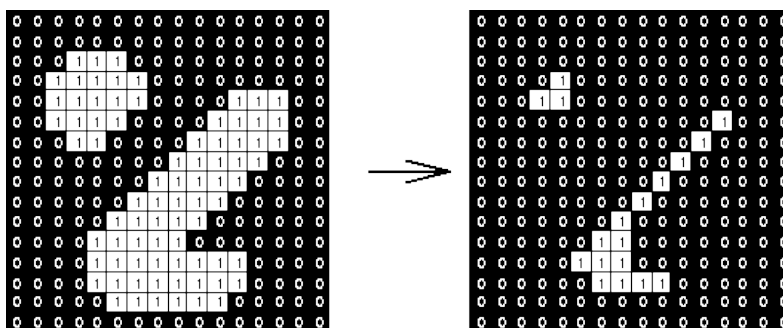


Figura 17 – Erosión [26]

### 2.2.6 Dilatación

En la sección anterior se mencionó la erosión, ahora se procede a explicar la dilatación. Ésta se aplica típicamente a imágenes binarias, pero hay versiones que funcionan en imágenes en escala de grises. El efecto básico del operador en una imagen binaria es alargar gradualmente los límites de las regiones de píxeles de primer plano (es decir, píxeles blancos típicamente). Por lo tanto, las áreas de los píxeles de primer plano crecen mientras que los huecos dentro de esas regiones se vuelven más pequeños. [27]

El operador de la dilatación toma dos piezas de datos como entradas. La primera es la imagen que va a ser dilatada. La segunda es un conjunto de coordenadas (usualmente pequeñas) conocidas como un elemento estructurador (también conocido como núcleo). Es este elemento estructurante el que determina el efecto preciso de la dilatación en la imagen de entrada.

La mayoría de las implementaciones de este operador esperan que la imagen de entrada sea binaria, usualmente con píxeles de primer plano en el valor de píxel 255 y píxeles de fondo en el valor de píxel 0.

En la Figura 18 se muestra la dilatación con un elemento estructurador cuadrado de 3x3 que da un efecto de expansión.

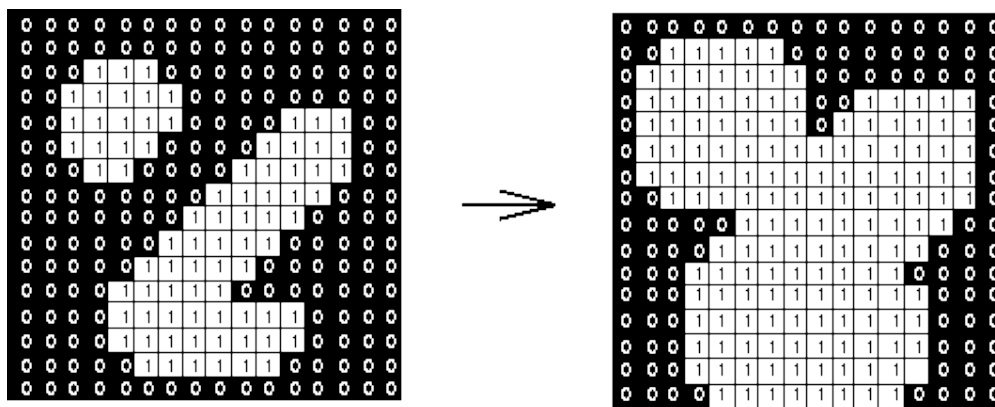


Figura 18 – Dilatación [27]

## 2.3 CARACTERIZACIÓN DE OBJETOS

### 2.3.1 Regiones

Una región en una imagen es un grupo de píxeles conectados que poseen propiedades similares (características). Las regiones son importantes para la interpretación de una imagen porque pueden corresponder a objetos en una escena. Una imagen puede contener varios objetos y a la vez, cada objeto puede contener varias regiones correspondientes a distintas partes del objeto. Para que una imagen pueda ser interpretada con precisión, Debe dividirse en regiones que correspondan a objetos o partes de un objeto. Sin embargo, debido a errores de segmentación, la correspondencia entre regiones y objetos no será perfecta, y el conocimiento específico del objeto debe ser usado en las siguientes etapas para la interpretación de imágenes.

Una vez un conjunto de regiones ha sido identificado, las propiedades de las regiones se vuelven entradas a procedimientos de alto nivel para que puedan realizar una toma de decisiones tales como reconocimiento o inspección. La mayoría de los paquetes de procesamiento de imágenes tienen operadores que pueden producir un conjunto de propiedades para cada región. Las propiedades comunes incluyen propiedades geométricas tales como las áreas de la región, el centroide, los puntos extremos; propiedades de figuras tales como las medidas de circularidad y alargamiento; y propiedades de intensidad tales como Tono gris medio y varias estadísticas de textura.

[28]

### 2.3.2 Caracterización

La caracterización es el proceso de obtener propiedades de una región. Para cada tipo de característica se ejecuta un algoritmo distinto que devuelve los valores obtenidos para la propiedad en cuestión. En esta sección se dan definiciones de algunas de las

características, particularmente aquellas que serán usadas más adelante en el desarrollo de algoritmos de procesamiento de imagen. La Tabla 1 es tomada de la documentación de MATLAB que explica algunas de las características más importantes y los valores que retornan. [29]

Tabla 1 – Características en Matlab

Característica	Descripción
'Area'	Devuelve un escalar que especifica el número real de píxeles en la región.
'Centroid'	Devuelve un vector 1-por-Q que especifica el centro de masa de la región. El primer elemento de es la coordenada horizontal (o coordenada x) del centro de masa, y el segundo elemento es la coordenada vertical (o coordenada y). Todos los demás elementos del 'Centroid' están en orden de dimensión.
'BoundingBox'	Devuelve el rectángulo más pequeño que contiene la región, especificado como un vector 1-por-Q*2, donde Q es el número de dimensiones de la imagen.
'MajorAxisLength'	Devuelve un escalar que especifica la longitud (en píxeles) del eje mayor de la elipse que tiene los mismos segundos momentos centrales normalizados que la región.
'MinorAxisLength'	Devuelve un escalar que especifica la longitud (en píxeles) del eje menor de la elipse que tiene los mismos segundos momentos centrales normalizados que la región.
'Perimeter'	Devuelve un escalar que especifica la distancia alrededor del límite de la región. Se calcula el perímetro calculando la distancia entre cada par adyacente de píxeles alrededor del borde de la región. Si la imagen contiene regiones no contiguas, pueden surgir resultados inesperados.
'Orientation'	Retorna los grados de inclinación con respecto a la horizontal.

'Eccentricity'	Es la razón de la distancia entre el foco de la elipse y su eje mayor.
'Equivalent diameter squared'	Retorna los diámetros equivalentes al cuadrado (eje menor y mayor).
'Extent'	Retorna la división del área del objeto sobre el área de su cuadro envolvente.

## 2.4 FUNDAMENTOS DE PROBABILIDAD Y ESTADÍSTICA

### 2.4.1 Conceptos

Probabilidad es la medida de la posibilidad de que ocurra un evento. La probabilidad se cuantifica como un número entre 0 y 1, donde 0 indica imposibilidad y 1 indica certeza. Cuanto mayor sea la probabilidad de un evento, más seguro será que ocurrirá el evento.

Media: La media de una variable aleatoria discreta  $X$  es un promedio ponderado de los posibles valores que puede tomar la variable aleatoria. La media de una variable aleatoria pesa cada resultado  $X_i$  según su probabilidad  $P_i$ .

Varianza es una medida de la dispersión de una variable aleatoria alrededor de su media. Siendo el valor esperado de un número cuadrado, la varianza es siempre positiva. Cuando una variable aleatoria  $X$  es constante (pase lo que pase, siempre toma el mismo valor), entonces su varianza es cero (porque  $X$  es siempre igual a su valor esperado). Por el contrario, cuanto mayores son las posibles desviaciones de  $X$  respecto a su valor esperado, mayor es la varianza de  $X$ . [30]



### 2.4.2 Tipos de distribuciones

Existen muchos tipos de distribuciones. En esta sección se muestran sólo dos tipos: la distribución normal y la binomial. La Figura 21 detalla un diagrama de decisión para ayudar a escoger el tipo de distribución adecuado siguiendo unas observaciones generales y también muestra otros tipos de distribuciones.

Distribución normal: Es una distribución continua que surge en muchos procesos naturales. "Continua" significa que entre dos valores de datos cualquiera se podría encontrar otro valor intermedio. Por ejemplo, las alturas de los hombres varían continuamente y son el resultado de tantas influencias aleatorias minúsculas que la distribución total de las alturas de los hombres en América es muy cercana a la normal. En la Figura 19 se muestra algunas formas de distribución normal.

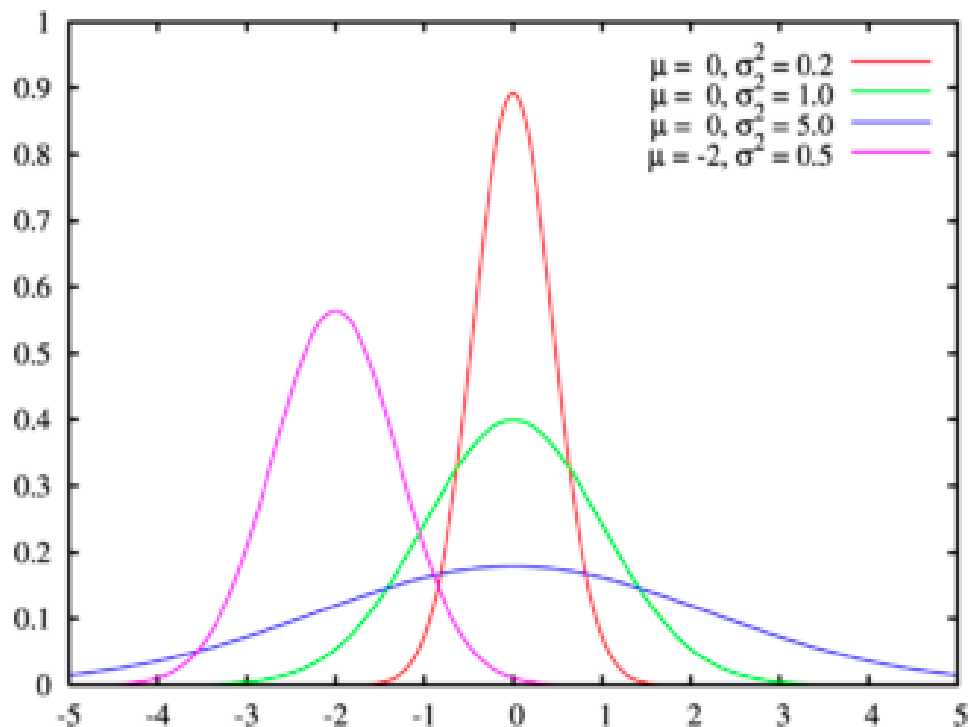


Figura 19 – Distribución Normal [31]

Distribución binomial: La distribución binomial mide las probabilidades del número de éxitos sobre un número dado de pruebas con una probabilidad especificada de éxito en cada intento. Una distribución binomial es muy distinta de una distribución normal, y si el tamaño de muestra es suficientemente grande, las formas pueden ser bastante familiares. La diferencia clave es que la distribución binomial es discreta, no continua. Ver Figura 20. [31]

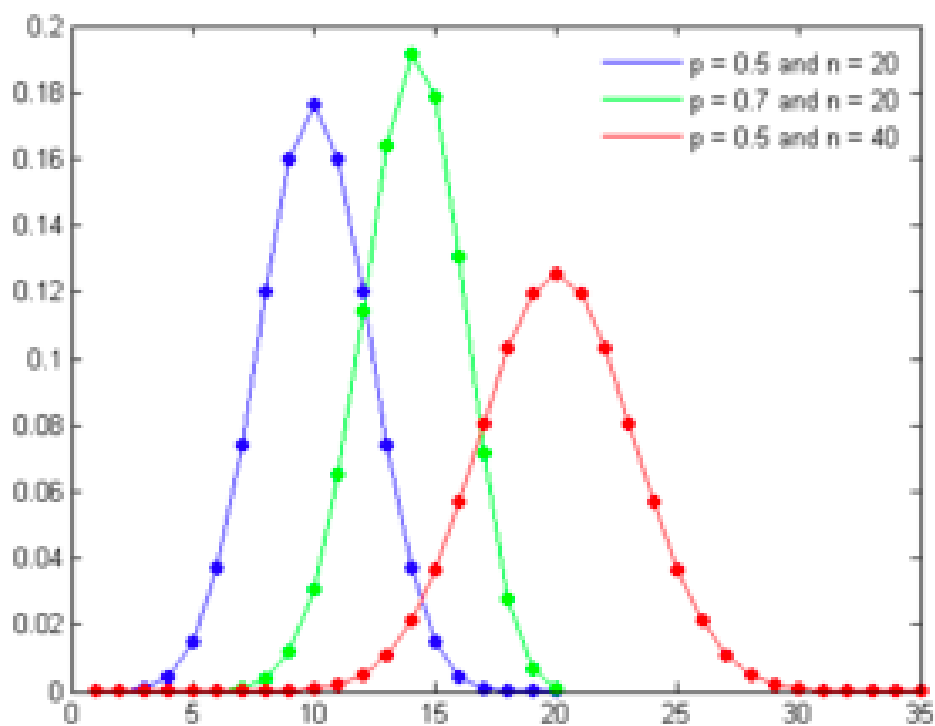


Figura 20 – Distribución Binomial [31]

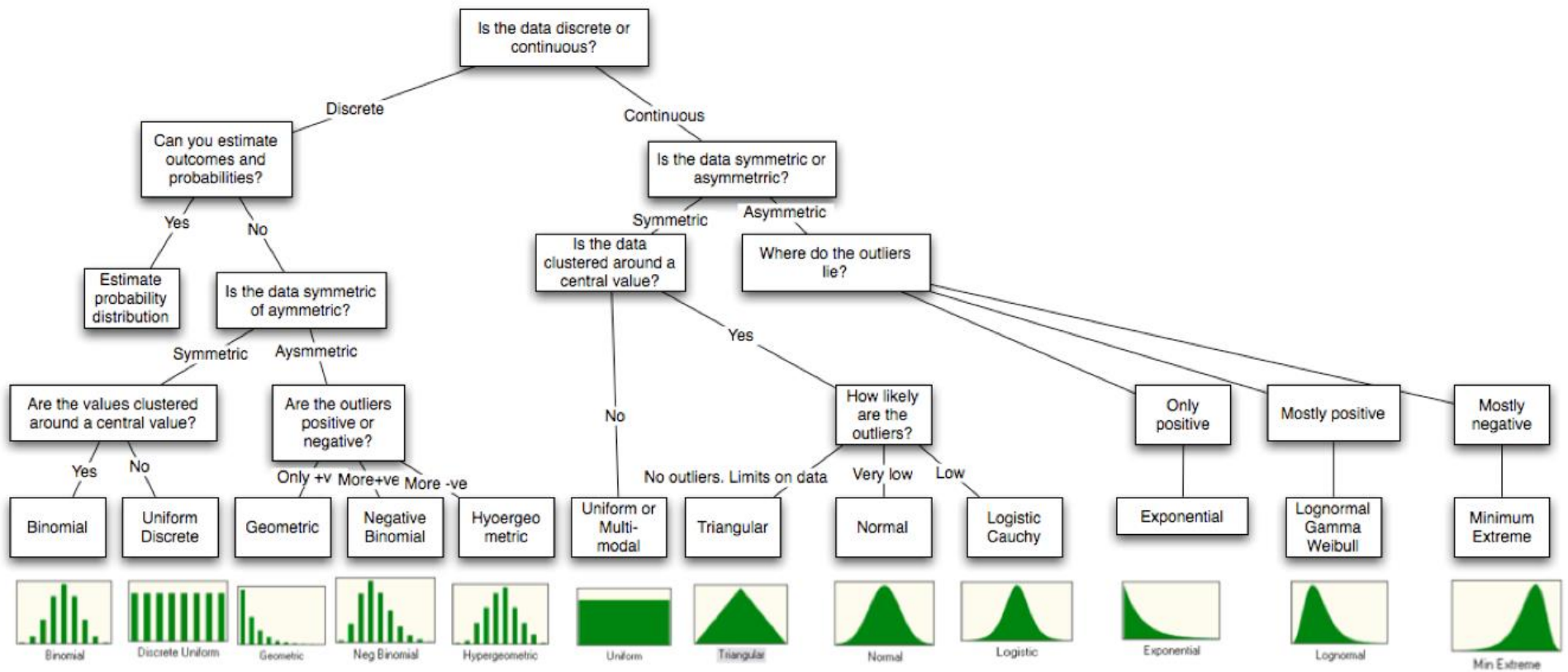


Figura 21 – Guías para elegir tipo de distribución [31]

### 2.4.3 Regla 68 – 95 – 99.7

Es una abreviatura utilizada para recordar el porcentaje de valores que se encuentran dentro de una banda alrededor de la media en una distribución normal con un ancho de dos, cuatro y seis desviaciones estándar, respectivamente; Con mayor precisión, el 68.27%, 95.45% y 99.73% de los valores se encuentran dentro de una, dos y tres desviaciones estándar de la media, respectivamente. De esta manera se puede establecer con seguridad que prácticamente el 100% de los datos se encontrarán a 3 desviaciones estándar de la media. Ver Figura 22. [32]

En la notación matemática, estos hechos pueden expresarse de la siguiente manera, donde  $X$  es una observación de una variable aleatoria normalmente distribuida,  $\mu$  es la media de la distribución y  $\sigma$  es su desviación estándar:

$$P_r(\mu - \sigma \leq X \leq \mu + \sigma) \approx 0.6827$$

$$P_r(\mu - 2\sigma \leq X \leq \mu + 2\sigma) \approx 0.9545$$

$$P_r(\mu - 3\sigma \leq X \leq \mu + 3\sigma) \approx 0.9973$$

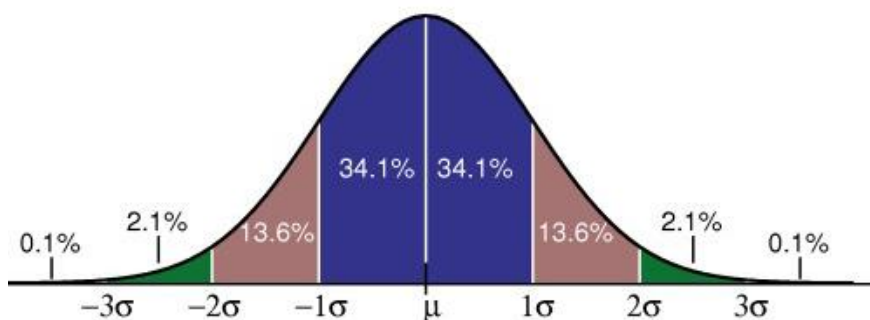


Figura 22 – La regla 68-95-99.7 [32]

## 2.5 RECONOCIMIENTO ESTADÍSTICO DE PATRONES (REP)

### 2.5.1 Introducción

El reconocimiento de patrones es un campo de estudio desarrollado significativamente en los años 60. Era un tema interdisciplinario, cubriendo desarrollos en las áreas de estadística, ingeniería, inteligencia artificial, informática, psicología y fisiología, entre otras. Algunas personas entraron al campo para resolver un problema real. El gran número de aplicaciones, abarcando desde las clásicas como reconocimiento automático de caracteres y diagnóstico médica a las más recientes en la minería de datos (como la puntuación de crédito, análisis de ventas al consumidor y análisis de transacciones de tarjetas de crédito), han atraído un considerable esfuerzo de investigación, con muchos métodos desarrollados y avances hechos. Otros investigadores fueron motivados por el desarrollo de máquinas con rendimiento parecido al cerebro humano, que en alguna manera podía emular el rendimiento humano.

Se ha hecho un significativo progreso dentro de estas áreas, particularmente, donde el dominio se superpone con estadísticas y probabilidad, y en los años recientes ha habido muchos progresos emocionantes, tanto en metodología como en aplicación. Estos edifican sobre los fundamentos sólidos de investigaciones pasadas y toman ventaja de los recursos computacionales mejorados ampliamente disponibles hoy en día. Estos avances incluyen, por ejemplo, los métodos basados en núcleos y métodos computacionales bayesianos. [33]

### 2.5.2 Clasificación supervisada versus no supervisada

Hay dos divisiones principales de la clasificación: clasificación supervisada (o discriminación) y clasificación sin supervisión (algunas veces simplemente referida como clasificación o agrupación en la literatura estadística). [33]

En la clasificación supervisada se tiene un conjunto de muestras de datos (cada uno consistiendo de medidas en un conjunto de variables) con etiquetas asociadas, los tipos de clase. Estos se utilizan como ejemplares en el diseño del clasificador.

En la clasificación sin supervisión, los datos no están etiquetados y se busca encontrar grupos en los datos y las características que distinguen un grupo de otro. Técnicas de agrupamiento también pueden ser usadas como parte de un esquema de clasificación supervisada mediante la definición de prototipos. Un esquema de agrupamiento puede ser aplicado a los datos de cada clase por separado y muestras representativas para cada grupo dentro de la clase (por ejemplo, las medias del grupo) usado como el prototipo para esa clase.

La Figura 23 muestra el proceso general de la obtención de datos, selección de características y clasificación de éstas.

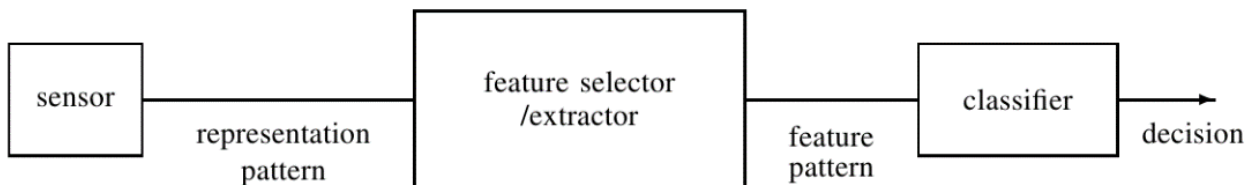


Figura 23 – Esquema general de un clasificador de patrones [33]

### 2.5.3 Clasificador bayesiano Ingenuo

La técnica del clasificador bayesiano ingenuo está basada en el teorema Bayesiano y es particularmente apropiado cuando la dimensionalidad de las entradas es demasiado alta. A pesar de su simplicidad, el clasificador bayesiano ingenuo usualmente puede sobrepasar a métodos más sofisticados de clasificación. [34]

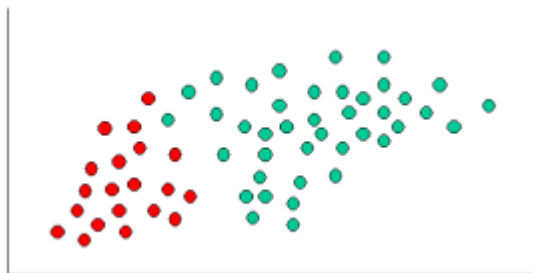


Figura 24 – gráfico de dispersión [34]

Para demostrar el concepto de la clasificación bayesiana ingenua considere el ejemplo demostrado en la Figura 24. Como se indica, los objetos pueden ser clasificados como VERDE o ROJO. Se busca clasificar casos nuevos apenas lleguen, es decir, decidir a qué clase de etiqueta pertenecen, basándose en los objetos actualmente existentes.

Como hay dos veces más objetos VERDES que ROJOS, es razonable creer que un caso nuevo (que no ha sido observado aún) es dos veces más posible que tengan asociación con los VERDES que con los ROJOS. En el análisis bayesiano, esta creencia es conocida como la probabilidad anterior. Las probabilidades anteriores están basadas en experiencias previas, en este caso el porcentaje de objetos VERDES y ROJOS, y a menudo usado para predecir resultados antes de que ocurran. Así que se puede decir:

$$\text{Probabilidad anterior de VERDE} \propto \frac{\text{Número de objetos VERDES}}{\text{Número total de objetos}}$$

$$\text{Probabilidad anterior de ROJOS} \propto \frac{\text{Número de objetos ROJOS}}{\text{Número total de objetos}}$$

Como hay un total de 60 objetos y 40 de ellos son VERDES y 20 ROJOS, las probabilidades anteriores para asociación a una clase son:

$$\text{Probabilidad anterior para VERDE} \propto \frac{40}{60}$$

$$\text{Probabilidad anterior para ROJO} \propto \frac{20}{60}$$

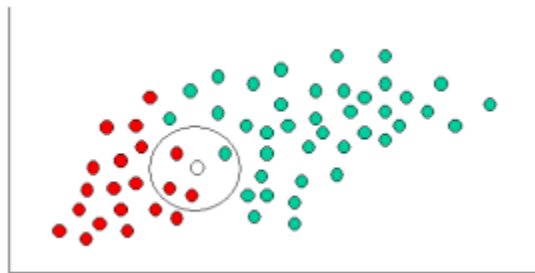


Figura 25 – Clasificación de un objeto nuevo (Blanco). [34]

Una vez calculada la probabilidad anterior, se puede empezar a clasificar un objeto nuevo (círculo blanco) como se muestra en la Figura 25. Como los objetos están propiamente agrupados, es razonable asumir que mientras más objetos VERDES (o ROJOS) en la proximidad de X, hay más probabilidades que el caso nuevo pertenezca a ese color en particular. Para medir la posibilidad, se dibuja un círculo alrededor de X que englobe un número (a elegir previamente) de puntos independientemente de sus etiquetas de clase. Entonces se calcula el número de puntos en el círculo perteneciente a cada etiqueta de clase. De esto se calcula la posibilidad:

$$\text{Posibilidad de } X \text{ dado VERDE} \propto \frac{\text{Número de VERDE en la proximidad de } X}{\text{Número total de casos VERDES}}$$

$$\text{Posibilidad de } X \text{ dado ROJO} \propto \frac{\text{Número de ROJO en la proximidad de } X}{\text{Número total de casos ROJOS}}$$

La posibilidad de X dado VERDE es menor a la posibilidad de X dado ROJO, ya que el círculo solo engloba 1 objeto VERDE y 3 ROJOS. Así que:

$$\text{Probabilidad de } X \text{ dado VERDE} = \frac{1}{40}$$

$$\text{Probabilidad de } X \text{ dado ROJO} = \frac{3}{20}$$

A pesar de que las probabilidades anteriores indicaban que X podía pertenecer a VERDE (ya que hay dos veces más VERDE comparado al ROJO) la posibilidad posterior indica lo contrario; que la asociación de clase de X es ROJO (ya que hay más objetos ROJOS en la proximidad de X que VERDES). En el análisis bayesiano, la clasificación



final es producida a través de la combinación de ambas fuentes de información, es decir, la probabilidad anterior y posterior, para formar una probabilidad total usando la regla de Bayes (nombrada en honor al reverendo Thomas Bayes 1702-1761).

*Probabilidad de X ser VERDE  $\propto$  Probabilidad anterior  $\times$  Posibilidad posterior*

$$= \frac{4}{6} \times \frac{1}{40} = \frac{1}{60}$$

*Probabilidad de X ser ROJO  $\propto$  Probabilidad anterior  $\times$  probabilidad posterior*

$$= \frac{2}{6} \times \frac{3}{20} = \frac{1}{20}$$

Finalmente, se clasifica a X como ROJO ya que su clase de asociación logra la probabilidad posterior más grande. Las probabilidades de arriba no están normalizadas. Sin embargo, esto no afecta el resultado de la clasificación ya que las constantes de normalización son las mismas.

Esto es muy importante y en el desarrollo ingenieril se utilizará esta teoría para construir el clasificador usado en los algoritmos.

### 3 DISEÑO INGENIERIL

#### 3.1 REQUERIMIENTOS TÉCNICOS

##### 3.1.1 Geometría

En esta sección se calcula el eje x y la altura a la que se debe encontrar la cámara para identificar los objetos. Las dimensiones del espacio de trabajo deben tener la amplitud suficiente para acomodar la herramienta a reconocer de mayor dimensión. El peso y las dimensiones de las herramientas seleccionadas se detallan en la Tabla 2.

Tabla 2 – Peso y dimensiones de las herramientas

Herramienta	Peso (g)	Base (cm)	Altura (cm)
Destornillador	20	9.2	0.9
Alicate	40	10	6
Regla	10	7.3	2.5
Cinta	5	4	4
Llave Inglesa (cabeza)	100	5.3	4.3

Según la Tabla 2 se observa que la herramienta más grande es el alicate con unas dimensiones de 10 cm x 6 cm. Usando un factor de seguridad del 30% se calcula la distancia del eje x del espacio de trabajo:

$$d_x = d_m * FS$$

$d_x$  = Distancia eje x

$d_m$  = Dimensión Máxima

$FS$  = Factor de seguridad

$$d_x = 10 \text{ cm} * 1.30 = 13 \text{ cm}$$

El campo de visión vertical depende de la geometría óptica de la cámara y tiene una relación fija con respecto al campo de visión horizontal. Por lo tanto, no es una variable.

Se procede a calcular la altura  $h$  en base a la distancia  $d_x$ . Según la hoja de datos técnicos de la RaspiCam (ver Anexos A – Especificaciones técnicas de la RaspiCam), el área de visión de la cámara tiene un ángulo de apertura vertical  $\alpha_v$  de 41.41 +/- 0.11 grados y un ángulo de apertura horizontal  $\alpha_h$  de 53.50 +/- 0.13 grados. Esto forma una pirámide con una altura, base y ancho definidos. Se calcula  $\beta$  con un triángulo isósceles.

$$180^\circ = \alpha_h + 2\beta_h$$

$$\beta_h = \frac{180^\circ - 53.50^\circ}{2} = 63.3^\circ$$

Se puede calcular la altura  $h$  para la cual se cumplen estas condiciones usando la ley de senos,

$$\frac{\sin(\beta_h)}{h} = \frac{\sin\left(\frac{\alpha_h}{2}\right)}{\frac{d_x}{2}}$$

$$h = \frac{\sin \beta_h * \frac{d_x}{2}}{\sin\left(\frac{\alpha_h}{2}\right)} = \frac{\sin(63.3^\circ) * \frac{13 \text{ cm}}{2}}{\sin\left(\frac{53.50^\circ}{2}\right)} = 17.19 \text{ cm}$$

Se observa entonces que la cámara debe ser ubicada a una altura de 17.19 cm con respecto al espacio de trabajo para obtener una distancia en el eje x de 13 cm. Con estos cálculos finaliza toda la parte del diseño geométrico, es decir, las características de posición y distancia entre la cámara y los objetos.

Una consideración importante a tener en cuenta es que la RaspiCam usa un lente de 35 mm fijo con una distancia focal de 60 cm en adelante, es decir, todo lo que se encuentre a menos de 60 cm se verá borroso. Para solucionar este problema es necesario adicionar otro lente que pueda acercar el punto focal a la altura calculada. Sin embargo, en esta aplicación particular no es necesario ya que es tolerable una imagen desenfocada.

### 3.1.2 Banda transportadora

En esta sección se calcula la potencia y torque del motor basándose en los requisitos de velocidad de vacío y peso máximo. Se desea que la banda opere a una velocidad sin carga de 0.15 m/s capaz de mover hasta un peso máximo de 3 kg. Con estas condiciones se procede a calcular la aceleración y potencia.

La aceleración lineal se calcula como el tiempo en que tarda la banda para pasar de una velocidad inicial a una final,

$$A = \frac{V - V_0}{t}$$

$V$  = Velocidad (m/s)

$V_0$  = Velocidad inicial (m/s)

$t$  = Tiempo (s)

Asumiendo que se desea pasar de una velocidad de 0 m/s a 0.15 m/s en 20 ms, se tiene que:

$$A = \frac{0.15 \text{ m/s} - 0 \text{ m/s}}{0.020 \text{ s}} = \frac{7.5 \text{ m}}{\text{s}^2}$$

Obtenida la aceleración se puede calcular la potencia mecánica del motor que se necesita:

$$P_m = \frac{T}{t}$$

$P_m$  = Potencia mecánica (J/s) ó (W)

$T$  = Trabajo realizado (J).

$t$  = Tiempo en que se realiza el trabajo (s).

$$T = F * D$$

$F$  = Fuerza (N)

$D$  = Distancia (m)

$$F = m * A$$

$m$  = Masa (Kg)

$A$  = Aceleración (m/s<sup>2</sup>)

Despejando la potencia mecánica se obtiene,

$$P_m = \frac{m * A * D}{t} = m * A * V$$

$$P_m = 3 \text{ kg} * \frac{7.5 \text{ m}}{\text{s}^2} * \frac{0.15 \text{ m/sm}}{\text{s}} = 3.375 \text{ W}$$

Por lo que se necesita un motor eléctrico de al menos 3.375 W de potencia. Para hallar la corriente que consume se puede calcular con la fórmula de potencia eléctrica,

$$P_m = P_E = V * I$$

$P_E$  = Potencia Eléctrica en Watts (W)

$V$  = Voltaje (V)

$I$  = Corriente (A)

Por principio de la conservación de energía se puede igualar la potencia mecánica a la potencia eléctrica. Tomando como referencia el valor estándar de 12 V para un motor de esta potencia se obtiene que la corriente consumida es:

$$I = \frac{P_E}{V} = \frac{3.375 \text{ W}}{12 \text{ V}} = 0.281 \text{ A}$$

Para manipular un motor de 3.375 W a 12 V con corriente de 0.281 A se escoge la tarjeta controladora de motores L298N, capaz de manipular hasta 2 motores de 2A con 46 V cada uno. Esta tarjeta será detallada en el ítem 3.2.4.

El torque del motor se puede calcular con la potencia hallada. Es importante tener en cuenta que cuando se habla del torque del motor realmente se refiere al torque del conjunto motor y sistema de transmisión, pues hay motores que incorporan una caja de reducción, por ejemplo los motorreductores. Esto significa que el motor puede tener un torque bajo y una alta velocidad, que pasa a través de una relación reductora de engranajes para aumentar el torque y reducir la velocidad a los niveles deseados.

$$P_m = M\omega$$

$M$  = Torque o par motor (N\*m)

$\omega$  = Velocidad angular (Rad/s)

Asumiendo un eje estándar de 11/16", se tiene un radio de 8.731 mm, y con la velocidad lineal definida anteriormente se puede calcular la velocidad angular por medio de la ecuación:

$$\omega = \frac{V}{r}$$

$r$  = Radio del eje (m)

$$\omega = \frac{0.15 \text{ m/s}}{0.008731 \text{ m}} = 17.80 \text{ Rad/s}$$

Finalmente, el torque del motor es:

$$M = \frac{P_m}{\omega} = \frac{3.375 \text{ W}}{17.80 \text{ Rad/s}} = 0.1896 \text{ Nm} = 1.933 \text{ kg} * \text{cm} \approx 2 \text{ kg.cm}$$

Adicionalmente hay que tener en cuenta varios factores como pérdidas por eficiencia, por fricción y rozamiento de componentes, por corrientes de fuga, márgenes de error y demás factores que causan incertidumbre. Por esta razón se aplica un factor de seguridad del 20% para concluir que se necesita entonces un motor con un torque de al menos 2.4 kg.cm con una potencia de 3.375 W.

### 3.1.3 Iluminación

Esta sección define la configuración de iluminación a usar. De los tipos de luz vistos en 2.1.2 se escoge la iluminación LED por su buena distribución de la gama de colores y su facilidad de implementación con tarjetas de desarrollo como la RPI. Los otros tipos de luz modifican mucho los colores entre los reales y lo que se visualiza en pantalla. De las técnicas de luz vistas en 2.1.6 se observa que:

- La luz directa es la de mayor potencia y más fácil implementación pero genera brillos y sombras no deseados, y no es uniforme.
- La luz indirecta no genera brillos ni sombras pero llega con mucha menor intensidad. Genera un campo uniforme dependiendo de la configuración.
- La luz difusa de campo brillante ilumina de manera uniforme el espacio de trabajo, genera buen contraste entre los bordes y el fondo. Se pierde un poco los relieves pero se visualiza bien la silueta. No genera brillos ni sombras. Tiene mayor potencia que la indirecta pero requiere filtros costosos para difuminar la luz.
- La contraluz es la más apropiada para esta aplicación porque genera excelente contrastes pero su implementación es complicada ya que las herramientas están sobre una banda transportadora.
- La luz de campo oscuro también es viable pero resulta un poco más costosa por el uso de filtros y materiales especiales.

Por lo tanto se opta por la luz indirecta o la luz difusa. Ambas son adecuadas.

Para calcular la potencia necesaria de los leds se utiliza el método de los lúmenes donde se debe conocer las dimensiones del espacio a iluminar y unos coeficientes relacionados con la utilización y el mantenimiento. [35]

$$\Phi = \frac{E * S}{\eta * f_m}$$

$\Phi$  = Flujo luminoso total

$E$  = Iluminancia media deseada

$S$  = Superficie del plano de trabajo

$\eta$  = Factor de utilización

$f_m$  = Factor de mantenimiento

Para hallar la iluminancia media deseada se debe comparar con algunos valores de lux recomendados para habitaciones estándares:

Tabla 3 – Valores típicos de luminosidad para habitaciones. [36]

Lugar	Luminosidad (Lux)
Sala.	150 – 250
Cocina	300 – 400
Recámara o dormitorio	150 – 250
Escaleras	150 – 250
Estudios	300 – 500
Cuartos de Aseo	150 – 200
Zonas de circulación	100 – 200

En general se necesita un alto nivel de lux para la identificación de imágenes por lo que se escoge unos niveles similares al del estudio, de 300-500 lux para el valor de iluminancia media deseada.

La superficie de trabajo es de 13 cm x 14 cm.

Para calcular el factor de utilización se deben conocer el coeficiente de reflexión, y el índice del local. El coeficiente de reflexión de techo, paredes y suelo se obtiene de la

Tabla 4. Todas las paredes y techo serán construidas en MDF, por lo que se consideran como de color medio claro. El suelo es de color negro conformado por el material de la banda transportadora. Se selecciona 0.3 para el techo, 0.3 para las paredes y 0.1 para el suelo.

Tabla 4 – Factor de reflexión según el color del material. [37]



	Color	Factor de reflexión
Techo	Blanco o muy claro	0.7
	Claro	0.5
	Medio	0.3
Paredes	Claro	0.5
	Medio	0.3
	Oscuro	0.1
Suelo	Claro	0.3
	Oscuro	0.1

El índice del local se calcula como:

$$k = \frac{a * b}{h * (a + b)}$$

$a$  = Ancho = 0.34 m


$b$  = Largo = 0.22 m

$h$  = Altura = 17.19 cm

$$k = \frac{0.34 \text{ m} * 0.22 \text{ m}}{0.18 \text{ m} * (0.34 \text{ m} + 0.22 \text{ m})} = 0.742$$

Con estos factores se puede calcular el factor de utilización utilizando la Tabla 5. Extrapolando los datos se toma 0.16.

Tabla 5 – Tabla del factor de utilización. [37]

Tipo de aparato de alumbrado	Índice del local k	Factor de utilización ( $\eta$ )								
		Factor de reflexión del techo								
		0.7			0.5			0.3		
		Factor de reflexión de las paredes								
		0.5	0.3	0.1	0.5	0.3	0.1	0.5	0.3	0.1
	1	.28	.22	.16	.25	.22	.16	.26	.22	.16
	1.2	.31	.27	.20	.30	.27	.20	.30	.27	.20
	1.5	.39	.33	.26	.36	.33	.26	.36	.33	.26
	2	.45	.40	.35	.44	.40	.35	.44	.40	.35
	2.5	.52	.46	.41	.49	.46	.41	.49	.46	.41
	3	.54	.50	.45	.53	.50	.45	.53	.50	.45
	4	.56	.52	.46	.55	.52	.46	.56	.52	.46
	5	.63	.60	.56	.63	.60	.56	.62	.60	.56
	6	.68	.63	.60	.66	.63	.60	.65	.63	.60
	8	.71	.67	.64	.69	.67	.64	.68	.67	.64
10	.72	.70	.67	.71	.70	.67	.71	.70	.67	

El factor de mantenimiento depende de si el ambiente es sucio o limpio. Se toma 0.6 para ambiente sucio.

Finalmente, se reemplazan todos los valores hallados en la ecuación de flujo total obteniendo:

$$\Phi = \frac{E * S}{\eta * f_m} = \frac{300 \text{ Lux} * 0.13 \text{ m} * 0.14 \text{ m}}{0.16 * 0.6} = 56.87 \text{ lm}$$

Con este valor de lm se puede conocer entonces cuántos LEDs se deben usar y de cuánta potencia. Para esto se consulta la Tabla 6. Con esto se observa que un Led de 3 W cubre los requerimientos de luminosidad. Se opta por instalar 2 LEDs para mantener una luz simétrica en el eje longitudinal, pues instalando un solo LED en un lado o esquina, quedaría una iluminación no uniforme.

Tabla 6 – Valores típicos de lúmenes para tipos de iluminación [37]

Lumens (lm)	Bombillas Incandescentes	Fluorescentes (CFL)	Tecnología LED
250 (lm)	25 W	4-9 W	3 W
450 (lm)	40 W	9-13 W	4-5 W
800 (lm)	60 W	13-15 W	6-8W
1.100 (lm)	75 W	18-25 W	9-13 W
1.600 (lm)	100 W	23-30 W	16-20 W
2.000 (lm)	125 W	28-40 W	20-25 W
2.600 (lm)	150 W	30-55 W	25-28 W

Para aislar lo más posible el sistema de las fuentes de luz externas como el sol, lámparas, y demás, se decide construir un cuarto con iluminación LED donde se identifican las piezas, este cuarto debe tener un área mayor a la superficie de trabajo calculada. Todo lo anterior con el fin de:

- Maximizar el contraste en los puntos de interés.
- Minimizar el contraste en el resto de la muestra.
- Proporcionar una medida de robustez.

## 3.2 HARDWARE UTILIZADO

En el diseño ingenieril se hablará de todo el proceso de construcción del proyecto. Comenzando por mencionar los elementos más relevantes de hardware y software utilizado. Luego se procederá a describir el proceso de su diseño y construcción comenzando por la parte mecánica, luego por la electrónica y por último terminando con la parte de programación.

En esta sección se procederá a describir todos los elementos importantes usados durante la construcción de este proyecto haciendo énfasis en aquellos que fueron primordiales para la construcción del proyecto. En la última parte, “otros”, se hace mención a otros elementos usados en la construcción pero que no fueron de vital impacto o importancia. Su uso o no uso no causa un impacto directo en el proyecto y por tanto se les mencionará brevemente con menor énfasis.

### 3.2.1 Raspberry Pi 3B (RPI)

Éste es el embebido donde está el corazón del proyecto. Todos los programas y los algoritmos desarrollados fueron creados en el software Simulink, posteriormente exportados al RPI para que su ejecución pueda correr nativamente en el embebido sin necesidad de los instrumentos de desarrollo o comunicación como el computador o el celular. Así mismo como es un componente tan vital se procederá a mencionar un poco de su historia, sus características técnicas y componentes. [38]

El Raspberry Pi es un computador de una única tarjeta que tiene un tamaño tan pequeño como una tarjeta de crédito, creado por la Raspberry Pi Foundation. La idea de la producción del Raspberry Pi comenzó en 2006 con la generalización de que las generaciones de jóvenes no tienen conocimientos sobre cómo operan las computadoras. Por lo tanto, un grupo de académicos e ingenieros de la Universidad de Cambridge

decidió desarrollar una computadora muy pequeña que todos puedan comprar para crear un ambiente de aprendizaje en la programación. El proyecto Raspberry Pi se convirtió en uno muy prometedor con la aparición de procesadores móviles baratos y potentes con muchas características avanzadas que permitieron hacer posible el desarrollo del Raspberry Pi, que se continuó con la creación de la Raspberry Pi Foundation con el primer producto lanzado en 2012 [39]. En la Figura 26 se muestra el RPI.

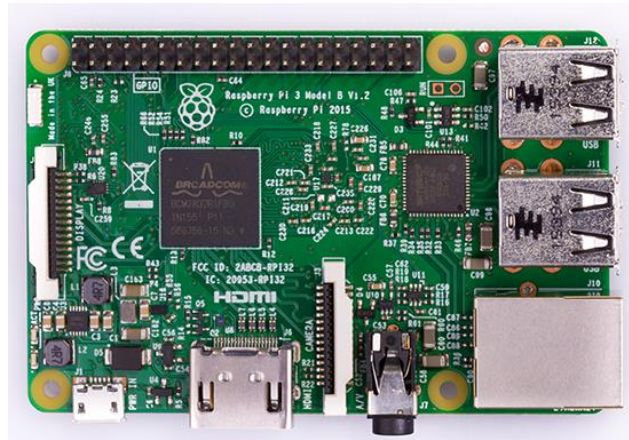


Figura 26 – Raspberry Pi 3B. [39]

Figuras, especificaciones y características de hardware y software del Raspberry Pi tomados directamente del fabricante. [40] Hay varios modelos en existencia a la fecha: Raspberry Pi 3 B (Éste es el usado en la tesis), Raspberry Pi Zero W, Raspberry Pi Zero, Raspberry Pi 2, Raspberry Pi modelo B y Raspberry Pi A+ Raspberry Pi. De aquí en adelante todo lo mencionado se referirá específicamente al modelo usado en esta tesis salvo que se indique lo contrario. El pinout del RPI se muestra en la Figura 27.

#### Características técnicas:

- Una CPU quad-core 1.2GHz 64-bit ARMv8
- LAN Inalámbrico 802.11n
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)
- 1GB RAM
- 4 puertos USB

- 40 pines GPIO
- Puerto full HDMI
- Puerto Ethernet
- Jack combinado 3.5mm audio y composite video
- Camera interface (CSI)
- Display interface (DSI)
- Ranura de tarjeta Micro SD

Al principio se pensaba usar un modo gráfico VPN, pero después se descartó esta idea para aligerar la carga de procesamiento en el RPI para maximizar el uso de los recursos disponibles.

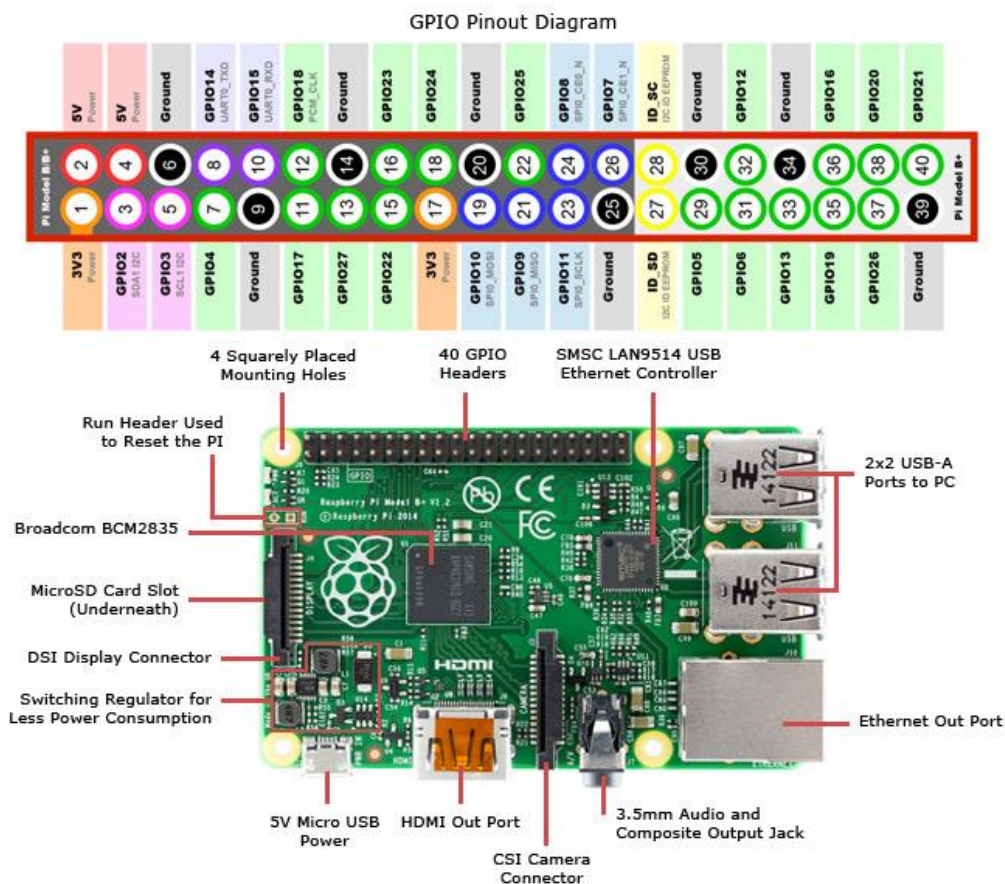


Figura 27 – Esquemático Raspberry Pi [41]

### 3.2.2 RaspiCam

Con esta cámara diseñada específicamente para el RPI fue posible la adquisición constante de imágenes para su posterior procesamiento. Con ella se tomaron las fotos usadas para los algoritmos de entrenamiento de clases, es decir, clasificación supervisada, y para ejecutar el algoritmo de reconocimiento. Así mismo, con la cámara se cerró el lazo para el control de posición de la banda transportadora, es decir, se utilizó como un sensor de luz utilizado para la retroalimentación de la posición de los objetos en la banda transportadora. El tema de control se entrará en detalle en la sección de programación más adelante. [42]

Especificaciones y características de hardware y software de la Raspicam tomadas directamente del fabricante. La RaspiCam se puede ver en la Figura 28. Toda la información técnica relevante de la RaspiCam se puede ver en Anexos A.

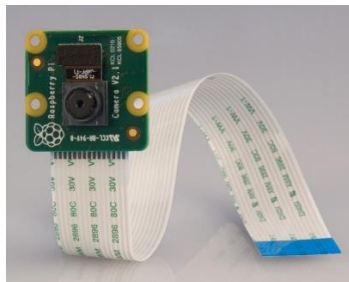


Figura 28 – Raspicam [42]

### 3.2.3 Fuente de energía oficial Raspberry PI

El RPI puede ser alimentado con cualquier fuente de energía que garantice al menos 5V 900mA. Para la mayoría de aplicaciones esto está bien y satisface los requerimientos de energía, sin embargo, el RPI viene equipado con una compuerta Schmitt Trigger que cuando detecta un voltaje menor a 4.75V o mayor a 5.25V se dispara y hace que el RPI entre un modo seguro [43]. La fuente de energía oficial se muestra en la Figura 29.



Figura 29 – Cargador oficial Raspberry Pi 5V 2.5A [43]

Este modo fue diseñado para proteger el sistema cuando hay niveles de tensión no ideales. Aquí el RPI entra a operar a menor velocidad pero se mantiene estable. El problema, dado la alta necesidad de que sea determinístico y constante los procesos de este proyecto, radica en que cualquier variación del voltaje (incluyendo ruido) hace que los algoritmos funcionen de manera imprevista. Esto genera una gran cantidad de error adicional difícil de controlar.

Para minimizar o eliminar este error innecesario se probaron varias fuentes de energía. De entre todos los que se probaron, solamente el cargador oficial logró estabilizar la operación del RPI haciendo que los procesos funcionaran constantes y relativamente determinísticos en el tiempo. Por esta razón es que se destaca este ítem.

#### 3.2.4 Controlador de Motores L298N

Éste es una tarjeta controladora para el manejo de motores de corriente continua con el driver L298N. Con éste se puede manejar dos motores con un consumo máximo de corriente de 2A cada uno y hasta 46V de voltaje de alimentación. Se utiliza para dar el cambio de marcha al motor, manipulando la dirección y velocidad de la banda transportadora. Las conexiones eléctricas se detallarán más adelante en su respectiva sección. [44] La tarjeta controladora de motores se muestra en la Figura 30.

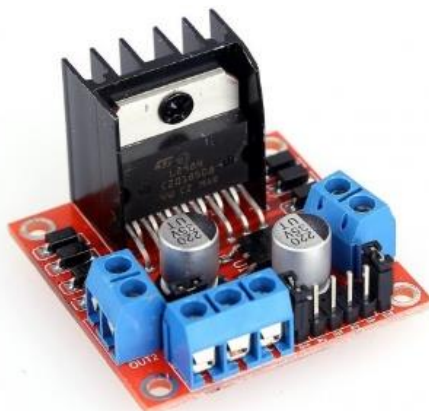


Figura 30 – Tarjeta controladora de motores L298N [44]

El funcionamiento se puede apreciar en la tabla de funcionamiento según las condiciones de las entradas. Imágenes y especificaciones tomadas directamente de un distribuidor en Bogotá [44]. El diagrama y funcionamiento mostrados en la Tabla 7.

Tabla 7 – Esquema y funcionamiento del controlador de motores.

Entradas		Función
$V_{en} = H$	$C = H ; D = L$	Adelante
	$C = L ; D = H$	Reversa
	$C = D$	Parada rápida del motor
$V_{en} = L$	$C = X ; D = X$	Libre, motor quieto.
L = Bajo    H = Alto    X = No importa		

### 3.2.5 Otros

Otros componentes de menor relevancia, puesto que no tienen un impacto directo sobre el funcionamiento del proyecto incluyen:

Fuentes de energía 5V 2A y 12V 2A: El de 5V se utilizó para alimentar los Leds y los ventiladores ya que tiene una potencia de 10W mientras que el consumo necesario



era de 6W. Por otro lado, al motor se le asignó la fuente de 12V ya que es un motor que funciona a este voltaje. Estas fuentes, junto con la oficial del RPI están montadas en una extensión que incluye protecciones eléctricas. Ver Figura 31.



Figura 31 – Extensión Forza con las 3 fuentes de energía utilizadas.

Pantalla táctil LCD: En el primer prototipo se utiliza una pantalla LCD para visualizar la ejecución del proceso. Más adelante se descartó la idea de usar una pantalla LCD porque el ambiente de desarrollo no ofrece métodos y funciones para el manejo de la pantalla. Esto volvería más complejo la interacción Humano-Máquina de lo necesario. Por lo tanto se optó por cambiarla por otra opción de comunicación vía UDP (*User Datagram Protocol*) que se detallará más adelante. Especificaciones de la pantalla LCD tomadas directamente del fabricante. [45] La pantalla LCD se muestra en la Figura 32 y sus especificaciones más importantes se muestran en la Tabla 8.

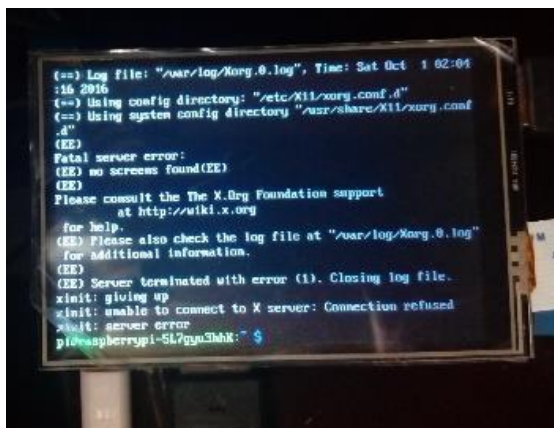


Figura 32 – Pantalla LCD en funcionamiento.

Tabla 8 – Especificaciones técnicas pantalla LCD

Tipo de LCD	TFT
Interfaz de la LCD	SPI
Tipo de Pantalla Táctil	Resistivo
Controlador de la pantalla táctil	XPT2046
Colores	65536
Luz de fondo	LED
Resolución	320*480 (Píxeles)
Relación de aspecto	8:5
Consumo de energía	Por ser definido
Corriente de luz de fondo	Por ser definido
Temperatura de operación (°C)	Por ser definido

Leds de Potencia 3W: Este LED de luz blanca se utiliza para iluminar el cuarto donde se realiza la captura de imágenes. Su alta potencia y un tamaño compacto lo hacen ideal para esta aplicación. Su alta intensidad provee muy buena iluminación. Con tan solo 2 leds se lograron condiciones óptimas de iluminación. La imagen de la Figura 33 es tomada del fabricante. [46]

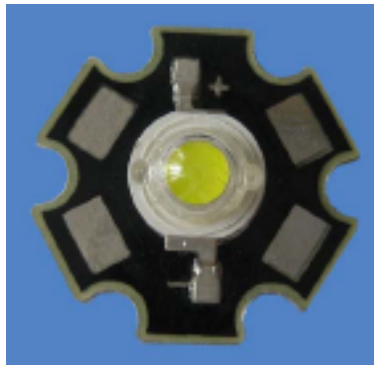


Figura 33 – Led de potencia 3W con su disipador [46]

Accesorios para el RPI: Los accesorios incluyen:

- Un ventilador para ventilar el sistema.
- Disipadores que fueron instalados en el procesador para disipar el calor.
- Una caja de acrílico transparente para proteger el RPI.
- Un cable HDMI para la visualización de imágenes un monitor HD.
- Un cable Ethernet para conexión directa RPI – PC.

Nota sobre el cable Ethernet: Debe ser cruzado para conexiones punto a punto. El RPI y computadores modernos pueden detectar el tipo de cable usado y comunicarse según estén conectados. Sin embargo, es buena práctica usar los cables cruzados para este tipo de aplicaciones, sobre todo en aquellos equipos antiguos que no poseen estas funcionalidades de detección del tipo de cable ya que la comunicación podría fallar.

### 3.3 SOFTWARE UTILIZADO

En esta sección se describe todo el software utilizado para la construcción del proyecto. Las licencias de los programas principales, Matlab/Simulink son proveídas por la Universidad (utilizando los computadores de la Universidad). En el restante de programas comerciales se utilizaron licencias de demostración, como por ejemplo 30 días de prueba.

#### 3.3.1 Matlab y Simulink

Se necesita un programa matemático capaz de computar código que involucra matrices para desarrollar una aplicación en visión artificial. Una opción fue el Octave que está desarrollado para correr nativamente en el RPI, es de código abierto y tiene complementos para visión artificial. Sin embargo, Matlab ha sido estudiado durante gran parte de la carrera de Ing. Mecatrónica en la USBBOG, por tanto se dio preferencia a trabajar sobre el camino ya conocido y recorrido.

En Matlab están creados los algoritmos usados para la visión artificial. Más adelante se entrará a explicar los algoritmos en su respectiva sección. Debida a la alta importancia de estos programas en el desarrollo de este proyecto, se entrará a enfatizar

sobre Matlab y Simulink, sus similitudes, diferencias y propósitos. Esto con el propósito de entender cuáles herramientas de programación se usaron y su motivo.

MATLAB (*Matrix Laboratory*) es un entorno de cómputo numérico multiparadigma y lenguaje de programación de cuarta generación. Un lenguaje de programación patentado desarrollado por MathWorks, MATLAB permite la manipulación de matrices, trazado de funciones y datos, implementación de algoritmos, creación de interfaces de usuario e interconexión con programas escritos en otros idiomas, incluyendo C, C ++, C #, Java, Fortran y Python. Ver Figura 34.

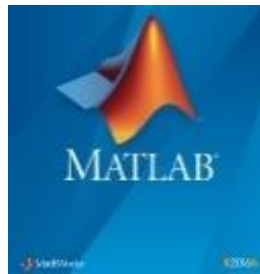


Figura 34 – MATLAB software. [47]

Simulink, desarrollado por MathWorks, es un entorno de programación gráfico para modelar, simular y analizar sistemas dinámicos de multidominio. Su interfaz principal es una herramienta gráfica de diagrama de bloques y un conjunto personalizable de librerías de bloques. Ofrece una estrecha integración con el resto del entorno de MATLAB y puede manejar MATLAB o ser secuenciado a partir de él. Simulink es ampliamente utilizado en control automático y procesamiento de señales digitales para la simulación multidominio y el diseño basado en modelos. Ver Figura 35.



Figura 35 – Simulink [47]

Si bien una gran cantidad de bloques que proporciona hacen la mayoría de las mismas cosas que se hacen en Matlab, hay que comprender que sirven distintos

propósitos. Simulink tiene acceso a MATLAB, pero tiene su propio motor de simulación. No es simplemente una vista gráfica de MATLAB. La base de MATLAB es álgebra lineal y Simulink está hecho para resolver ODEs (*Ordinary Differential Equation*). Eso no significa que no se puedan resolver ODEs en MATLAB o álgebra lineal en Simulink, pero la base difiere entre los dos.

Lo siguiente es un punto clave del desarrollo de la tesis y es que hay que entender que si bien hay varios métodos posibles de exportar código a aplicaciones ejecutables, solamente una de ellas resultó ser apta para el RPI. Toda esta información tomada de la documentación de Matlab y sus foros. [47]

MATLAB Compiler: puede generar ejecutables, DLLs, etc. del código de MATLAB, pero hace uso de las librerías de ejecución de MATLAB. Para esto el usuario debe instalar el MATLAB Compiler Runtime (MCR) en la máquina de implementación para ejecutar los binarios producidos por MATLAB Compiler. Esto también es útil para generar plugins y aplicaciones con C/C++, Java, .NET e inclusive Microsoft Excel. Sin embargo, esto no es soportado en el RPI.

MATLAB Coder: se puede generar código completamente independiente, sin embargo, sólo se admiten un conjunto limitado de funciones para la generación de código. Es útil para ejecutar aplicaciones compiladas de Matlab en equipos que no tienen instalado Matlab. Sin embargo esta opción no es válida porque sólo puede ejecutarse en procesadores x86 y no procesadores ARM como del RPI.

Simulink Embedded Coder: genera código C y C++ legible, compacto y rápido para su uso en procesadores embebidos, tarjetas de prototipado rápido y microprocesadores utilizados en producción masiva. Ésta es la opción que permite exportar código de Simulink a ejecutables directamente al RPI.

Ambos Matlab y Simulink tienen su propio conjunto de funciones para manipular el RPI, sin embargo, dado el propósito de generar código embebido e independiente, solamente se pueden usar aquellos bloques de Simulink. Ver Figura 36.

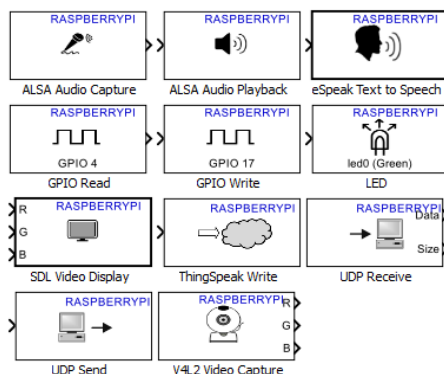


Figura 36 – Librería Simulink para el RPI

Los bloques *GPIO Read* y *GPIO Write* son usados para leer y escribir los pines de entradas y salidas digitales GPIO. Los bloques *SDL Video Display* y *V3L2 Video Capture* son usados para la configuración, uso y visualización de la RaspiCam. Los bloques *ALSA Audio Capture*, *ALSA Audio Playback*, *eSpeak Text to Speech* y *ThingSpeak* no fueron usados. El bloque *LED* controla el único led integrado para programación en el RPI. Por último los bloques *UDP Send* y *UDP Receive* son usados para el protocolo de comunicación vía UDP con otras aplicaciones del mundo exterior.

Adicionalmente, el bloque *Interpreted MATLAB Function* de Simulink permite llamar funciones escritas en Matlab teniendo en cuenta que tiene un número limitado de comandos compatibles. El bloque se muestra en la Figura 37.

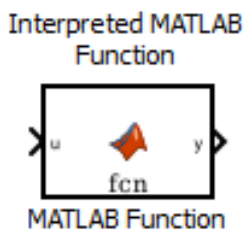


Figura 37 – Función de Matlab Interpretada

### 3.3.2 Rhinoceros

Se necesita un programa de diseño asistido por computadora para realizar el modelado del módulo de visión artificial. Algunos programas como SolidWorks y SolidEdge son candidatos para los diseños enfocados hacia análisis de esfuerzos de la estática y dinámica, y hacer simulaciones. Estos análisis están fuera del alcance y por tanto se prefiere un programa más enfocado al modelado con objetivo de crear piezas para impresión 3D y la estructura mecánica para cortes 2D en material MDF. Una opción para esto con la que el autor ha trabajado previamente es Rhinoceros.

Rhinoceros (Rhino, Rhino3D) es un software comercial de computación 3D y software de diseño asistido por computadora (CAD) desarrollado por Robert McNeel & Associates; Una empresa estadounidense de propiedad privada que fue fundada en 1980. La geometría de Rhinoceros se basa en el modelo matemático de NURBS, que se centra en producir representación matemáticamente precisa de curvas y superficies de forma libre en gráficos de computadora (en oposición a las aplicaciones basadas en mallas de polígonos). En la Figura 38 se muestra en el ensamble en la vista de 4 pantallas.

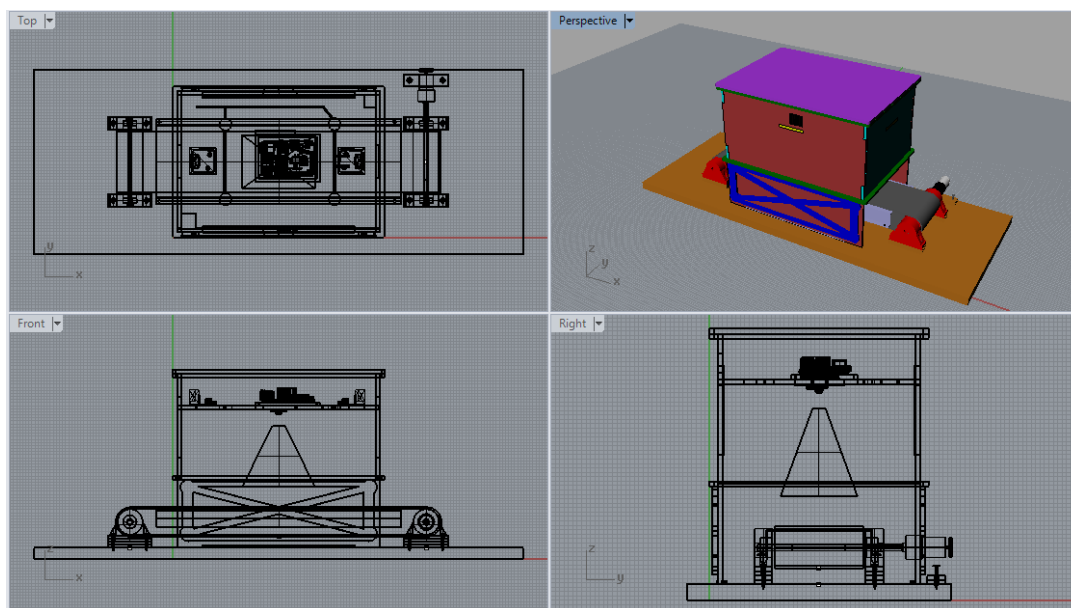


Figura 38 – Diseño mecánico en Rhinoceros

### 3.3.3 PuTTY

El Raspberry Pi es un computador totalmente completo y equipable como uno de escritorio o una laptop. Inclusive soporta el uso de periféricos como teclado, mouse, monitor, etc. Ahora bien, aunque es un computador en sí, no se hace programación directa sobre él porque el ambiente de desarrollo escogido para programación funciona en equipos de escritorio y no tarjetas de prototipado rápido como el RPI.

Según lo anterior surge un problema de comunicación de cómo manipular el RPI mientras se generan códigos en otro equipo independiente. Una opción sería dotar al RPI de periféricos para convertirlo en un computador completo, pero esto sería contradictorio con el propósito de los embebidos ya que se volvería más voluptuoso ocupando más espacio, y estos están diseñados para ejecutar funciones dedicadas, no como un computador de propósito general.

Entonces se opta por usar un programa de conexión segura SSH para iniciar sesión en el RPI y manipularlo desde un computador de escritorio. Para esto se utilizó PuTTY. Con él se lograron tareas como: Configurar el RPI, manipular sus archivos, configurar la LCD, instalar controladores, etc.

PuTTY es un emulador terminal libre, una consola serie y una aplicación de transferencia de archivos de red de código abierto. Soporta varios protocolos de red, incluyendo SCP, SSH, Telnet, rlogin y conexión de socket sin procesar.

### 3.3.4 SocketTest v3.0.0

En este sistema *soft-real time* es necesario supervisar y adquirir datos para gestionar procesos e interactuar con el sistema de visión artificial. De manera que es necesario desarrollar una interface de comunicación humano-máquina HMI para esto.



Un HMI es una Interfaz de usuario en un sistema de control de fabricación o proceso. Proporciona una visualización gráfica, monitoreo y/o interfaz de interacción con los sistemas o controles.

Desde PuTTY se puede establecer comunicación con el RPI para manipular su configuración y hacer transmisión de archivos, pero no puede comunicarse directamente con los datos del programa en ejecución. Para esto se desarrolló un pequeño protocolo UDP para comunicarse entre el programa del RPI y el PC. Esto implica una comunicación básica cliente-servidor. El RPI viene a ser el servidor en escucha de mensajes entrantes para responder información. De manera que se necesita un cliente para hacer las peticiones y enviar mensajes.

SocketTest v3.0.0 es una herramienta hecha en Java para hacer pruebas de sockets. Puede crear cliente o servidor TCP y UDP. Se puede utilizar para probar cualquier servidor o cliente que utilice el protocolo TCP o UDP para comunicarse. Pruebas de comunicación mostradas en la Figura 39.

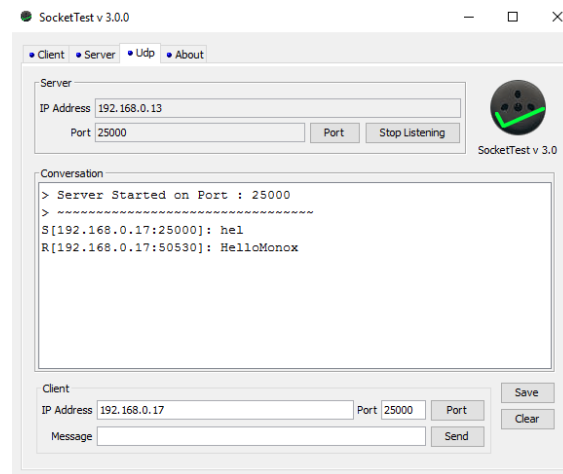


Figura 39 – Pruebas comunicación UDP entre PC y RPI

Este programa es útil para depurar la comunicación. En él no se puede programar ni desarrollar ningún complemento, pero sirve para los pasos iniciales de la comunicación. Una vez probado y creado el protocolo de comunicación se puede

reemplazar este programa por otro ambiente de desarrollo donde entonces ya pueda crearse una interfaz gráfica y ejecutar código. Este reemplazo es lo que nos lleva inmediatamente a la siguiente sección.

### 3.3.5 LiveCode

Según lo mencionado en la anterior sección se requiere un ambiente de desarrollo para crear una interfaz gráfica que permita la comunicación Humano-Máquina, es decir un HMI. Para esto se necesita un ambiente de desarrollo rápido que permita crear un cliente que funcione como HMI. Opcionalmente también se podría considerar la opción de exportar este cliente a un Smartphone vía aplicación Android. Es aquí donde entra LiveCode.

LiveCode es un entorno de desarrollo multiplataforma que permite crear aplicaciones que se ejecutan en todos los dispositivos y plataformas principales: iOS y Android, así como Mac, Windows y Linux, los entornos Web y de servidor.

Con su lenguaje de programación con sintaxis basada en el idioma Inglés y un entorno de desarrollo visual se puede crear un cliente HMI con comunicación vía UDP de manera rápida.

Algunas funciones desarrolladas son el envío de imágenes para la captura de fotos de las clases a entrenar y el ajuste de ciertos parámetros como el área disponible para el procesamiento digital de imagen.

### 3.3.6 Fritzing

Se necesita desarrollar un esquemático electrónico para realizar las conexiones de los elementos Hardware como el RPI, el motor, el ventilador, los LEDs, las tarjetas controladoras de motores, y por supuesto la RaspiCam.

Debido a la sencillez de los elementos y al hecho de que todas las tarjetas ya vienen listas para su uso, evitando la necesidad del uso de componentes básicos de electrónica y el diseño de circuitos, se optó por usar un programa amigable que permitiera hacer estas conexiones de una manera rápida. Un programa con modelos gráficos de los componentes para una mejor visualización.

Fritzing es una gran herramienta de código abierto para generar prototipos de proyectos electrónicos. Permite diseñar esquemas y PCBs para ser fabricados en masa. Otro programa considerado y trabajado por el autor fue EAGLE pero es más complejo y está enfocado exclusivamente a hacer PCBs. El alcance de este proyecto va hasta la implementación del esquemático sin la necesidad de hacer un PCB. Por lo tanto se prefirió Fritzing por su facilidad de uso y amigabilidad con el usuario. Se muestra el esquemático en proceso de diseño en la Figura 40.

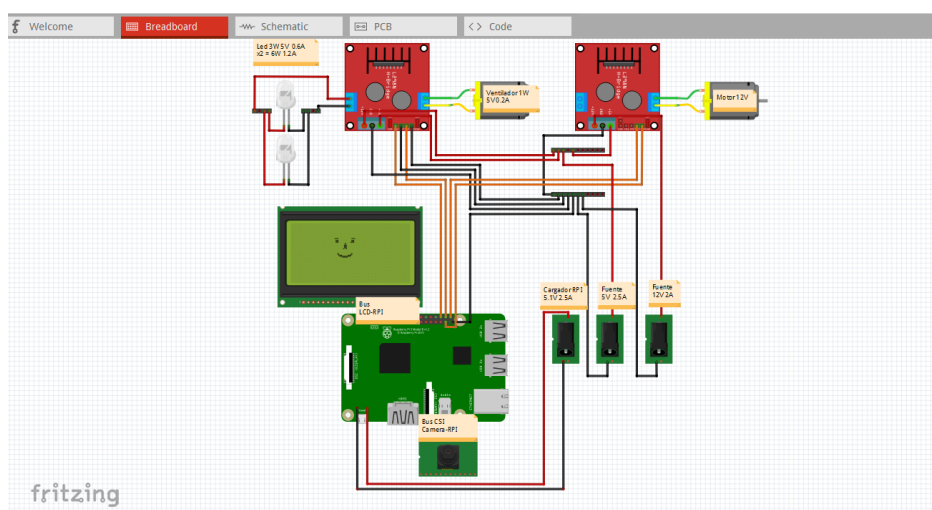


Figura 40 – Programa Fritzing

### 3.3.7 Otros

En los programas comerciales se utilizaron licencias de demostración, como por ejemplo 30 días de prueba. Otros programas de menor relevancia, puesto que no tienen un impacto directo sobre el funcionamiento del proyecto incluyen:

DiskImager: Para el formateo y acondicionamiento de la memoria SD donde se carga el sistema operativo del Raspberry Pi.

Android Studio SDK y Java SDK: usados como extensiones del LiveCode para desplegar aplicaciones LiveCode en Android.

Photoshop y AutoCad: Con este se diseñó el logo de la tapa para luego ser exportado en formato vectorial a la máquina.

WinSCP: Para facilitar la transferencia de archivos entre el RPI – PC, que es diferente a la comunicación Humano-Máquina.

Wireless Network Watcher: Programa que detecta todas las IPs de los dispositivos conectados a la red. Útil para encontrar equipos y al RPI cuando se migra a una nueva red y se desea establecer una comunicación de red local.

Htop: Es el único programa nativo del RPI usado. Útil para monitorear el rendimiento de los 4 núcleos del procesador y maximizar su uso aumentando o disminuyendo las ejecuciones por segundo de los procesos diseñados.

## 3.4 CONSTRUCCIÓN

### 3.4.1 Modelado 3D

Comprender e interpretar planos 2D mientras se estima su profundidad puede ser difícil. Es posible obtener la idea general de un plano, pero puede ser complicado visualizar el diseño final. Si bien es cierto que las operaciones mecánicas como cortes, doblados, taladrado y demás solamente necesitan planos 2D, es mucho mejor

experimentar el diseño en 3D para entender cómo va a funcionar. Cuanto mejor sea la visualización del diseño, más fácil será su comprensión para tomar decisiones y hacer modificaciones.

A partir de un boceto se modela la máquina en 3D. Con este modelado se puede realizar Impresión 3D (termo formado de Nylon) o la realización de cortes láser a partir de planos 2D exportados. El uso de maquinaria ofrece muy buena precisión con gran detalle en los acabados a un relativo bajo costo.

El ensamble final se encuentra en la Figura 41. Una vez terminado el ensamble, revisado y modificado múltiples veces se procedió a construir las piezas por medio de distintas técnicas.

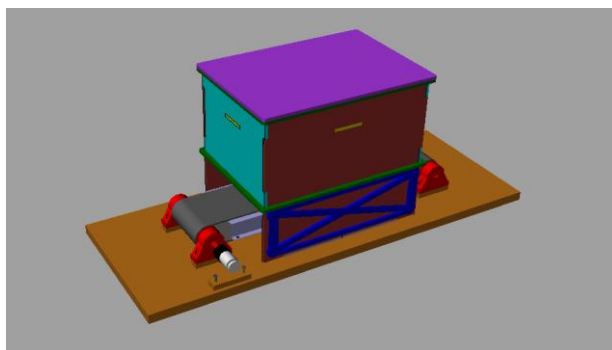


Figura 41 – Ensamble 3D en Rhinoceros

### 3.4.2 Proceso de manufactura

En esta sección se procederá a describir cómo se llevó a cabo la construcción de las piezas digitales. Se utilizaron las siguientes técnicas:

Cortes Láser: Toda la estructura de la caja está hecha en material MDF 5mm. Las piezas se exportaron a archivos de planos y se hicieron los cortes en máquina cortadora láser. Inicialmente el acabado de las piezas es regular puesto que las piezas salen con

quemaduras. Sin embargo, esto se soluciona fácilmente lijando el MDF. Otra opción de material era el acrílico. Éste viene con un plástico cobertor que lo protege de las quemaduras. Se escogió el MDF porque se considera más manipulable, es decir, la madera al ser flexible cede un poco ante los esfuerzos en cambio del acrílico que al ser frágil se rompe con facilidad al exceder un esfuerzo. En la Figura 42 se encuentra la máquina ejecutando los cortes.

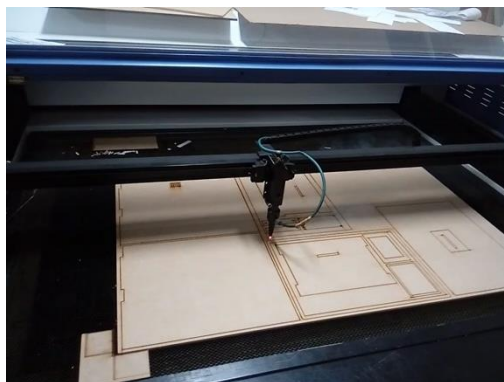


Figura 42 – Cortes láser en proceso

Cortes con sierra: Los soportes de aluminio que sostienen la caja fueron cortados con una sierra de diente afilado. El tablón que soporta todo el proyecto es un MDF de 1.8cm cortado con maquinaria industrial.

Impresión 3D: Las piezas de Rhino fueron exportadas al formato STL (STereoLithography) para impresión 3D. La impresora Ultimaker recibe la información vía el programa Cura. Se decidió imprimir algunas piezas por la gran facilidad que ofrece esta tecnología para crear piezas rápidas con una geometría bastante compleja. Para armar piezas conjuntas 3D a partir de piezas planas 2D es un poco difícil, entonces se optó imprimir las piezas para facilitar los ensambles complejos.

Éste fue uno de los pasos más complicados en la manufactura ya que la impresora 3D se descalibra con facilidad y es bastante exhaustivo recalibrarla. Además, el material utilizado, Nylon para guadaña de color rojo, es un material barato pero difícil de manipular. Esta impresora puede trabajar con cualquier Nylon ya que son termoformables. Sin embargo, este Nylon específico tiene un diámetro irregular a lo largo

de su recorrido. Esta variación de diámetro, junto con sus impurezas dificultó bastante la impresión. Después de incontables horas de lidiar con la impresora se logró calibrar e imprimir las piezas deseadas.

Torneado, Doblado y Taladrado: Las piezas de metal como el soporte de aluminio de la banda transportadora, el soporte del motor y la manzana de acople con el eje se hicieron en un taller mecánico con un técnico especializado. La pieza más complicada fue la manzana de acople ya que el eje del motor y el eje de la banda transportadora eran de distintos diámetros. Para acoplarlos se necesitaba una pieza que garantizara una concentricidad perfecta y para esto un buen torno y un buen mecánico.

Ensamble: Finalmente, con todas las piezas construidas y disponibles se procedió a ensamblar todo. De manera que toda la parte mecánica quedó lista. En la siguiente sección se procede con el diseño electrónico. Se muestra el ensamblado final en la Figura 43



Figura 43 – Ensamblado Final

### 3.4.3 Conexiones electrónicas

Las complejidades del diseño electrónico son uno de los pasos más difíciles en el diseño de proyectos mecatrónicos. Sin embargo, la electrónica es cada vez más accesible. Esto se debe al auge de las tarjetas para la creación de prototipos rápidos en la industria electrónica. Algunas de ellas como la Raspberry Pi han dado lugar a nuevas puertas abiertas para los ingenieros que desean una manera rápida y eficaz de hacer prototipos o de construir sistemas de bajo volumen.

Todos los componentes principales usados en este proyecto vienen listos de fábrica para conectar y programar. Gracias a la avanzada electrónica del Raspberry Pi que es un SoC (System on Chip) se pueden hacer diseños electrónicos bastante sencillos. Este proyecto utiliza un RPI, un motor, un ventilador, dos LEDs de potencia, una cámara, fuentes de energía y dos tarjetas controladoras de motores.

El motor, los dos leds y el ventilador, todos funcionan como cargas en el sistema y por tanto los tres pueden ser manejados independientemente con la tarjeta controladora de motores. Los led son una carga resistiva (R) mientras que el ventilador y el motor es una carga resistiva inductiva (RL). La cámara, y en su momento la LCD, tienen conexiones directas a los pines RPI. Solamente hay que descargar controladores y configurarlos. En Figura 44 está el esquemático y la Figura 45 muestra las conexiones reales.



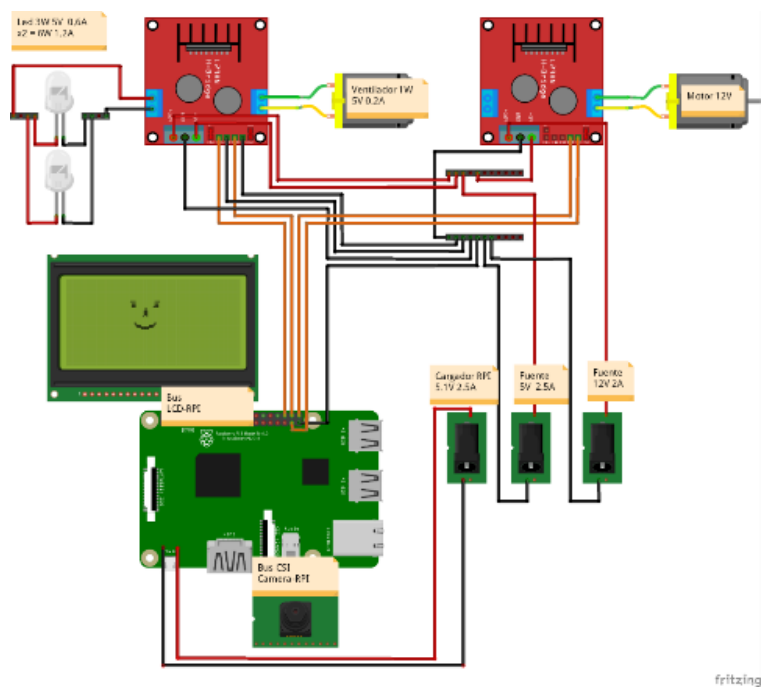


Figura 44 – Esquema electrónico

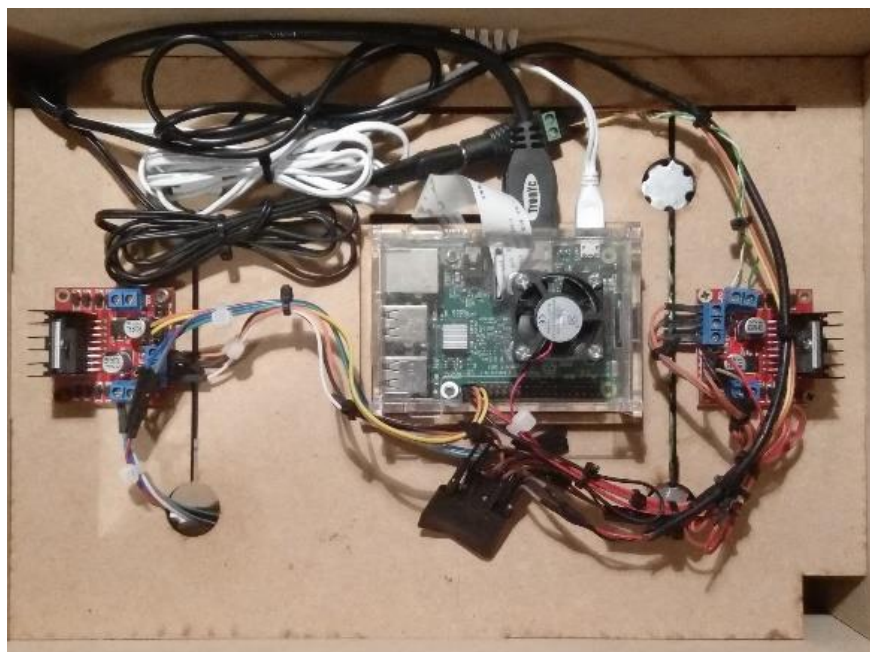


Figura 45 – Conexiones

### 3.5 PROGRAMACIÓN

La técnica de clasificación utilizada es la del Clasificador Bayesiano Ingenuo el cual es explicado en el ítem 2.5.3. El modelo práctico de este clasificador varía un poco con respecto al modelo teórico que se irá explicando a lo largo de esta sección.

#### 3.5.1 Primeros pasos

La parte más difícil de todo el trabajo son los primeros pasos de aprender a manipular el RPI y programarlo desde Simulink. Para lograr esto se requieren conocimientos mínimos de Linux y un muy buen manejo del dúo Simulink/Matlab.

En esta importante parte de la programación, se mencionan una serie de pequeños pasos hasta llegar al punto final de ser capaz de programar el RPI desde Simulink e implementar los algoritmos propuestos para el reconocimiento de objetos. Los pasos son:

- Instalar el sistema operativo del RPI y aprender un poco de Linux.
- Configurar la IP del RPI para conexión punto a punto.
- Configurar la red Wifi del RPI.
- Aprender a iniciar sesión vía SSH o vía VNC si se prefiere un modo gráfico.
- Configurar la resolución de pantalla y manipular los archivos de configuración del RPI.
- Instalar y configurar la LCD.
- Instalar todos los paquetes de soporte de RPI para Matlab y Simulink.
- Aprender a configurar la RaspiCam desde Matlab y luego desde Simulink.
- Comunicar el RPI con el mundo exterior vía protocolo UDP.

Una vez realizados estos pasos se puede entonces entrar a programar el RPI. La mayor preocupación durante la fase de programación del RPI es la velocidad de procesamiento. Cuántas veces por segundo se puede ejecutar el código. Todo depende de la cantidad de código que deba procesar. El RPI con su procesador de 800Mhz y 4 núcleos ofrece recursos apenas suficientes para imágenes de muy baja resolución.

Es importante tener en cuenta que cuanto mayor sea la resolución de la imagen mucho mejor será el reconocimiento puesto que se cuenta con mayor información del objeto entre más grande sea la resolución. Por el contrario, tener una alta resolución implica una mayor carga de procesamiento al RPI lo que lo vuelve más lento. En la Tabla 9 se muestran algunos ejemplos empíricos de los resultados de distintos algoritmos comparados contra la resolución de imagen. La velocidad resultante se le denomina FPS (Franjas Por Segundo).

Tabla 9 – Velocidad de ejecución versus FPS

<b>FPS</b>	160x120	320x240	640x480	800x600
Edge detection (Sobel operator)	72.4	18.2	4.5	2.9
Corner detection (Harris&Stephens)	22.6	4.7	1.2	0.7
Line detection (Hough transform)	19.6	5.7	1.6	1.0

Como se estima manejar un muy alto nivel de procesamiento y se desea tener una alta velocidad de respuesta, es decir, una ejecución rápida del código entonces, bajo pruebas empíricas se escoge una resolución personalizada de 200x200 pixeles. Ésta es lo suficientemente pequeña para mantener rápido el procesamiento y lo suficientemente grande para adquirir buena información sobre la imagen.

### 3.5.2 Localización de objetos y selección de características

Por objeto se entiende una región contigua de píxeles blancos (binario 1 o entero 255) con un área mínima de 100 px. Aquellas regiones menores a este valor se consideran ruido de la imagen. No hay una limitante para el área máxima. En una misma imagen puede haber varias regiones separadas lo que indica que hay varios objetos. Es necesario ejecutar el Algoritmo de procesamiento de imagen en el ítem 3.5.3 para obtener estas imágenes conectadas. Sobre estas imágenes conectadas se aplica la extracción de características.

A continuación se procede a hacer una selección de las características más viables y descartar aquellas que no son útiles. En el ítem 2.3.2 se explican las características. Todas tienen propósitos particulares y diferentes usos. Se debe limitar el número de características utilizadas para reducir la carga de procesamiento en el RPI. La Figura 46 muestra aquellas características disponibles en el bloque *Blob Analysis* que son evaluadas para determinar su relevancia en el reconocimiento.

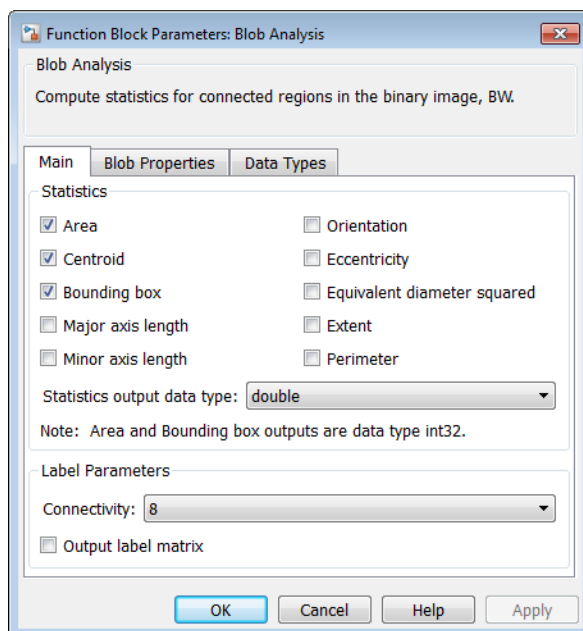


Figura 46 – Características disponibles en el bloque *Blob Analysis* [48]

Las características evaluadas son:

- Área, Perímetro, Eje Menor y Eje Mayor por ser características morfológicas poco dependientes de la posición u orientación del objeto. Existe una mínima variabilidad porque la RaspiCam puede percibir una geometría distinta del objeto según su ubicación, pero es un error tolerable.
- El Centroide (*centroid*) para el control de posición. La variable a controlar es la coordenada  $x$  del centroide.
- El Cuadro Envolvente (*bounding box*) para fines gráficos de localización de objeto. Esto para indicarle al usuario en pantalla cuál es el objeto que está reconociendo.

Algunas características se descartan porque no son útiles para discriminar los objetos ya que no son pertenecientes a su morfología. Otras se descartan porque dependen de características primarias, en general se prefiere tener características en crudo, con la menor cantidad de operaciones matemáticas para mantener la incertidumbre al mínimo. Otras se descartan porque tienen una alta variabilidad, es decir poca precisión, haciendo que sean poco discriminantes (por precisión se entiende que los datos medidos sean muy cercanos entre sí).

Las características descartadas son:

- Orientación (*orientation*): No es útil porque todos los objetos pueden estar orientados en cualquier dirección generando una baja precisión.
- Excentricidad (*eccentricity*): Ésta es una característica útil pero ya se está usando el eje mayor.
- Diámetro Equivalente al Cuadrado: (*equivalent diameter squared*): Se descarta porque ya se están usando los ejes menores y mayores.
- Extensión (*extent*): No sirve porque al rotar un objeto, varía su cuadro envolvente mientras que el área sí se mantiene constante y esto hace que el resultado de la división tenga una bajísima precisión. En general se buscan características constantes y precisas.

### 3.5.3 Algoritmo de procesamiento de imagen

El propósito de este algoritmo es obtener las características de interés de cada objeto en una imagen. Las características fueron escogidas en el ítem 3.5.2.

El proceso es totalmente secuencial y consta de: adquirir la imagen de la fotografía o la cámara, recortar el área de interés, rotar la imagen si aplica, convertirla de color a escala de grises, umbralizarla con un valor dependiente a la iluminación promedio, erosionarla para borrar píxeles desconectados, dilatarla para recuperar la forma original, obtener las imágenes conectadas y extraer las características. Todos estos procesos son explicados en los ítems 2.2 y 2.3.

Este algoritmo se aplica únicamente al espacio de trabajo que se define como el área entre las líneas rojas de la Figura 47. Si el espacio se reduce, mejora la velocidad de procesamiento puesto que al reducir la cantidad de píxeles a procesar, se reduce considerablemente la carga del procesador.

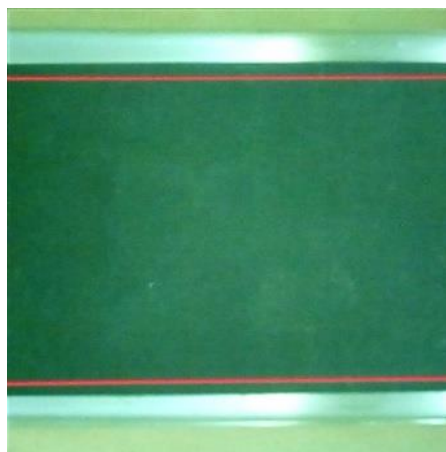


Figura 47 – Espacio de trabajo

El proceso ejecutándose en Matlab se muestra en la Figura 48 y el diagrama de flujo del algoritmo se muestra en la Figura 49.

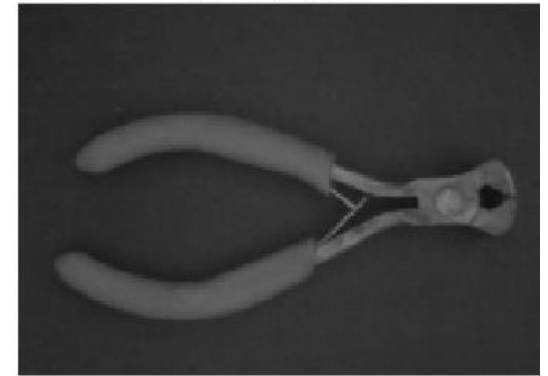
Original



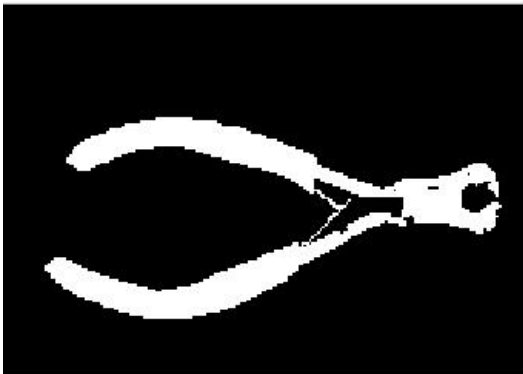
Recortada



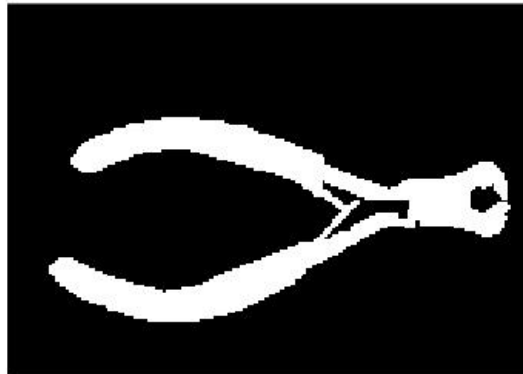
Escala de grises



Binarizada



Dilatada



Erosionada

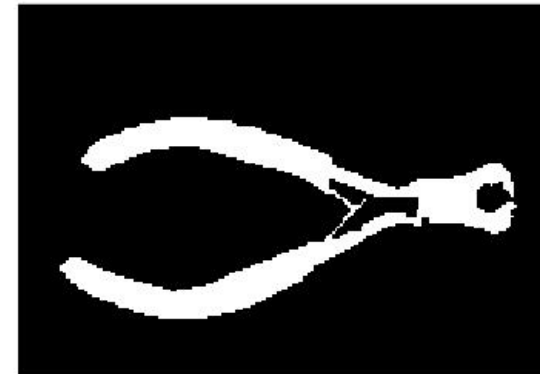


Figura 48 – Procesamiento de imagen en Matlab

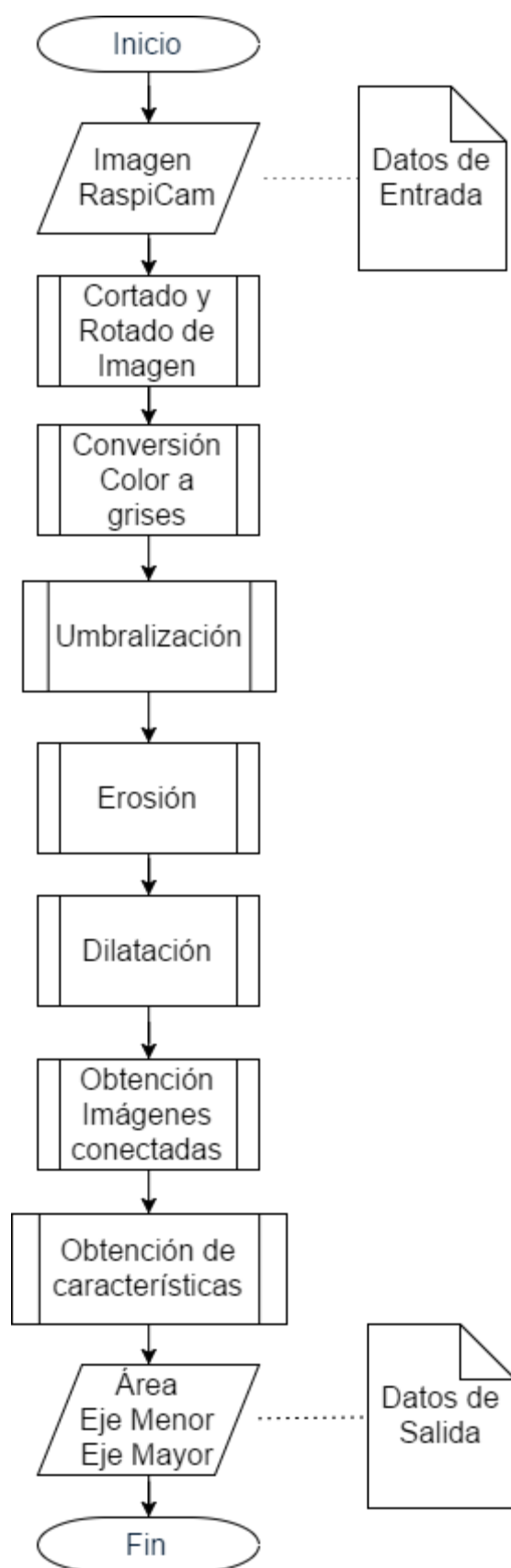


Figura 49 – Algoritmo de procesamiento de imagen



### 3.5.4 Algoritmo de toma de muestras

El proceso de entrenamiento consiste en ejecutar el algoritmo de esta sección para generar una matriz final con las clases resultantes y sus características (datos crudos no procesados aún). Por cada clase que se desee crear, por ejemplo 5 clases para las 5 herramientas: Destornillador, Alicates, Regla, Cinta y Llave Inglesa, se deben tomar alrededor de 100 fotos. La estructura resultante consta del número de clases, donde cada clase tiene un valor por cada característica por cada foto. Por ejemplo, si se tiene 1 clase con 100 fotos, y se desean 2 características, entonces se tiene una matriz de dimensiones  $1 \times 100 \times 2$  para un total de 200 valores. El análisis estadístico de la matriz resultante se realiza en otro algoritmo explicado en el ítem 3.5.5

El algoritmo recorre todas las fotos, en todas las carpetas de entrenamiento mientras les extrae las características de interés. En la Figura 50 se muestra la jerarquía de cómo se distribuyeron las fotos de cada clase para realizar la clasificación supervisada.

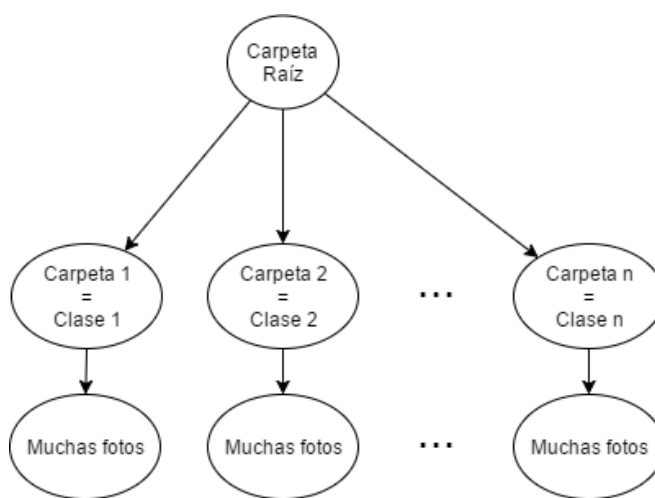


Figura 50 – Jerarquía de clases y fotos de entrenamiento

Para obtener las características este algoritmo hace un llamado del algoritmo visto en 3.5.3. En la Figura 51 se muestra el diagrama de flujo del algoritmo entrenamiento y construcción de clases.

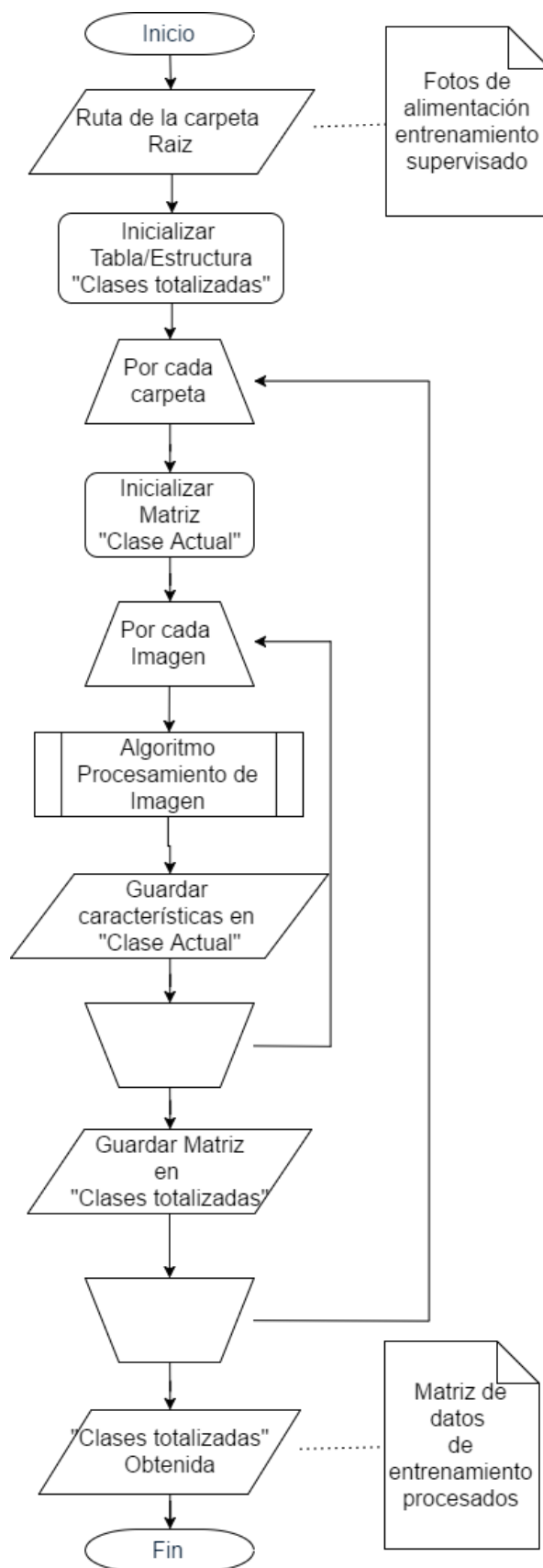


Figura 51 – Algoritmo de entrenamiento y construcción de clases

### 3.5.5 Algoritmo constructor de las clases

El propósito de este algoritmo es generar una matriz de clases entrenadas para luego ser usado en el RPI como base de comparación para clasificar nuevas herramientas.

Primero, este algoritmo toma como entrada la matriz de datos generada por el algoritmo del ítem 3.5.4. Estos son datos crudos sin procesar que equivalen a las características tomadas por cada foto por cada clase. Segundo, este algoritmo los procesa haciendo un llamado del algoritmo del ítem 3.5.6 que retorna la media y la desviación estándar, valor máximo y valor mínimo por cada clase. Finalmente, este algoritmo agrupa los valores estadísticos en la matriz de clases entrenadas. Esta matriz se usa como base de datos que alimenta el clasificador. Por ende, es una clasificación supervisada.

En la Figura 52 se muestra el algoritmo constructor de la matriz del clasificador. Los resultados de este algoritmo son las matrices que se visualizan en el ítem 4.3. Cada clase corresponde al nombre de la carpeta, entrenada con las fotos que contiene y los resultados individuales son la distribución estándar y la media junto con los rangos de máximos y mínimos para cada característica.

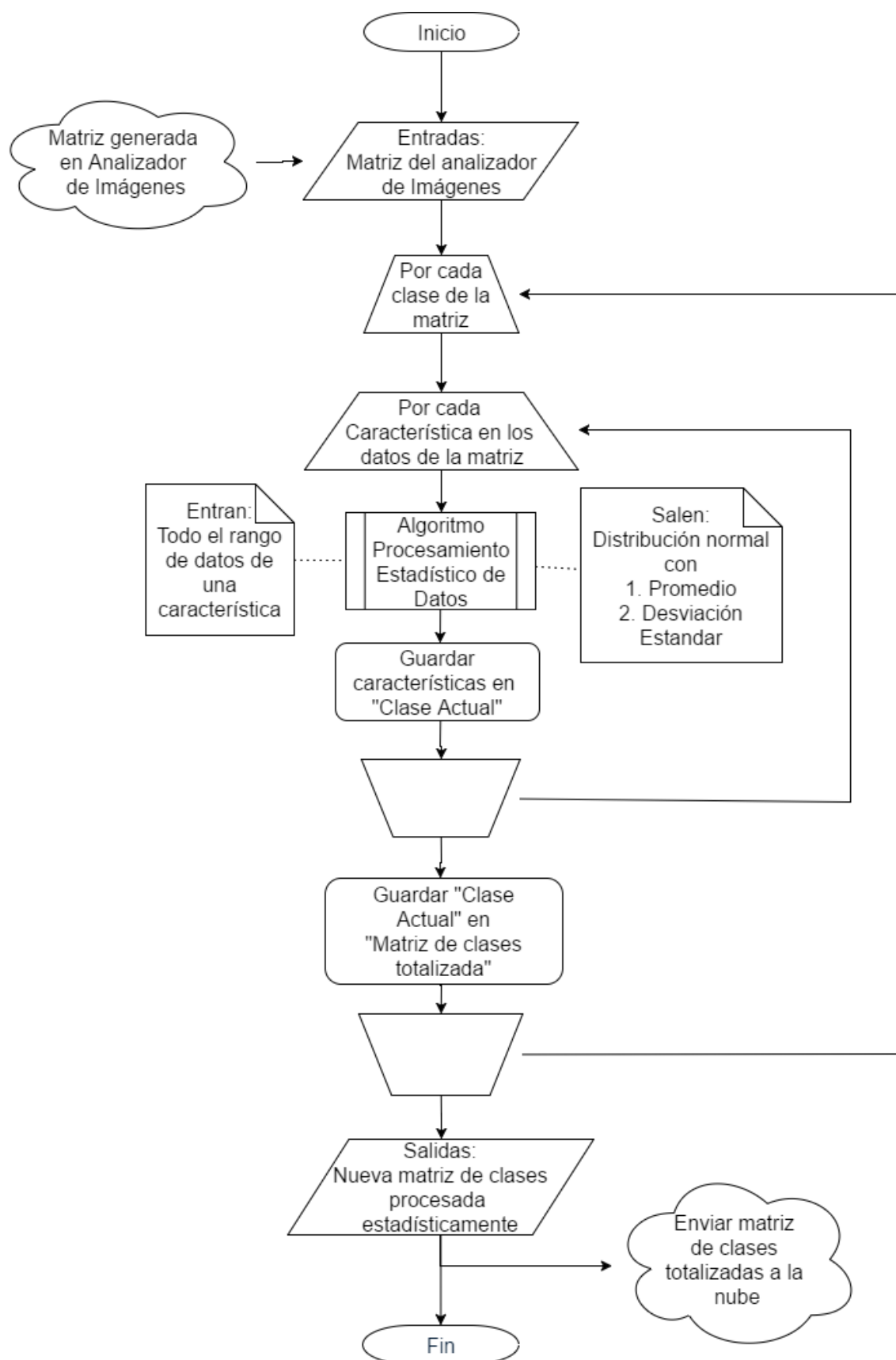


Figura 52 – Algoritmo constructor de la matriz del clasificador

### 3.5.6 Algoritmo modelador estadístico de datos

El propósito de este algoritmo es tomar una muestras de datos, hacerles un ajuste para modelarlas como una distribución normal, y finalmente obtener variables en términos de media y distribución estándar. Estas dos variables son usadas para crear la matriz del clasificador.

Se escoge la distribución normal porque es ideal para modelar características morfológicas. También puede usarse la distribución binomial al ser valores discretos. En la Figura 53 se muestra el algoritmo modelador estadístico de datos.

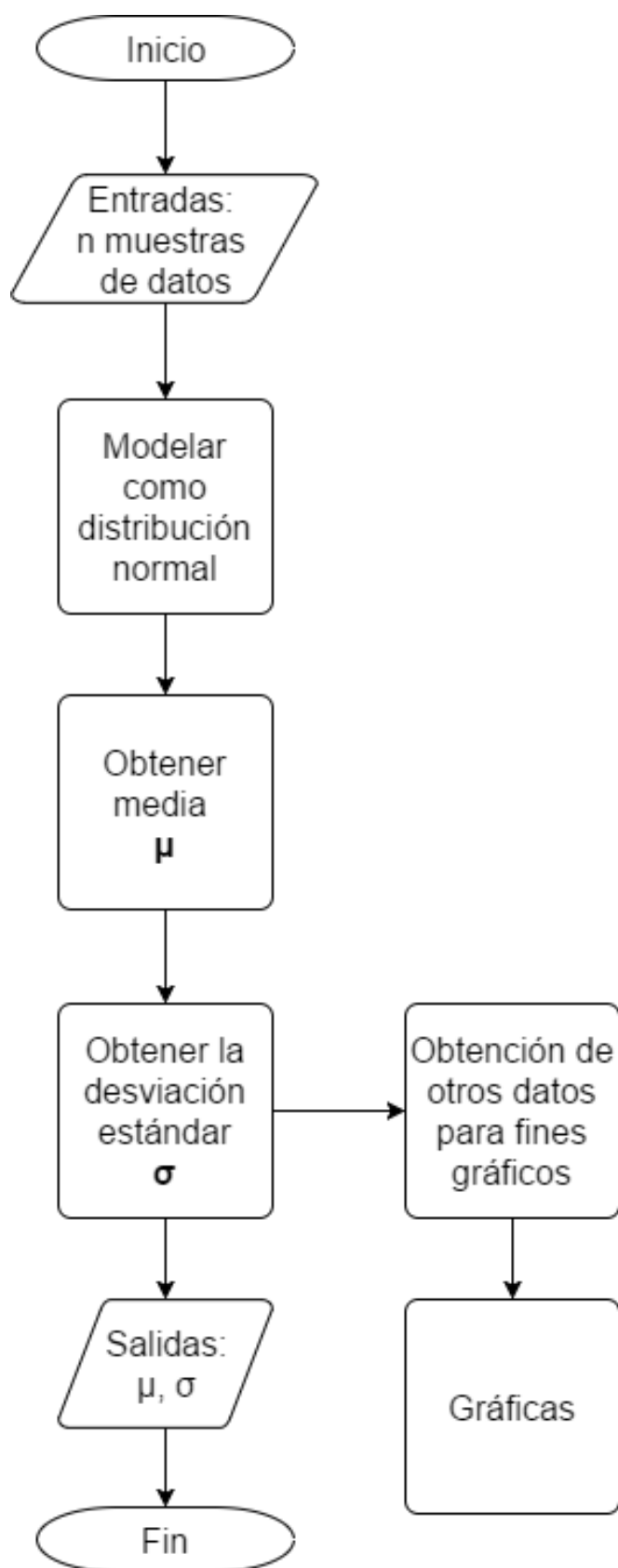


Figura 53 – Procesamiento estadístico de datos

### 3.5.7 Algoritmo clasificador

Éste es el algoritmo más complejo de todos y es dónde reside el propósito de este trabajo. En él se encuentra la implementación del Clasificador Bayesiano Ingenuo. Su propósito es determinar a qué clase pertenecen los datos que se le envía. Durante el funcionamiento continuo del RPI, se colocan las clases (las herramientas) en la banda transportadora. Cuando la herramienta entra en el área de trabajo de la cámara, este algoritmo determinará a cuál clase pertenece la herramienta.

El clasificador Bayesiano maneja la probabilidad anterior (*a priori*) y la posterior (*a posteriori*) como es mencionado en 2.5.3. En la implementación de este algoritmo la probabilidad anterior equivale a la probabilidad de pertenecer a un grupo de media y desviación estándar y la probabilidad posterior es darle mayor probabilidad a aquellas clases que tengan la menor desviación estándar.

Para ello, este algoritmo recibe únicamente valores escalares de las características, toma estos valores y los empieza a evaluar uno a uno en todas las clases dándole un porcentaje de probabilidad (la probabilidad anterior). De esta manera se logra que todas las clases obtengan una probabilidad específica, y al final, el algoritmo promedia las probabilidades y escoge aquella clase resultante con la mayor probabilidad.

Puede suceder un empate en algunas ocasiones, es decir, las dos o más clases probables tienen el mismo porcentaje. Esto ocurre cuando hay valores de características que tienen un 100% de probabilidad de pertenecer a dos o más clases. En caso de empate, entonces el algoritmo escoge aquella clase que tenga la menor distancia vectorial entre las características obtenidas y las características ideales, en otras palabras, se le da prioridad a aquella clase que tenga la menor desviación estándar (siendo ésta la probabilidad posterior). A la clase escogida se le agrega un 5% pasando así a un máximo de 105%. Se podría re ajustar los porcentajes de 0-100% pero se prefiere dejar el 105% para que el usuario conozca que ha ocurrido un empate. Ver Figura 54 y Figura 55.

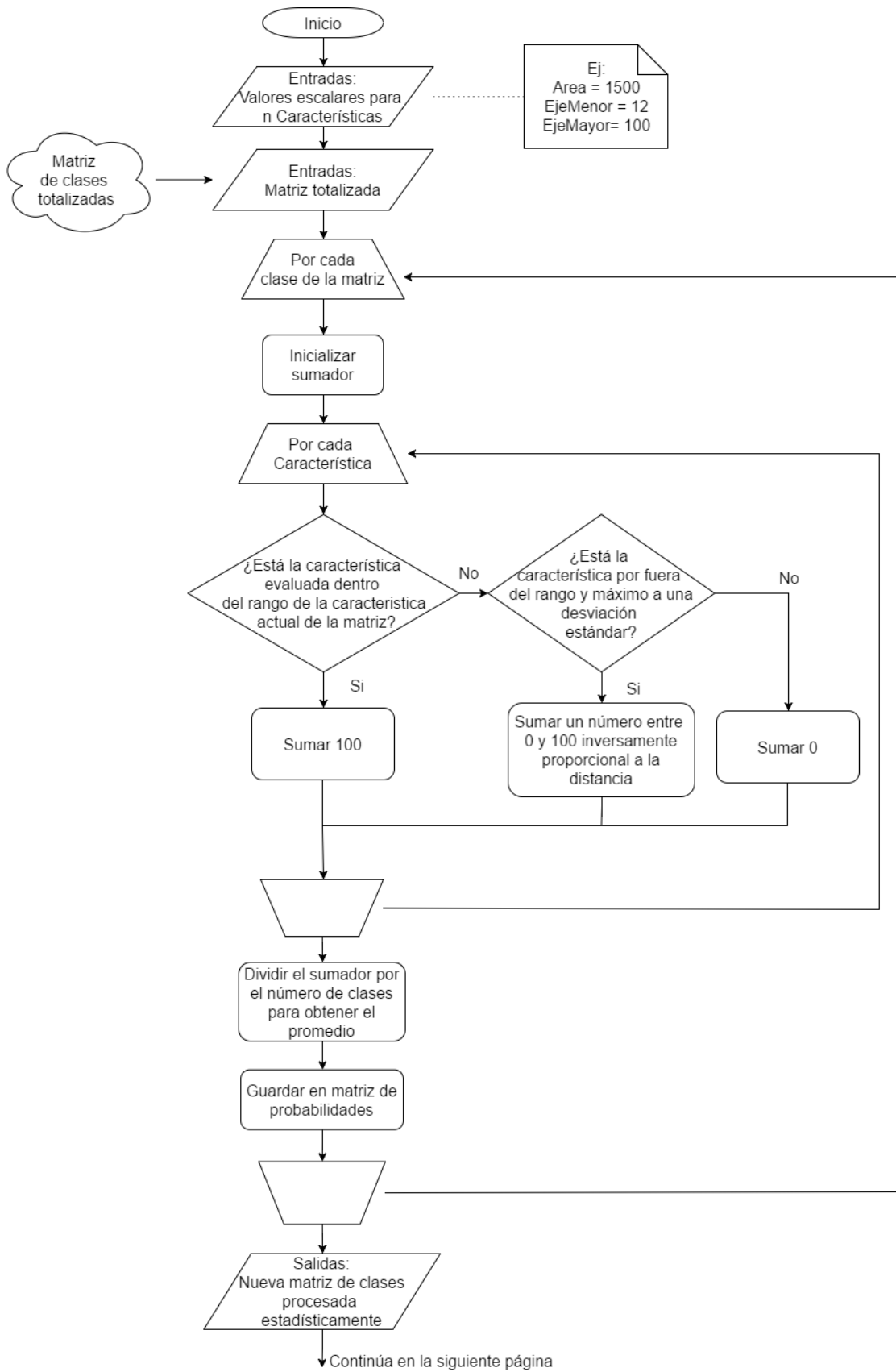


Figura 54 – Parte uno del algoritmo clasificador



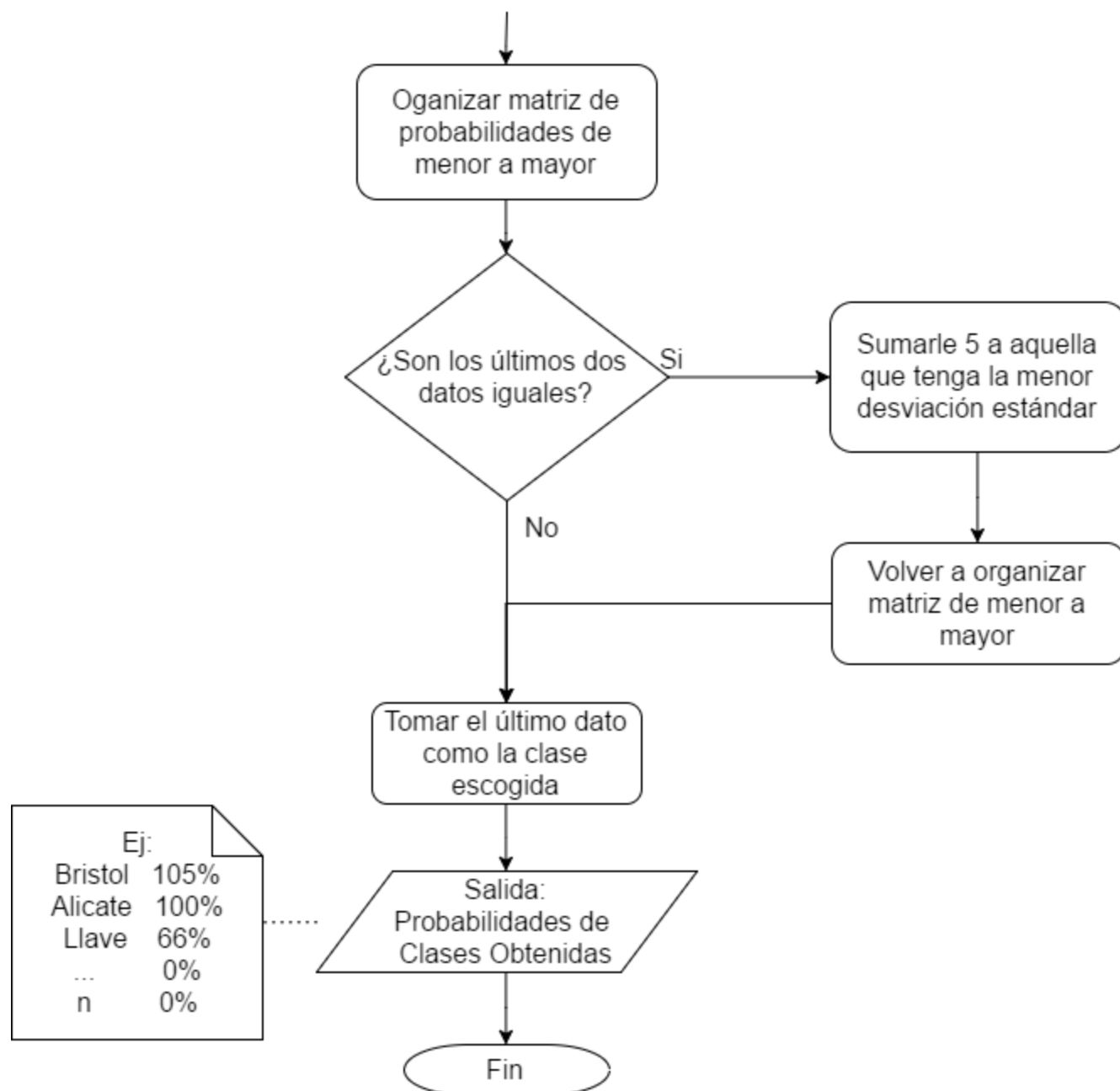


Figura 55 – Parte dos del algoritmo clasificador

### 3.5.8 Algoritmo estimador de la precisión

Una vez creado el clasificador lo inmediatamente siguiente es determinar cuál es su porcentaje de precisión. Para esto, se someten a prueba todos los datos de entrenamiento al clasificador y se analiza el porcentaje de aciertos versus el total de datos suministrados.

El algoritmo desarrollado involucra dos ciclos donde se recorre la matriz que contiene todos los datos de características de las fotos tomadas y las procesa en el algoritmo de Clasificador. La clase resultante del clasificador es comparada contra la clase real, si son iguales entonces se considera un acierto. Si son distintos, entonces falló en clasificar correctamente ese dato. Al final se evalúa el total de aciertos con respecto al total de muestras. Ver Figura 56.

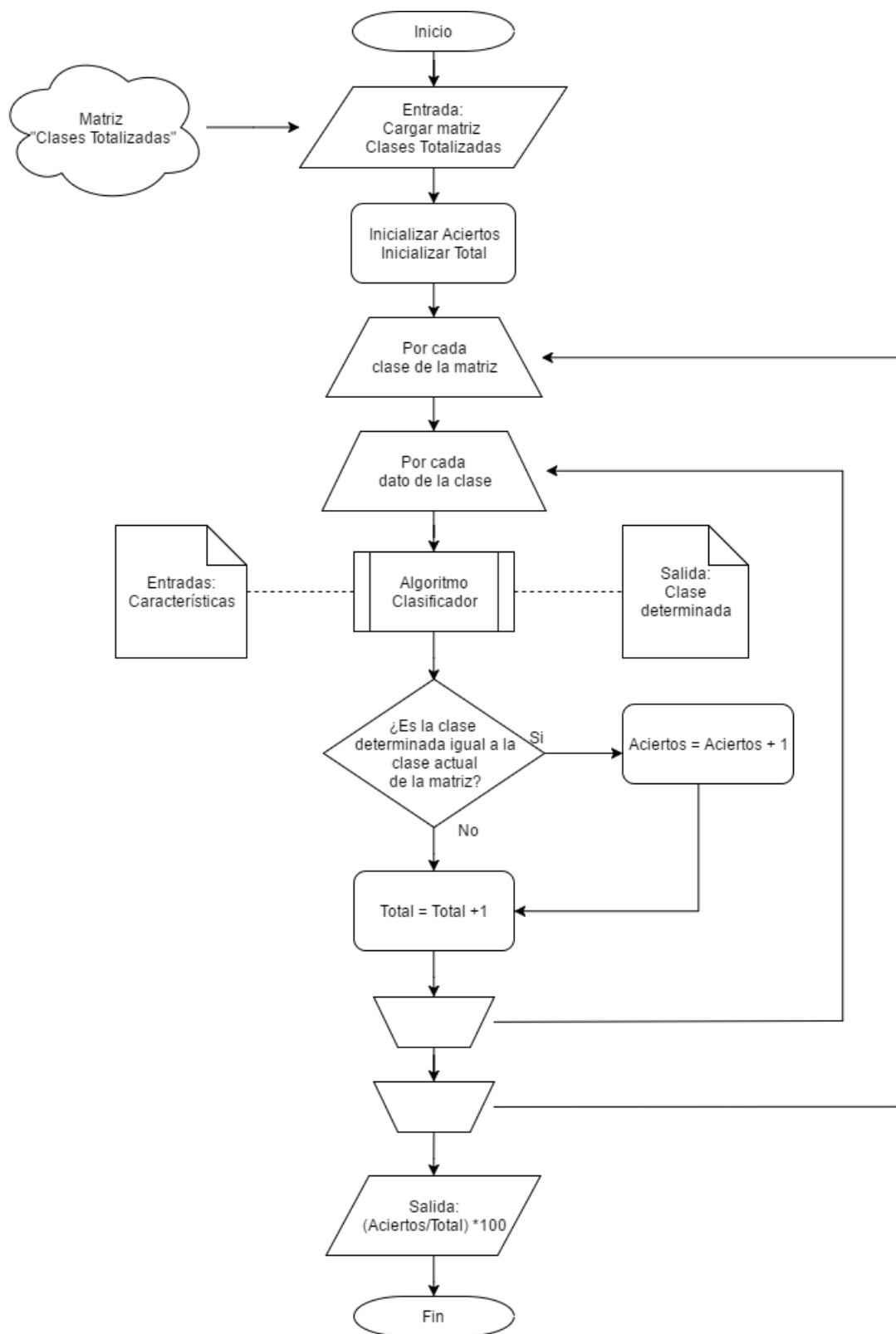


Figura 56 – Algoritmo para calcula la precisión del clasificador

### 3.5.9 Control de posición de la banda transportadora.

La banda transportadora es un sistema estable en todos los puntos de su trayectoria. Aunque se retire la acción de control, en cualquier posición la planta permanecerá en equilibrio. Esto facilita bastante el diseño del control debido a que es una planta naturalmente estable. Por lo tanto, el control se diseña como una máquina de estados que depende de la señal del error, es decir, maneja unas velocidades pre definidas junto con una banda de histéresis. Se controla la posición del objeto cerrando el lazo con la RaspiCam y variando la velocidad del motor en lazo abierto. Los estados del controlador en su respectivo orden son:

- Buscando objeto
- Centrando objeto
- Objeto centrado
- Expulsión de objeto

Las velocidad nominal de la banda (0.15 m/s) se divide en 4 percentiles, de 0-25% no se supera el umbral de activación y se considera detenido, del 26-50% se activa con velocidad Lenta, del 51% al 75% velocidad media y del 76% se considera velocidad rápida.

Los estados del actuador son las acciones de control resultantes:

- Dirección A – Rápido
- Dirección A – Lento
- Dirección B – Rápido
- Dirección B – Lento
- Bloqueado o detenido.

Se utiliza la velocidad rápida cuando no se ha detectado ningún objeto en el área de la RaspiCam y posteriormente se pasa a la velocidad lenta para centrar un objeto después de haber sido detectado.

La variable de retroalimentación de la posición es la coordenada  $x$  del centroide el objeto. El centroide del objeto se obtiene junto con las demás características durante el procesamiento de imagen. El error es la diferencia entre la coordenada  $x$  y el punto de ajuste. El punto de ajuste es el valor en pixeles entre el 20% y 80% del ancho de la imagen (200px). La banda de histéresis tiene un valor de  $\pm [5-30]$  px sobre el punto de ajuste. Estos valores son configurables desde el HMI, ítem 3.5.10. El control de lazo cerrado de posición se ilustra de una manera simplificada en la Figura 57.

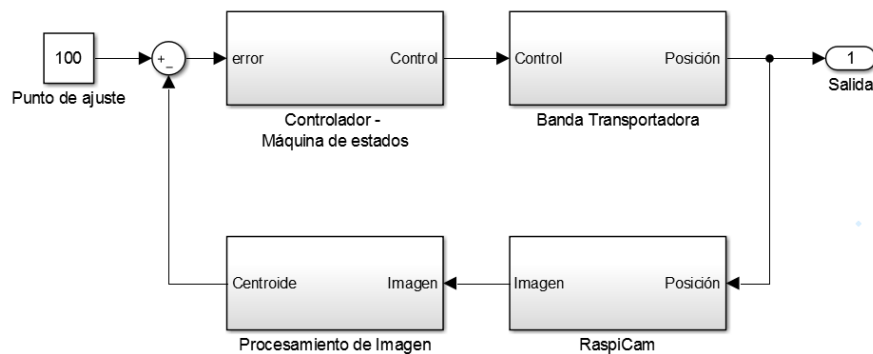


Figura 57 – Control de lazo cerrado de posición (Simplificado)

### 3.5.10 Adicional – Interfaz Máquina Humano.

El soporte que ofrece Simulink con respecto al RPI es bastante limitado. Como se vio anteriormente en la sección de Software utilizado, Simulink tiene muy pocos bloques para manipular el RPI, a diferencia con Matlab, que si tiene mayor integración.

El principal problema radica en cómo acceder a los datos, las variables, y las imágenes tomadas de los procesos en ejecución en el RPI. No hay bloques para leer ni guardar archivos desde el RPI. Simulink permite realizar simulaciones donde el código

se ejecuta en el RPI y los datos se visualizan en el PC, sin embargo, esta opción está sólo disponible cuando se trabaja en un único núcleo del procesador del RPI. La naturaleza de los algoritmos desarrollos requirió al menos dos núcleos.

Sólo hay dos formas de comunicación con el mundo exterior. Una es usando el bloque de ThingsSpeak, pero es de sólo escritura y para variables escalares, no hay de lectura. Por tanto, descartado este bloque. La otra forma es vía protocolo UDP que si cuenta con ambos bloques de lectura y escritura. Con estos dos bloques el RPI puede enviar y recibir información a otros equipos en la red local enviando datagramas con dirección IP, sin necesidad de estar conectados a la red. Para esto se requiere que el RPI y un dispositivo HMI – (Human Machine Interface) el cual es un PC o celular Android, estén conectados en LAN.

Adicionalmente, cabe la posibilidad de configurar el RPI como un servidor y aceptar datagramas del mundo exterior eliminando así la necesidad de una LAN. Con esto se logra montar el sistema en el Internet de las Cosas (Internet of Things – IoT)

Finalmente, se realiza una comunicación cliente-servidor donde el RPI es el servidor, y el computador o Smartphone es el cliente. Toda la información se envía vía protocolo UDP usando mensajes de 6 o más caracteres, 3 caracteres para el ID y 3 o más para el valor. Los objetivos del HMI son los siguientes:

- Tomar fotos desde la RaspiCam del servidor y guardarlas en el cliente.
- Ser capaz de establecer y resetear parámetros del servidor desde el cliente.
- Visualizar acciones del servidor en el cliente de manera *Soft Real-Time*.
- Presentar una interfaz gráfica amigable al usuario.
- Configurar parámetros de funcionamiento (Ej.: el punto de ajuste).

La parte más importante es la de captura y envío de fotos para alimentar los algoritmos de entrenamiento y poder construir el clasificador.

## 4 RESULTADOS

### 4.1 OBSERVACIONES GENERALES

El área de visión de la cámara debe ser de 13 cm \* 14 cm. En el montaje, el área es de 13.2 cm \* 13.73 cm. Hay ligeras variaciones debida a la tolerancia del campo de visión de la RaspiCam. El campo es óptimo y fue posible reconocer todas las herramientas.

La banda transportadora debe desplazar herramientas de hasta 3 kg y con una velocidad sin carga de 0.15 m/s. El motor debe consumir hasta 3.375 W, funcionar a 12 V y consumir 0.281 A de corriente. Se realizaron pruebas de medición y estas indican que la banda es capaz de transportar hasta 3.05 kg con una velocidad sin carga de 0.171 m/s La banda transportadora consume hasta 3 W, operando a 6 V con 0.500 A de corriente. La banda cumple con los requisitos de velocidad, peso y potencia.

La iluminación calculada es de 300 Lux. La iluminación actual es de 50 Lux. Esto significa que el sistema está subexpuesto. Se obtuvo un nivel de iluminación menor porque está calculada para configuración en iluminación directa, sin embargo, durante las pruebas se cambió la configuración a iluminación indirecta lo que hizo que se reduzca la intensidad de la iluminación, por el contrario, este cambio generó una luz uniforme que mejoró el algoritmo de procesamiento de imagen. Se hizo un cambio de intensidad de luz por mejoras en el algoritmo.

Debido a las limitaciones mecánicas de la banda transportadora, se requiere un PWM de al menos el 50-60% de potencia para poner la banda en marcha. Por lo que fue posible sólo manipular la banda en los rangos de velocidad media y alta.

La velocidad de procesamiento del RPI limita el número de muestreos por segundo, limitándolo a 5 muestras por segundo (200ms entre muestras) lo que produce una incertidumbre en el posicionamiento de 30 px. La distancia en el eje X es de 13.2 cm

con una resolución de 200px. Cada pixel equivale a 0.066 cm dando un total del EP de  $\pm 1.98$  cm. La precisión podría aumentar si la velocidad de la banda transportadora pudiera ser inferior al umbral de 50%, o si la velocidad del procesamiento del RPI pudiera ser mayor a las 200 muestras por segundo.

Hay dos variantes del Algoritmo de procesamiento de imagen, uno es usado en Matlab para procesar las imágenes que alimentan el clasificador, y otro es usado en Simulink en un bloque de función Matlab interpretada para procesar imágenes de manera continua en el RPI cuando éste está ejecutándose. Se trata con máxima prudencia de mantener ambas versiones fieles, idénticas entre sí. Es probable hacer cambios en uno de los algoritmos olvidándose del otro lo que adiciona un error innecesario al resultado.

Se le agregó una persiana a las entradas del módulo para mejorar el aislamiento de luz porque se observó que la luz incidente del exterior estaba entrando por la parte de ingreso y salida de las herramientas en la banda transportadora.

Con estos resultados se concluye que se cumplen los requerimientos técnicos dentro de un margen de error tolerable.

## 4.2 OBSERVACIONES SOBRE LA TOMA DE MUESTRAS

Es posible que la media y desviación estándar puedan estar sesgados porque el total de muestras puede ser no suficiente, y sólo se tomaron fotos en las partes centrales del área y en las partes del extremo. Además, aunque la naturaleza de las características es de una distribución normal, el sistema de adquisición de muestras es digital y por tanto genera valores discretos con un pequeño margen de error.

En la Figura 58 se nota una discontinuidad en los datos. Sin embargo, el interés del algoritmo radica en los valores extremos y no los del medio. Por ende se puede



asumir que todo lo que esté dentro del rango de máximos y mínimos tiene una probabilidad de pertenecer a esa clase.



Figura 58 – Discontinuidades en la muestra

### 4.3 ANÁLISIS ESTADÍSTICO DE LAS MUESTRAS

El objetivo principal del algoritmo constructor del clasificador es obtener una matriz de datos que contenga solamente la información esencial para clasificar correctamente futuras muestras. La interpretación de la matriz es la siguiente: cada característica individualmente se puede graficar como una distribución normal o binomial. Por ejemplo, con una muestra de  $n$  fotos tomadas para una herramienta, se pueden sacar 3 distribuciones normales (una por cada característica). La gráfica contiene la probabilidad de pertenecer a la clase según el valor de la característica. En la Tabla 10 se tabulan los resultados.

Tabla 10 – Matriz de resultados en forma de distribución normal

	Área $\mu$	Área $\sigma$	Eje Menor $\mu$	Eje Menor $\sigma$	Eje Mayor $\mu$	Eje Mayor $\sigma$
Alicate	1376	22	107	6	146	6
Bristol 1-16	184	14	15	1	86	2
Bristol 1-4	1137	75	31	1	144	2
Bristol 21-8	419	23	24	1	101	2
Bristol 3-16	732	47	31	1	115	2
Bristol 3-32	312	17	22	1	94	1
Bristol 5-32	582	41	27	1	108	1
Bristol 5-64	243	17	19	1	88	2
Bristol 7-32	914	54	32	1	126	1

Cinta	2005	42	72	1	74	1
Destornillador	608	8	8	0	110	1
Llave 10-10	2358	18	22	0	197	1
Llave 8-9	2616	12	24	0	211	1
Pasta	3830	23	70	0	71	0
Pinza Roja	3303	51	121	11	184	3
Regla	3911	18	42	0	126	0
Transportador	6654	43	83	0	143	1

Importante recordar que en toda distribución normal, la media se escribe con  $\mu$ , y la desviación estándar con  $\sigma$ . A la forma de la gráfica se le conoce como campana de Gauss. En la Figura 59 se hace un gráfico de la distribución de todas las características individuales de cada una de las herramientas.

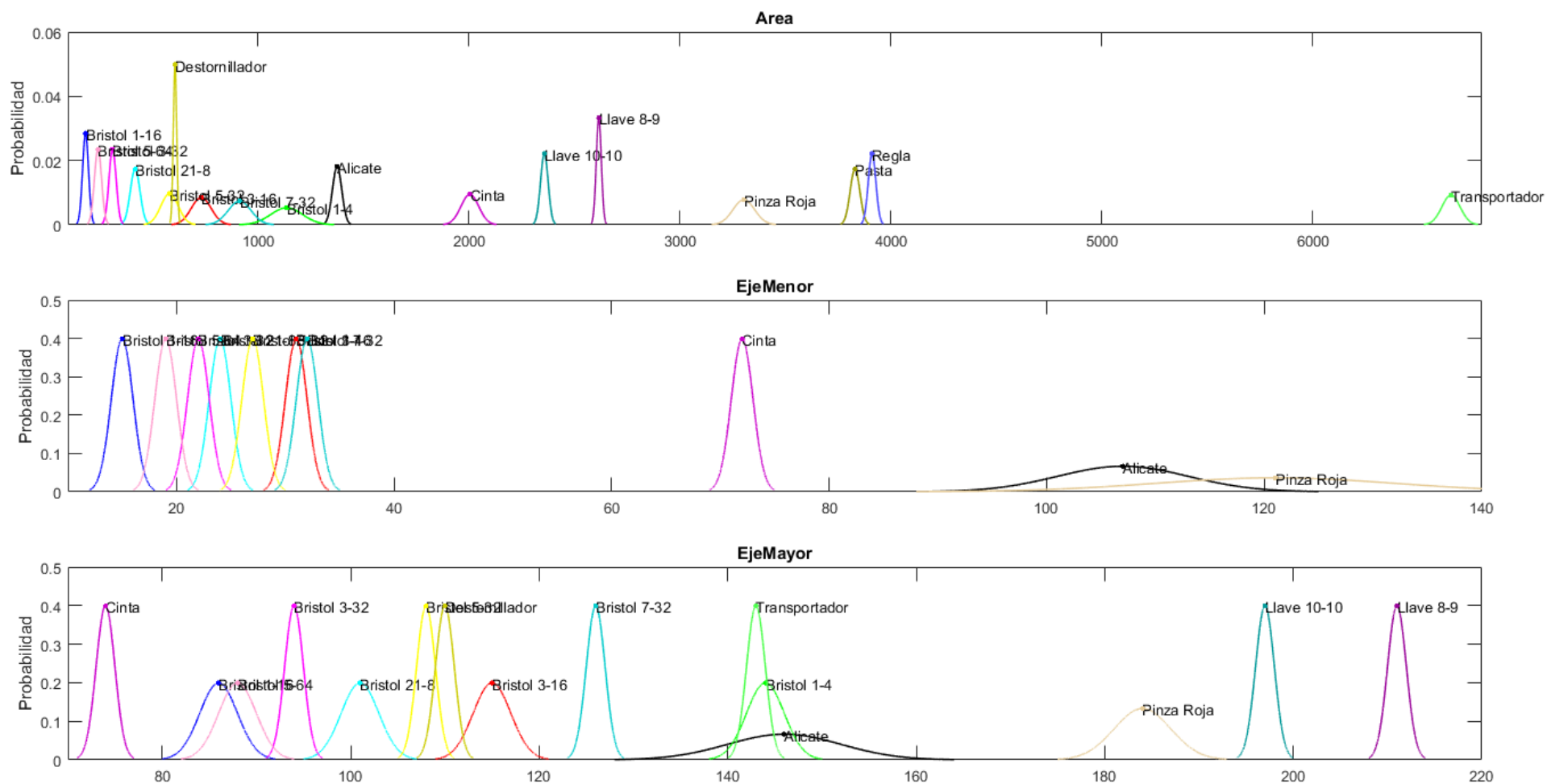


Figura 59 – Características modeladas como distribuciones normales

En la Figura 59 se puede apreciar cuán separables son las clases. Cada nuevo objeto siempre generará tres valores escalares (todas las medidas son en pixeles) para las tres características. Es posible ir a la gráfica, evaluar a cada valor a qué probabilidad corresponde con cada clase, y aquella clase que obtenga las 3 probabilidades más altas es la que el clasificador determinará como la clase más probable. Este enfoque se basa en la interpretación del clasificador Bayesiano Ingenuo.

Ahora bien surge un problema. Cuando una campana está sola, sin intersectarse con otra, se puede garantizar 100% que cualquier valor que esté dentro de su área corresponderá a su clase. El problema es, ¿qué pasa cuando el valor está en áreas intersectadas por dos o más campanas (es decir, pertenece a dos o más clases)?, especialmente en los puntos de intersección de campanas donde la altura (probabilidad) es idéntica para ambas.

La respuesta según el enfoque del clasificador bayesiano es examinar el vecindario del valor en cuestión y darle mayor probabilidad a la clase dominante. En este caso se otorga mayor probabilidad a aquella clase que tenga la menor desviación estándar. Esto se usa en el algoritmo para generar desempates, mencionado en el ítem 3.5.7.

Para manipular y visualizar mejor los datos se decide transformar la matriz obtenida en una de rangos de máximos y mínimos. Para esto se utiliza la regla 68 – 95 – 99.7 explicada en el ítem 2.4.3.

En la Tabla 11 se tabula la nueva matriz, donde reside la base de datos del clasificador, con los máximos y mínimos definidos como:

$$C_{Max} = C_{\mu} + 3C_{\sigma}$$

$$C_{Min} = C_{\mu} - 3C_{\sigma}$$

Tabla 11 – Matriz de resultados en forma de máximos y mínimos

	Área Mín.	Área Máx.	Eje Menor Min	Eje Menor Máx.	Eje Mayor Mín.	Eje Mayor Máx.
Alicate	1310	1441	89	124	129	163
Bristol 1-16	140	227	13	16	80	91
Bristol 1-4	910	1363	28	33	138	149
Bristol 21-8	350	487	22	26	96	105
Bristol 3-16	590	874	28	33	110	120
Bristol 3-32	261	362	20	24	91	97
Bristol 5-32	459	705	25	29	104	111
Bristol 5-64	193	293	17	21	83	92
Bristol 7-32	750	1077	30	34	122	130
Cinta	1879	2131	69	74	71	76
Destornillador	583	632	7	8	107	112
Llave 10-10	2303	2412	21	23	194	199
Llave 8-9	2581	2651	24	24	209	212
Pasta	3761	3898	69	71	70	71
Pinza Roja	3150	3455	88	154	175	192
Regla	3857	3964	41	42	125	127
Transportador	6523	6784	82	83	141	144

En la Figura 60 se grafican las muestras con Eje Menor vs Eje Mayor, en la Figura 61 se grafican Área vs Eje Mayor y en la Figura 62 se grafica Área contra Eje Mayor. Estas gráficas pueden considerarse en 2 dimensiones y muestran claramente al lector la alta separabilidad de clases.

En la Figura 63 se hizo un gráfico en 3 dimensiones que involucra al Área vs Eje Menor vs Eje Mayor. Acá es donde puede obtener la máxima información del muestreo y se puede especular qué tan bueno será la clasificabilidad.

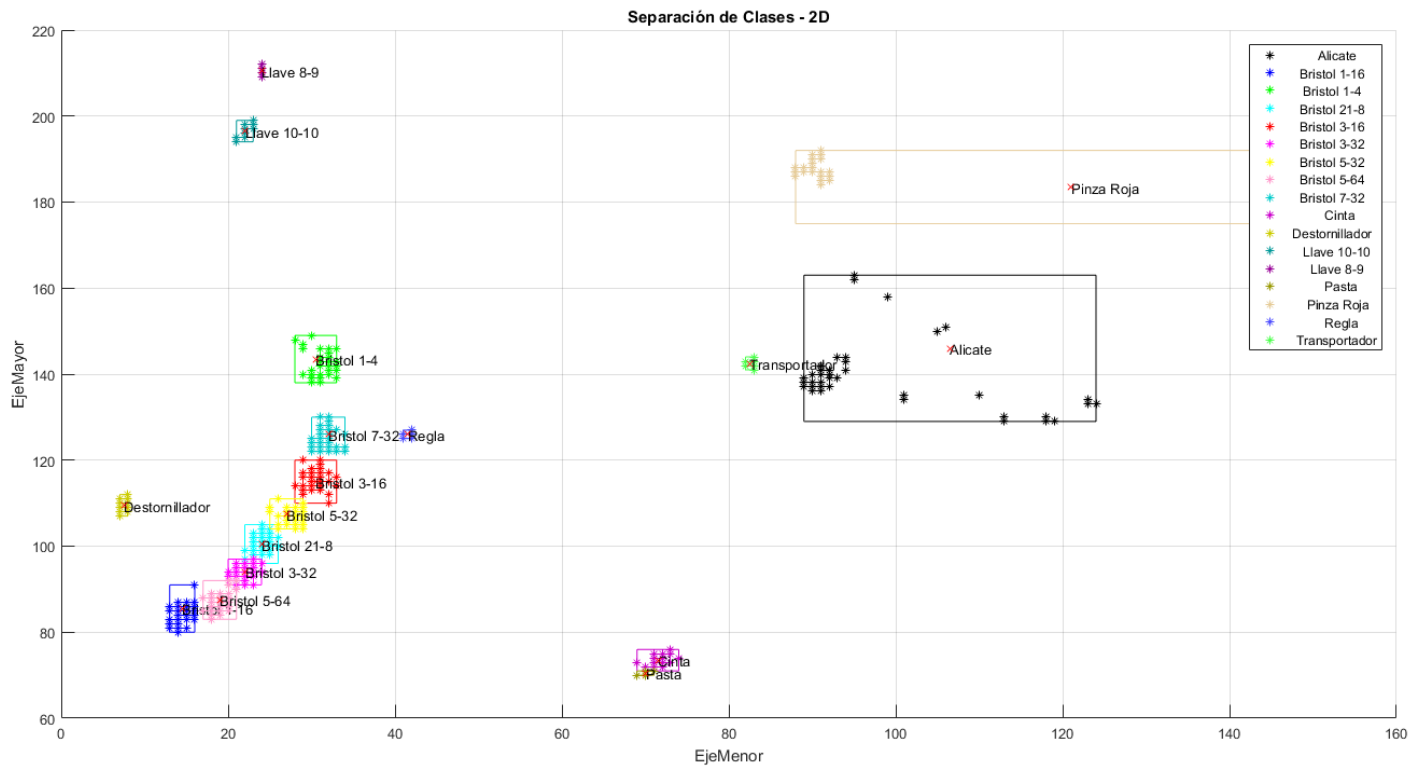


Figura 60 – Separación de clases 2D– Eje Menor vs Eje Mayor.

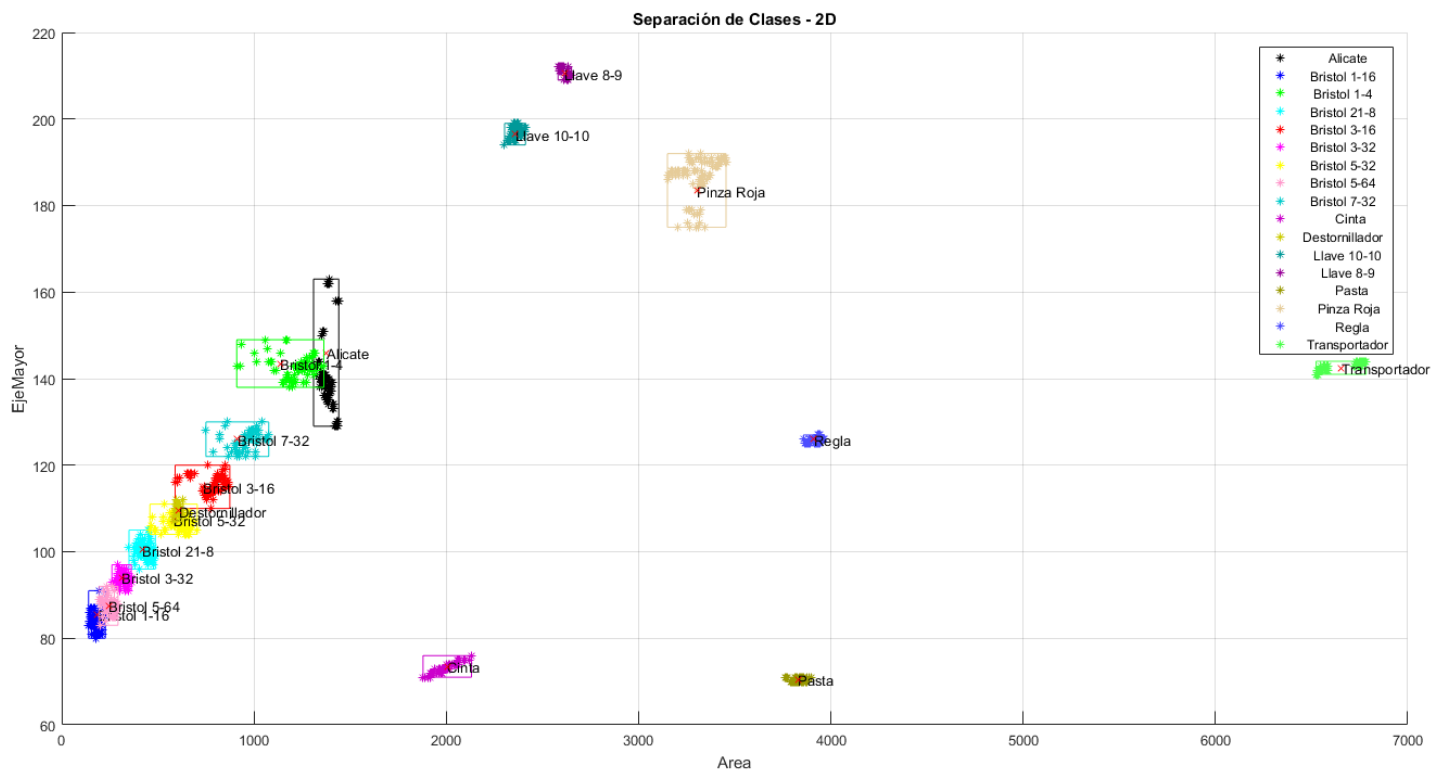


Figura 61 – Separación de clases 2D – Área vs Eje Mayor

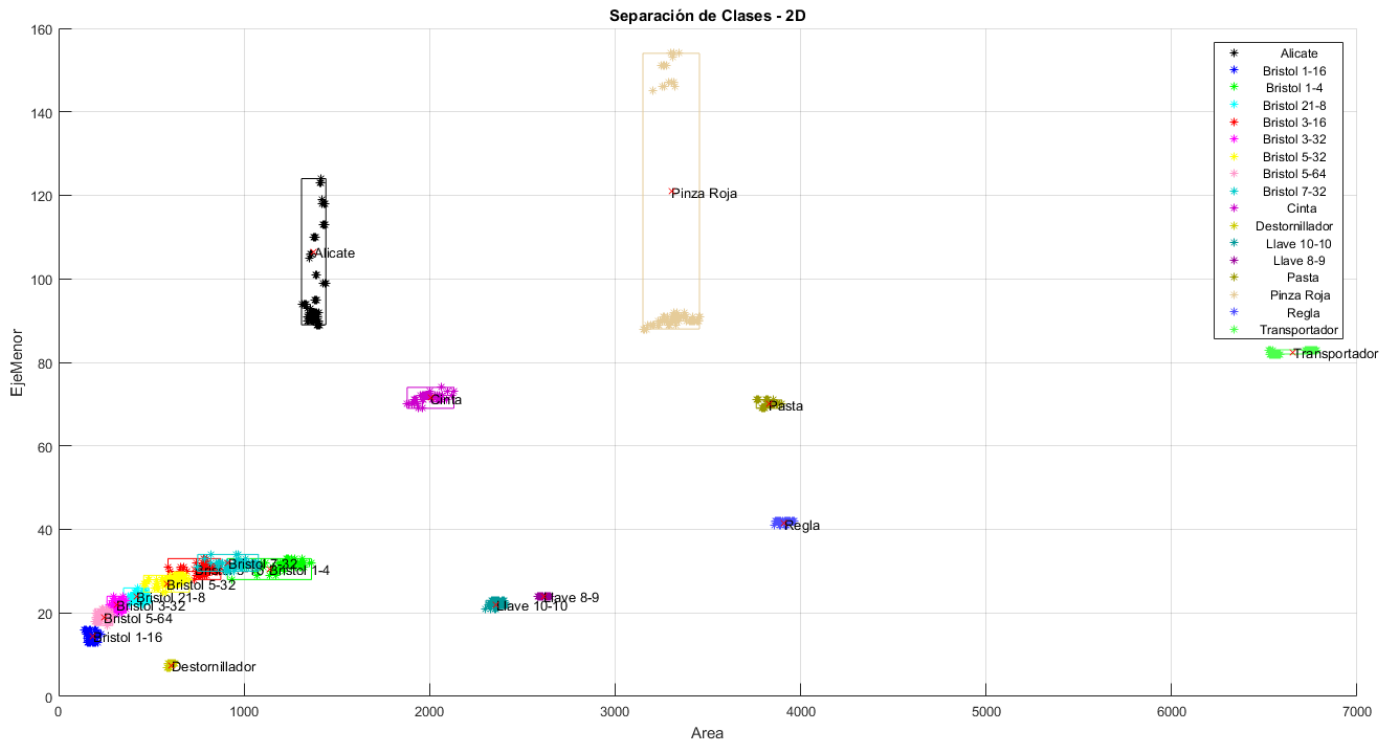


Figura 62 – Separación de clases 2D – Área vs Eje Menor

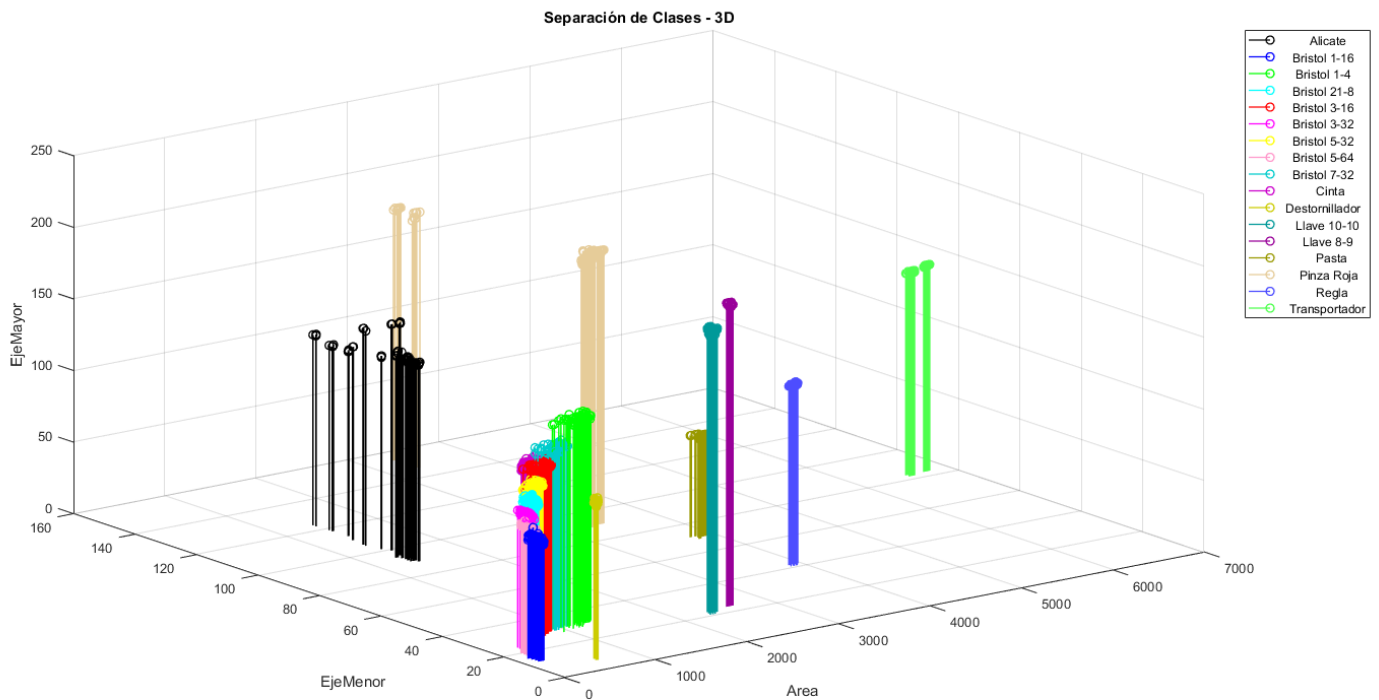


Figura 63 – Separación de Clases 3D – Área vs Eje Menor vs Eje Mayor

¿Qué pasaría si se tuviera 4 o más características? Ya de por si visualizar puntos en el espacio 3D es difícil por la falta de sensación de profundidad, graficar 4 dimensiones o más, hasta la n dimensión es mucho más complejo. Para esto hay otra técnica que permite visualizar n características comparando sus magnitudes. Esto es, haciendo una gráfica de datos paralelos. En la Figura 64 se puede apreciar una gráfica de datos paralelos generada por la aplicación *Classification Learner* de Matlab. Esta gráfica permite ver la alta separabilidad de las clases.

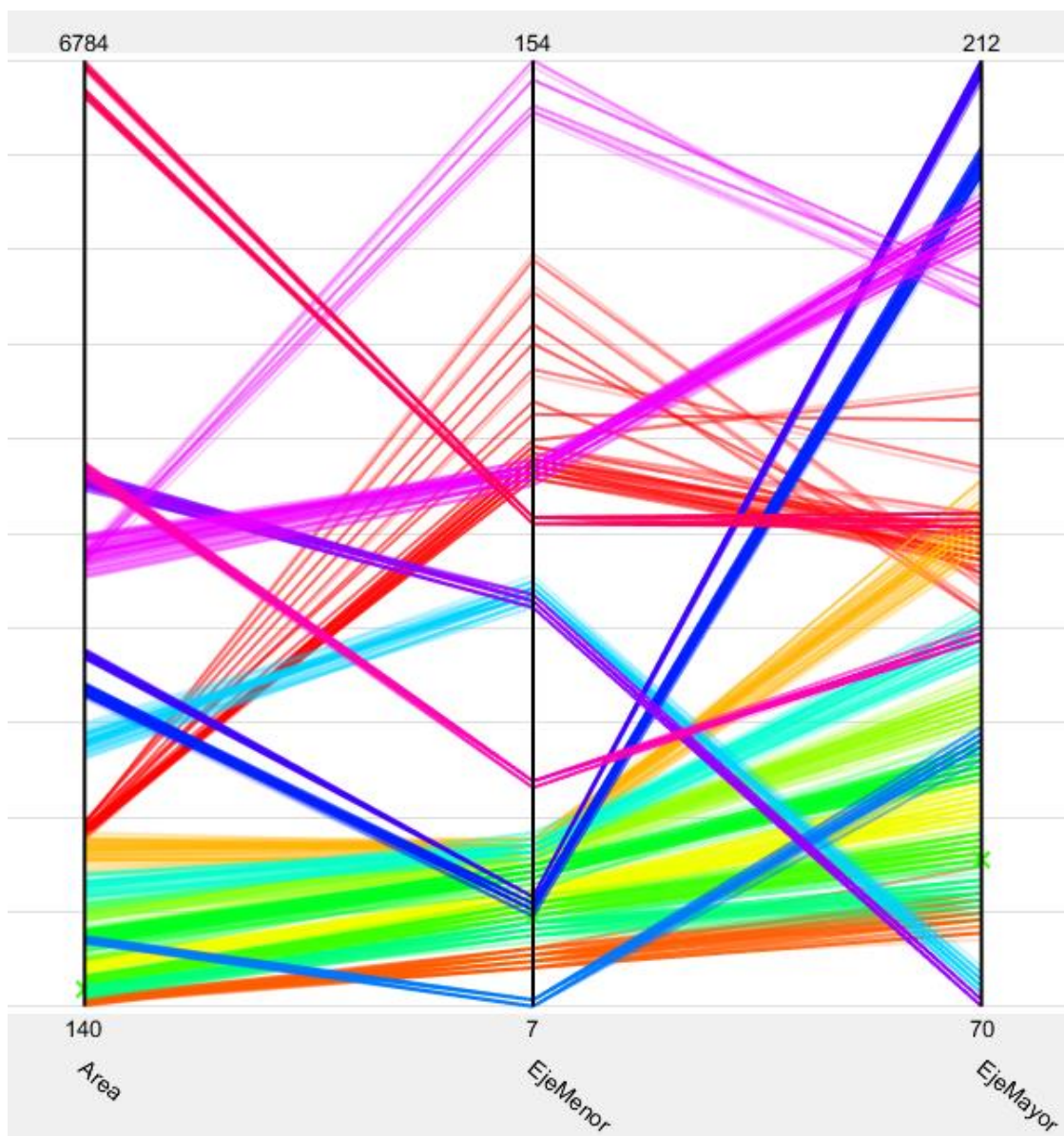


Figura 64 – Gráfica de datos paralelos



#### 4.4 EFECTIVIDAD DE LAS CARACTERÍSTICAS

A continuación se presentan estimaciones de qué tan efectivo es el algoritmo haciendo una comparación de la combinación de tres características. Primero, los resultados con cada característica individual, luego de a dos y finalmente se usan todas las tres características.

Se consideró usar el Perímetro pero existe una discrepancia en cómo Matlab calcula el perímetro con respecto a cómo Simulink lo hace. Se desconoce la fórmula exacta de cómo la calcula el programa. En un principio se hicieron algoritmos que calculaban de distintas maneras, pero ninguno coincidió con el Perímetro usado en el bloque *Blob analysis* de Simulink. Estas discrepancias llevaron a descartar el uso del Perímetro ya que esto genera que los dos algoritmos dependientes de las características (el entrenador y el clasificador) obtengan resultados distintos. Esto redujo el número inicial de características de cuatro a tres.

Los resultados se obtuvieron por medio de algoritmos para estimar la precisión utilizando el 100% de los datos de entrenamiento. Esto significa que los mismos datos que se usaron para entrenar el clasificador, se usaron también para probar si el clasificador podía clasificarlos correctamente. Estos resultados fueron satisfactorios. Las precisiones obtenidas se tabulan en la Tabla 12.

Tabla 12 – Precisión del clasificador según características

	Área	Eje Menor	Eje Mayor	Área Vs Eje Menor	Área Vs Eje Mayor	Eje Menor Vs Eje Mayor	Área Vs Eje Menor Vs Eje Mayor
Precisión	86,64%	66,73%	79,62%	95,28%	95,48%	99,06%	100%

Un clasificador de una sola característica alcanza resultados decentes, clasificando exitosamente entre el 66% y 87% de las muestras. Dos características resultan mucho más eficientes logrando clasificar exitosamente entre 95% y 99% de las muestras. Finalmente con el uso de 3 características se logra la precisión ideal, el 100%.

Esto demuestra que los objetos son altamente separables y este reconocimiento de visión artificial es capaz de detectar confiablemente diferentes herramientas basándose en los datos entrenados. En la siguiente sección se hacen ya pruebas de campo para determinar la precisión real.

#### 4.5 PRECISIÓN DE ACIERTOS

Se realizaron pruebas con la máquina totalmente operativa. De una muestra de 32 pruebas, clasificó correctamente 26. La precisión obtenida fue del 81.25%, la cual es bastante cercana a la precisión obtenida con los datos de entrenamiento. Se muestran los datos tabulados de las pruebas en la Tabla 13.

Tabla 13 – Pruebas de clasificación de objetos

Herramienta	Clasificación	Resultado
Alicate	Alicate	1
Bristol 1-16	Bristol 1-16	1
Bristol 1-4	Bristol 1-4	1
Bristol 21-8	Bristol 21-8	1
Bristol 3-16	Bristol 3-16	1
Bristol 3-32	Bristol 5-64	0
Bristol 5-32	Bristol 5-32	1
Bristol 5-64	Bristol 3-32	0
Bristol 7-32	Bristol 7-32	1
Cinta	Cinta	1
Destornillador	Destornillador	1
Llave 10-10	Llave 8-9	0
Llave 8-9	Llave 8-9	1
Pasta	Pasta	1
Regla	Regla	1
Transportador	Transportador	1
Alicate	Alicate	1
Bristol 1-16	Bristol 1-16	1
Bristol 1-4	Bristol 1-4	1

Bristol 21-8	Bristol 21-8	1
Bristol 3-16	Bristol 3-16	1
Bristol 3-32	Bristol 5-64	0
Bristol 5-32	Bristol 5-32	1
Bristol 5-64	Bristol 3-32	0
Bristol 7-32	Bristol 7-32	1
Cinta	Cinta	1
Destornillador	Destornillador	1
Llave 10-10	Llave 8-9	0
Llave 8-9	Llave 8-9	1
Pasta	Pasta	1
Regla	Regla	1
Transportador	Transportador	1
Total		81.25%

En la Figura 65 se muestra una de las pruebas realizadas donde el objeto probado obtuvo 100% de ser Bristol 5-32, 59% Destornillador, 33% Bristol 21-8 y 25% Bristol 3-16.

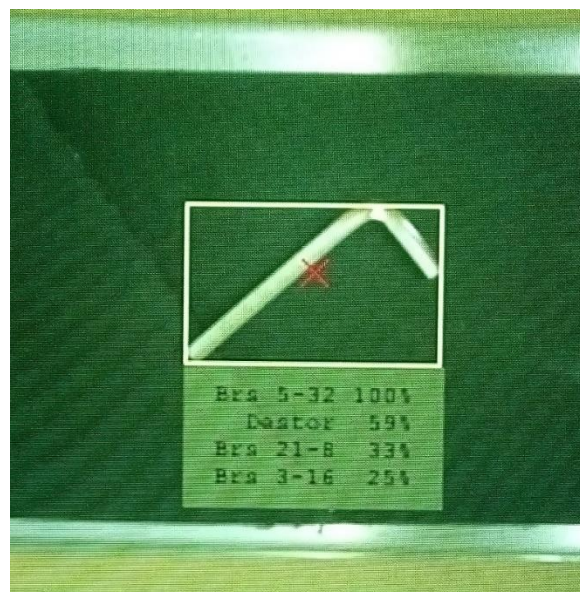


Figura 65 – Prueba de clasificación

El sistema es bastante robusto y puede identificar muy bien las piezas aun cuando no están centradas, no se evidencia una correlación entre los aciertos y la ubicación del objeto por lo que se descarta el error de posición como causante de fallos.

Los errores se atribuyen a la interacción de la luz ambiental con el proceso. Si bien la caja aísla buena parte de la iluminación exterior, aún logran filtrarse pequeñas cantidades de luz. La luz interna varía un poco debido a los efectos que causa la variación de voltajes en los LEDs. Sin embargo, estas variaciones son soportadas por los algoritmos. El procesamiento estadístico es capaz de discriminar las clases aún con estas variaciones. Para un futuro prototipo, no es necesario controlar la luz u otros parámetros pero si mejorar bastante el aislamiento.

#### 4.6 ERROR DE POSICIÓN

El control de la banda transportadora requiere un elevado nivel de sensado y de respuesta para ser óptimo. El Error de Posición (EP) se mide en pixeles (px) y se calcula como la diferencia de la posición actual y la posición ideal promediada en el número de pruebas realizadas tomadas.

Al principio, cuando había poca carga por parte de los algoritmos, se podía ejecutar el código a una velocidad de 1/10s. Con esto se calibró el control y se logró un EP de 10 px. A medida que el código se hizo más robusto, fue necesario bajar la velocidad eventualmente llegando a 1/5s, la mitad. Esto repercutió en una degradación tremenda sobre el controlador con un EP final de 30 px. En últimas se calibró lo mejor posible el control a cambio de un aumento considerable en el EP. Las pruebas se encuentran tabuladas en la Tabla 14.

Tabla 14 – Resultados error de posición

Herramienta	Posición Ideal (px)	Desfase (px)
Brs 3-32	50	1
Brs 3-32	50	19
Brs 3-32	50	8
Brs 3-32	50	23
Brs 3-32	50	6
Brs 3-32	50	24
Brs 3-32	50	2
Brs 3-32	50	14
Brs 3-32	50	2
Brs 3-32	50	15
Brs 3-32	20	5
Brs 3-32	20	6
Brs 3-32	20	30
Brs 3-32	20	6
Brs 3-32	20	1
Brs 3-32	20	7
Brs 3-32	20	1
Brs 3-32	20	8
Brs 3-32	20	7
Brs 3-32	20	0
Brs 3-32	80	5
Brs 3-32	80	6
Brs 3-32	80	10
Brs 3-32	80	0
Brs 3-32	80	2
Brs 3-32	80	20
Brs 3-32	80	20
Brs 3-32	80	12
Brs 3-32	80	15
Brs 3-32	80	5
Error de posición máximo (px)		30

La reducción de la velocidad de la banda transportadora al 60% de su máximo permite mejorar el EP a cambio de volver el controlador más lento, tomando más tiempo en centrar la pieza.

La banda tiene algunos aspectos por mejorar como: alinearla perfectamente, aumentar la relación de reducción del motor y aumentar el diámetro de los ejes. Con esto se podría trabajar a un ciclo de trabajo menor al 60%, idealmente cercano al 0% lo que significa un movimiento muy lento, casi nulo que permitiría un EP ideal casi 0.

La Figura 66 muestra un ejemplo del control de posición. La línea azul vertical del medio es el punto de ajuste establecido en 50% (100px). El texto en la parte superior “Ep: 22” significa Error de Posición: 22 pixeles. De esta manera se realizaron las pruebas de resultados del EP.

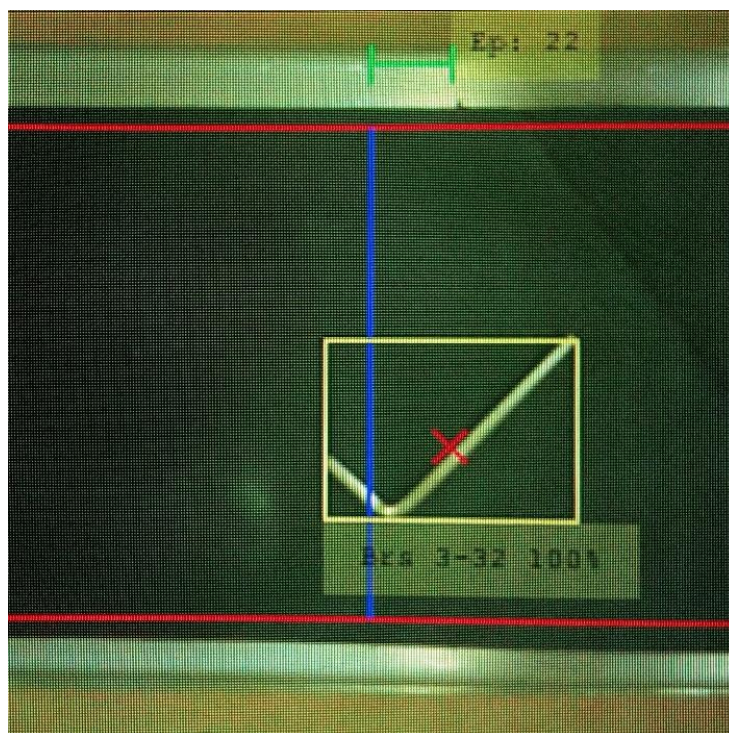


Figura 66 – Ejemplo de control de posición y clasificación con banda del 20%

## 4.7 COMPARACIÓN DE VARIOS CLASIFICADORES

Esta sección está más destinada a dejar posibilidades abiertas a futuro para mejorar los resultados del proyecto mediante el uso de otras técnicas de clasificación. Los siguientes resultados son todos ejecutados y corridos en Matlab, usando las 3 características disponibles, más no se pusieron a prueba los modelos en el RPI. Sirven como un punto de comparación para análisis y estudio a futuro.

Matlab tiene múltiples aplicaciones útiles para el desarrollo rápido de software. Una de ellas es la de *Classification Learner*. Con esta app se puede rápidamente desarrollar decenas de clasificadores de toda gama y comparar los rendimientos, la precisión de aciertos, entre otros parámetros.

El procedimiento es secuencial. Abrir la aplicación. Importar los datos estadísticos obtenidos de todas las muestras de cada una de las clases. Indicarle al programa cuáles son los vectores de características y cuáles los de resultados. Escoger los clasificadores a comparar y finalmente hacer clic en *train*. En menos de 10 minutos, se podrán apreciar decenas de clasificadores que tomarían semanas completas en estudiar e implementar.

Se deja como trabajo tentativo a futuro explorar esta aplicación de Matlab, probar diferentes modelos de clasificación y ponerlos a prueba. En la Tabla 15 se muestran los resultados de precisión para los distintos clasificadores simulados en Matlab.

Tabla 15 – Comparación de varios tipos de clasificadores

	<b>Clasificador</b>	<b>Tipo</b>	<b>Precisión</b>
1	Árbol	Árbol complejo	99.4%
2	Árbol	Árbol mediano	97.4%
3	Árbol	Árbol pequeño	35.4%
4	Discriminante Lineal	Discriminante Lineal	98.6%
5	Discriminante Cuadrático	Discriminante Cuadrático	99.9%
6	SVM	Lineal	99.9%
7	SVM	Cuadrático	99.9%
8	SVM	Cúbico	99.9%

9	SVM	Gaussiano Fino	99.9%
10	SVM	Gaussiano Medio	99.9%
11	SVM	Gaussiano bajo	99.6%
12	KNN	Fino	99.8%
13	KNN	Medio	99.7%
14	KNN	Bajo	95.1%
15	KNN	Coseno	93.9%
16	KNN	Cúbico	99.7%
17	KNN	Ponderado	99.8%
18	Ensamble	Árboles mejorados	63.5%
19	Ensamble	Árboles ponderados	99.7%
20	Ensamble	Discriminante del espacio	98.4%
21	Ensamble	Sub espacio KNN	98.5%
22	Ensamble	Árboles mejorados RUS	99.5%

Es importante notar que la mayoría de estos valores tienden al 100% lo que significa que la clasificabilidad de las herramientas es prácticamente perfecta. Sin embargo, se le da prioridad al valor de precisión de 81.25% obtenido en la sección anterior porque es el único valor con pruebas de campo realizadas.



## CONCLUSIONES

1. Se presentó un prototipo de un sistema de visión artificial completo. El principal enfoque fue el reconocimiento de objetos mediante el reconocimiento estadístico de patrones. Resultó posible clasificar herramientas por medio de un clasificador bayesiano. La precisión de aciertos es de 81.25%.
2. Resultó ser posible cerrar un lazo de control con una cámara para controlar la posición de una banda transportadora, sin embargo, no es recomendable usarla en procesos muy exigentes puesto que a lo que aumenta la carga de procesamiento se degrada el controlador. Aún con las altas diferencias de tiempo entre el objeto y su visualización en pantalla, el control de posición de la banda retroalimentados por cámara probó ser eficiente. El error de posición es de 30 px.
3. La iluminación es uno de los problemas más influyentes en el resultado de un sistema de visión artificial. El cuarto de captura de imagen con luz difusa logró hacer que la imagen de los objetos se mantenga bastante constante con mínima variación. Esto disminuyó considerablemente el error inducido por iluminación.
4. La RPI es un embebido limitado. Uno de los factores críticos fue el tiempo de procesamiento. A medida que creció el algoritmo, aumentaba este tiempo y disminuía la cantidad de imágenes que podía procesar por segundo. La mejor acción para maximizar este tiempo es reducir la resolución de imagen.
5. Se optó por utilizar los algoritmos más simples y rápidos. La mayoría de los bloques de Simulink ya están optimizados para código C. Para este proyecto fueron suficientes, pero para proyectos más grandes se necesitarán algoritmos más robustos capaces de tolerar más errores y condiciones extremas.
6. El modo en que fue diseñado el REP, permitió que el sistema esté abierto a la expansión y adición de más clases (herramientas). Basta con entrenar nuevas clases y agregarlas a la matriz de clases.

## BIBLIOGRAFÍA

- [1] D. Chitnis, N. Thejo kalyani , H. Swart y S. Dhoble, «Escalating opportunities in the field of lighting,» *Renewable and Sustainable Energy Reviews*, vol. 64, pp. 730-740, 2016.
- [2] Y. Xu, Z. Zheng, G. Lu y J. Yang, «Approximately symmetrical face images for image preprocessing in face recognition and sparse representation based classification,» *Pattern Recognition*, vol. 54, pp. 1-4, 2016.
- [3] A. Segovia, M. Garduño, M. Balcázar y L. Ledezma, «Automatic evaluation of solid state track detectors by artificial vision,» *Computers and Electrical Engineering*, vol. 39, pp. 1-3, 2013.
- [4] J. Gao, J. Yang, G. Wang y M. Li, «A novel feature extraction method for scene recognition basedon Centered Convolutional Restricted Boltzmann Machines,» *Neurocomputing*, vol. Article in Press, pp. 1-4, 2016.
- [5] M. Sofu, E. O. M. Kayacan y B. Cetisli, «Design of an automatic apple sorting system using machine vision,» *Computers and Electronics in Agriculture*, vol. 127, pp. 2-6, 2016.
- [6] M. R. J. GC y K. r. Alagusunda, «Application of Computer Vision Technique on Sorting and Grading of Fruits and Vegetables,» *Journal of Food Processing & Technology*, pp. 2-6, 2012.
- [7] R. A. O y A. ALAMUTU , «PROSPECTS OF COMPUTER VISION AUTOMATED SORTING SYSTEMS IN AGRICULTURAL PROCESS OPERATIONS IN NIGERIA,» *Agricultural Engineering International: the CIGR Journal of Scientific Research and Development*, vol. 3, pp. 2-6, 2005.
- [8] D.-W. Sun, «Computer Vision: An Objective, Rapid and Non-Contact Quality Evaluation Tool for the Food Industry,» *Journal of Food Engineering*, vol. 61, p. 1, 2004.
- [9] J. Cuida y C. C. Robayo, «DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE DETECCIÓN Y CLASIFICACIÓN AUTOMÁTICA DE OBJETOS MEDIANTE MÉTODO LOOK AND MOVE,» *Universidad San Buenaventura*, vol. 1, nº 1, p. 140, Noviembre 2014.
- [10] R. Szeliski , *Computer Vision: Algorithms and Applications*, Londres: Springer Science & Business Media, 2010.
- [11] S. Papert, *The Summer Vision Project*, MIT AI Memos, 2004.
- [12] M. A. Boden, *Mind as Machine: A History of Cognitive Science*, Clarendon Press., 2006.
- [13] K. Takeo, *Three-Dimensional Machine Vision*, Springer Science & Business Media, 2012.
- [14] N. Sebe, I. Cohen, A. Garg y T. S. Huang, . *Machine Learning in Computer Vision*, Dordrecht: Springer Science & Business Media., 2015.
- [15] W. Freeman, P. Perona y B. Scholkopf , «Guest Editorial: Machine Learning for Computer Vision,» *International Journal of Computer Vision.*, vol. 1, nº 1, p. 77, 2008.
- [16] D. Martin, «Basic Lighting Techniques for Machine Vision,» *Advanced illumination*, Inc., Rochester, 2016.

- [17] Vision Doctor, «Types of illumination,» [En línea]. Available: <http://www.vision-doctor.com/en/illumination-types.html>. [Último acceso: 20 febrero 2017].
- [18] D. Coleman, «Lighting and Optics for Machine Vision,» [En línea]. Available: <http://www.wernermn.com/assets/files/PDF/Solution%20Expo/Presentations/SE/SE07.LightingandOpticsforMachineVision.pdf>. [Último acceso: 4 Mayo 2017].
- [19] D. Martin, «Practical Guide to Machine Vision Lighting,» [En línea]. Available: [http://www.rauscher.de/downloads/public/datenblaetter/Machine-Vision-Lighting\\_Practical-Guide\\_2012.pdf](http://www.rauscher.de/downloads/public/datenblaetter/Machine-Vision-Lighting_Practical-Guide_2012.pdf). [Último acceso: 5 marzo 2017].
- [20] National Instruments, «A Practical Guide to Machine Vision Lighting,» National Instruments, 30 Enero 2017. [En línea]. Available: <http://www.ni.com/white-paper/6901/en/#toc1>. [Último acceso: 13 Abril 2017].
- [21] S. IMAGING, «Illumination techniques for industrial image processing,» Stemmer Imaging, [En línea]. Available: <https://www.stemmer-imaging.co.uk/en/technical-tips/illumination-techniques/>. [Último acceso: 1 Mayo 2017].
- [22] T. Kumar, «A Theory Based on Conversion of RGB image to Gray image,» *International Journal of Computer Applications* (0975 – 8887), vol. 7, n° 2, pp. 1-10, Septiembre 2010.
- [23] MathWorks, «What Is Image Filtering in the Spatial Domain?,» MathWorks, Febrero 2017. [En línea]. Available: <https://www.mathworks.com/help/images/what-is-image-filtering-in-the-spatial-domain.html>. [Último acceso: 5 Mayo 2017].
- [24] Mathworks, «Noise Removal,» Mathworks, Febrero 2017. [En línea]. Available: <https://www.mathworks.com/help/images/noise-removal.html>. [Último acceso: 5 Mayo 2017].
- [25] Opencv dev team, «Basic Thresholding Operations,» OpenCv, 4 Mayo 2017. [En línea]. Available: <http://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html>. [Último acceso: 4 Mayo 2017].
- [26] R. Fisher, S. Perkins, A. Walker y E. Wolfart, «Erosion,» 2003. [En línea]. Available: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/erode.htm>. [Último acceso: 6 Mayo 2017].
- [27] R. Fisher, S. Perkins, A. Walker y E. Wolfart, «Dilation,» 2003. [En línea]. Available: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm>. [Último acceso: 7 Mayo 2017].
- [28] M. Nixon y A. Aguado, «Low-level feature extraction (including edge detection),» de *Feature Extraction & Image Processing*, London, Elsevier, 2008, pp. 134-144.
- [29] Mathworks, «regionprops,» Mathworks, Febrero 2017. [En línea]. Available: <https://www.mathworks.com/help/images/ref/regionprops.html>. [Último acceso: 8 Mayo 2017].
- [30] M. Hansen, «Normal vs. Binomial: What are the hallmarks and differences?,» ST. ALBANS SCHOOL, 3 25 2003. [En línea]. Available: [http://staweb.sta.cathedral.org/departments/math/mhansen/public\\_html/23stat/handouts/normbino.htm](http://staweb.sta.cathedral.org/departments/math/mhansen/public_html/23stat/handouts/normbino.htm). [Último acceso: 11 Mayo 2017].

- [31] NYU, STERN, «Statistical Distributions,» 2016. [En línea]. Available: [http://people.stern.nyu.edu/adamodar/New\\_Home\\_Page/StatFile/statdistns.htm](http://people.stern.nyu.edu/adamodar/New_Home_Page/StatFile/statdistns.htm). [Último acceso: 9 Mayo 2017].
- [32] V. Mindel, «68 95 99.7 Rule in Statistics,» Statistics How To, 25 Diciembre 2015. [En línea]. Available: <http://www.statisticshowto.com/68-95-99-7-rule/>. [Último acceso: 10 Mayo 2017].
- [33] . A. R. Webb, Statistical Pattern Recognition, Malvern: John Wiley & Sons, Ltd., 202.
- [34] Quest Software Inc, «Naive Bayes Classifier,» Statistica, 2017. [En línea]. Available: <http://www.statsoft.com/Textbook/Naive-Bayes-Classifer>. [Último acceso: 12 Mayo 2017].
- [35] J. Garcia Fernandez, «Cálculo de instalación de alumbrado,» Oriol Boix, [En línea]. Available: <http://recursos.citcea.upc.edu/llum/interior/iluint2.html>. [Último acceso: 3 7 2017].
- [36] National Optical Astronomy Observatory, «Recommended Light Levels,» [En línea]. Available: [https://www.noao.edu/education/QLTkit/ACTIVITY\\_Documents/Safety/LightLevels\\_outdoor+indoor.pdf](https://www.noao.edu/education/QLTkit/ACTIVITY_Documents/Safety/LightLevels_outdoor+indoor.pdf). [Último acceso: 6 7 2017].
- [37] Carlos, «¿Cuánta potencia lumínica necesito?,» nergiza, 12 5 2014. [En línea]. Available: <https://nergiza.com/cuanta-potencia-luminica-necesito/>. [Último acceso: 3 7 2017].
- [38] RASPBERRY PI FOUNDATION, «The making of Pi,» RASPBERRY PI FOUNDATION, [En línea]. Available: <https://www.raspberrypi.org/about/>. [Último acceso: 03 May 2017].
- [39] JAMECO ELECTRONICS, «Raspberry Pi Pinout Diagram,» 2017. [En línea]. Available: <http://www.jameco.com/Jameco/workshop/circuitnotes/raspberry-pi-circuit-note.html>. [Último acceso: 3 May 2017].
- [40] RASPBERRY PI FOUNDATION, «RASPBERRY PI 3 MODEL B,» 1 Febrero 2016. [En línea]. Available: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. [Último acceso: 1 Marzo 2017].
- [41] J. Holton y T. Fratangelo, *Raspberry Pi Architecture*, Cambridge: Raspberry Pi Foundation, 2011.
- [42] RASPBERRY PI FOUNDATION, «CAMERA MODULE,» 2016. [En línea]. Available: <https://www.raspberrypi.org/documentation/hardware/camera/>. [Último acceso: 1 Marzo 2017].
- [43] Raspberry Pi Foundation, «RASPBERRY PI UNIVERSAL POWER SUPPLY,» [En línea]. Available: <https://www.raspberrypi.org/products/universal-power-supply/>. [Último acceso: 20 Abril 2017].
- [44] Tdrobótica, «Módulo Driver L298,» Sparkfun, Enero 2000. [En línea]. Available: <http://tdrobotica.co/modulo-driver-l298/544.html>. [Último acceso: 1 Marzo 2017].
- [45] Waveshare, «3.5inch RPi LCD (A), 320x480,» Waveshare, 2016 Diciembre 2016. [En línea]. Available: <http://www.waveshare.com/product/3.5inch-RPi-LCD-A.htm>. [Último acceso: 1 Marzo 2017].
- [46] WAYJUN Technology, «3W High Power LED,» Marzo 2011. [En línea]. Available: <http://www.wayjun.com/Datasheet/Led/3W%20High%20Power%20LED.pdf>. [Último acceso: 1 Marzo 2017].

- [47] Mathworks, «Supported and Compatible Compilers for R2017a,» Mathworks, 2017. [En línea]. Available: <https://www.mathworks.com/support/compilers.html>. [Último acceso: 1 Febrero 2017].
- [48] MathWorks, «Blob Analysis,» MathWorks, [En línea]. Available: <https://www.mathworks.com/help/vision/ref/blobanalysis.html>. [Último acceso: 30 Junio 2017].

## Anexos A – Especificaciones técnicas de la RaspiCam

Tabla 16 – Especificaciones de Hardware Raspicam

	<b>Camera Module v1</b>	<b>Camera Module v2</b>
Net price	\$25	\$25
Size	Around 25 × 24 × 9 mm	
Weight	3g	
Still resolution	5 Megapixels	8 Megapixels
Video modes	1080p30, 720p60 and 640	1080p30, 720p60 and 640
Linux integration	V4L2 driver available	V4L2 driver available
C programming API	OpenMAX IL and others	OpenMAX IL and others
Sensor	OmniVision OV5647	Sony IMX219
Sensor resolution	2592 × 1944 pixels	3280 × 2464 pixels
Sensor image area	3.76 × 2.74 mm	
Pixel size	1.4 μm × 1.4 μm	
Optical size	1/4"	
Full-frame SLR lens	35 mm	
S/N ratio	36 dB	
Dynamic range	67 dB @ 8x gain	
Sensitivity	680 mV/lux-sec	
Dark current	16 mV/sec @ 60 C	
Well capacity	4.3 Ke-	
Fixed focus	1 m to infinity	
Focal length	3.60 mm +/- 0.01	
Horizontal field of view	53.50 +/- 0.13 degrees	
Vertical field of view	41.41 +/- 0.11 degrees	
Focal ratio (F-Stop)	2.9	

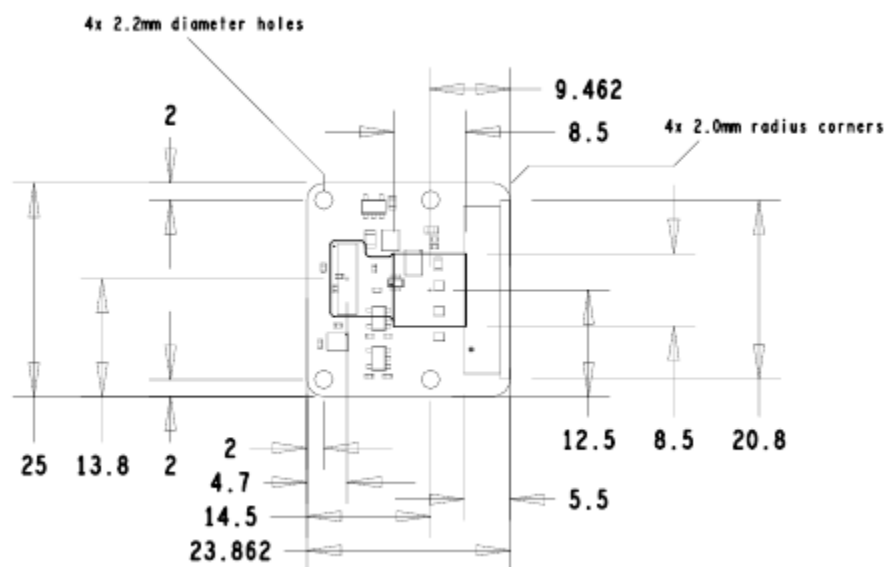
Tabla 17 – Características de Hardware Raspicam

Available	Implemented
Chief ray angle correction	Yes
Global and rolling shutter	Rolling shutter
Automatic exposure control (AEC)	No - done by ISP instead
Automatic white balance (AWB)	No - done by ISP instead
Automatic black level calibration (ABLC)	No - done by ISP instead
Automatic 50/60 Hz luminance detection	No - done by ISP instead
Frame rate up to 120 fps	Max 90fps. Limitations on frame size for the higher frame rates (VGA only for above 47fps)
AEC/AGC 16-zone size/position/weight control	No - done by ISP instead
Mirror and flip	Yes
Cropping	No - done by ISP instead (except 1080p mode)
Lens correction	No - done by ISP instead
Defective pixel cancelling	No - done by ISP instead
10-bit RAW RGB data	Yes - format conversions available via GPU
Support for LED and flash strobe mode	LED flash
Support for internal and external frame synchronization for frame exposure mode	No
Support for 2 × 2 binning for better SNR in low light conditions	Anything output res below 1296 x 976 will use the 2 x 2 binned mode
Support for horizontal and vertical sub-sampling	Yes, via binning and skipping
On-chip phase lock loop (PLL)	Yes
Standard serial SCCB interface	Yes
Digital video port (DVP) parallel output interface	No
MIPI interface (two lanes)	Yes
32 bytes of embedded one-time programmable (OTP) memory	No
Embedded 1.5V regulator for core power	Yes

Tabla 18 – Características de Software Raspicam

Picture formats	JPEG (accelerated), JPEG + RAW, GIF, BMP, PNG, YUV420, RGB888
Video formats	raw h.264 (accelerated)
Effects	negative, solarize, posterize, whiteboard, blackboard, sketch, denoise, emboss, oilpaint, hatch, gpen, pastel, watercolour, film, blur, saturation
Exposure modes	auto, night, nightpreview, backlight, spotlight, sports, snow, beach, verylong, fixedfps, antishake, fireworks
Metering modes	average, spot, backlit, matrix
Automatic white balance modes	off, auto, sun, cloud, shade, tungsten, fluorescent, incandescent, flash, horizon
Triggers	Keypress, UNIX signal, timeout
Extra modes	demo, burst/timelapse, circular buffer, video with motion vectors, segmented video, live preview on 3D models

En la Figura 67 se muestran las dimensiones de la RaspiCam.




 <b>Raspberry Pi</b> <a href="http://www.raspberrypi.org">www.raspberrypi.org</a> © Raspberry Pi 2015			
TITLE	RASPBERRY PI CAMERA MODULE V2.1		
DATE	12/11/2015	REF	RPI-CAM-V2_1
DRAWN	Mike Stimson	APVD	James Adams

Figura 67 – Dimensiones Raspicam



Error de Posición – Valor A 30 px. Precisión – Valor B 81.25%

Área mínima de imagen conectada – Valor C 100 px

Peso Objetivo – Valor D 3 kg

Peso máximo real – Valor E 3.05 kg

Velocidad Objetivo – Valor F 0.15 m/s

Velocidad Real – Valor G 0.171 m/s

Potencia Objetivo – Valor H 3.375 W

Voltaje Objetivo en el motor – Valor I 12 V

Corriente Objetivo en el motor – Valor J 0.281 A

Voltaje Consumido en el motor – Valor K 6 V

Corriente Consumida en el motor – Valor L 0.500 A

Potencia Consumida – Valor M 3 W

Radio del eje Objetivo – Valor N 8.731 mm

Diámetro del Eje Objetivo – Valor O 17.462 mm

Diámetro del Eje medido – Valor P 17.6 mm

Torque objetivo – Valor Q 2 kg.cm

Ángulo de apertura vertical – Valor R 41.41 +/- 0.11 grados

Ángulo de apertura horizontal – Valor S 53.50 +/- 0.13 grados

Eje X Objetivo – Valor T 13 cm

Eje X Real – Valor U 13.2 cm

Eje Y Objetivo – Valor V 14 cm

Eje Y Real – Valor W 13.73 cm

Altura cámara Objetivo – Valor X 17.19 cm

Altura cámara Real – Valor Y 17 cm

a = Largo del interior de la caja – Valor Z 0.34 m

b = Ancho del interior de la caja – Valor AA 0.22 m

h = Alto del interior de la caja – Valor BB 0.18 m

Objetivo General – Valor CC Destornillador, Alicates, Regla, Cinta y Llave Inglesa

Lúmenes Objetivo – Valor DD 56.87 lm

Lux Objetivo – Valor EE 300 Lux

Lux Reales – Valor FF 50 Lux