

RAE

1. **TIPO DE DOCUMENTO:** Trabajo de grado
2. **TÍTULO:** Desarrollo de un prototipo para el reconocimiento de objetos en una tienda de ropa para personas con discapacidad visual.
3. **AUTOR (ES):** Cristian Felipe de Jesús Casallas Prieto, Esteban Daniel Fonseca Moreno
4. **LUGAR:** Bogotá, D.C.
5. **FECHA:** 22 de febrero de 2018
6. **UNIDAD PATROCINANTE:** Universidad de San Buenaventura
7. **PALABRAS CLAVES:** Sistema Embebido, Adquisición de imágenes, Procesamiento digital de imágenes, Reconocimiento de objetos, Segmentación de imágenes, C++.
8. **DESCRIPCIÓN DEL TRABAJO:** El objetivo general de este proyecto de grado consiste en desarrollar un prototipo con el fin de reconocer las prendas en una tienda de ropa para las personas con discapacidad visual, utilizando el sistema embebido ODROID XU4. Este se encarga de ejecutar el algoritmo de adquisición de imágenes, procesamiento de imágenes y clasificación de prendas. Luego se realizó un algoritmo para alertar al usuario sobre los tipos de prendas por medio el adaptador de audio de ODROID. Antes de esto, se examinaron las fuentes de adquisición de imágenes y técnicas de reconocimiento de objetos, con el fin de evaluar e implementar el método más adecuado.
9. **LÍNEA DE INVESTIGACIÓN:** Análisis y procesamiento de señales, Sistemas embebidos.
10. **METODOLOGÍA:** Este trabajo se utiliza el método de tipo empírico-analítico enfocado a la análisis, teóricos, matemáticos y experimentales para el desarrollo del mismo.
11. **CONCLUSIONES:** Se desarrolló un prototipo con el fin de reconocer las prendas en las tiendas de ropa para las personas con discapacidad visual. Se encontraron dificultades para la captura de las imágenes de las prendas y la movilidad dentro del área de interacción del prototipo con las prendas de vestir. Esto se debió a que los establecimientos comerciales poseen diferentes diseños en sus áreas interiores, que afectan la posición, el color de la luz artificial y el color del fondo de las paredes y de los estantes, además de la forma de organización de las prendas y del espacio de movimiento dentro de la tienda.



UNIVERSIDAD DE
SAN BUENAVENTURA
BOGOTÁ, D.C.

DESARROLLO DE UN PROTOTIPO PARA EL RECONOCIMIENTO DE OBJETOS EN
UNA TIENDA DE ROPA PARA PERSONAS CON DISCAPACIDAD VISUAL.

CRISTIAN FELIPE DE JESÚS CASALLAS PRIETO

ESTEBAN DANIEL FONSECA MORENO

UNIVERSIDAD DE SAN BUENAVENTURA

FACULTAD DE INGENIERÍA

INGENIERÍA ELECTRÓNICA

BOGOTÁ, D.C.

2018

DESARROLLO DE UN PROTOTIPO PARA EL RECONOCIMIENTO DE OBJETOS EN
UNA TIENDA DE ROPA PARA PERSONAS CON DISCAPACIDAD VISUAL.

CRISTIAN FELIPE DE JESÚS CASALLAS PRIETO

ESTEBAN DANIEL FONSECA MORENO

TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE INGENIERO ELECTRÓNICO

DIRECTOR A CARGO DEL PROYECTO

ING. JOSÉ DAVID ALVARADO

UNIVERSIDAD DE SAN BUENAVENTURA, BOGOTÁ

FACULTAD DE INGENIERÍA

INGENIERÍA ELECTRÓNICA

BOGOTÁ, D.C.

2018

Dedicatoria

Dedico este trabajo principalmente a Dios, por haberme dado la vida y permitirme el haber llegado hasta este punto tan importante de mi vida. A mis padres, por todo el apoyo incondicional que me han brindado, las lecciones las cuales me guiaron para llegar a este momento y que me servirán para más adelante. A mi hermano Diego, que siempre estuvo presente y me ayudó en todas las necesidades. A mis abuelos, mi hermano Sebastián, mi tío Ricardo y mi prima Natalia, por la buena voluntad y la disposición que me ofrecieron en aquellos momentos que solicite de su ayuda y el apoyo incondicional.

Esteban Daniel Fonseca Moreno

Dedicatoria:

Dedico este trabajo a Dios, por haberme dado fortaleza para luchar en la vida y vencer todos los obstáculos. Mi reconocimiento a mis padres y hermanos por el apoyo recibido, por los valores y sentimientos inculcados, los cuales son pilares fundamentales para salir adelante en los momentos difíciles. A mi familia toda, por brindarme el apoyo incondicional y por compartir conmigo en todos los momentos. Y a mis amigos de infancia por darme aliento, amistad y conocimiento.

Cristian Felipe de Jesús Casallas Prieto

Agradecimientos

Agradezco a Dios por protegerme durante todo mi camino y por darme la sabiduría y la fuerza para superar aquellos obstáculos y dificultades a lo largo de mi vida.

A mi madre, que con su demostración de una madre ejemplar he aprendido a no rendirme ante nada y permanecer fuerte ante cualquier dificultad.

A mi padre, que a sus enseñanzas me permitió plantearme objetivos y centrarme en mis metas para siempre alcanzarlas sin importar los obstáculos.

A mis abuelos, Elizabeth y Anselmo, a mi hermano Sebastián, a mi tío Ricardo y a mi prima Natalia, por todo ese apoyo incondicional que me brindaron y la gran fe que tienen en mí.

A mis amigos y compañeros, que compartieron conmigo alegrías y fracasos.

Al Ing. José David Alvarado, director del proyecto, por su valiosa guía y asesoramiento en la realización de la misma.

Al Ing. Juan Gómez y Ing. Geovanny Villanueva, por los momentos de comprensión y reflexión los cuales ayudaron en mi formación académica y profesional.

A los establecimientos comerciales, que nos permitieron realizar las pruebas del dispositivo desarrollado y así concluir este proyecto.

Gracias a todas las personas que ayudaron, ya sea directa o indirectamente, en la realización de este proyecto.

Esteban Daniel Fonseca Moreno

Agradecimientos

Agradezco a Dios por protegerme y darme fuerzas para salir adelante y superar obstáculos y dificultades a lo largo toda de mi vida.

A mis padres, por brindarme su apoyo incondicional y formarme de manera ejemplar, siendo una persona perseverante, llena de valores y que, a pesar de mi limitación auditiva, puedo lograr cosas muy grandes si me lo propongo.

A mi hermano Juan David por sus consejos que me han ayudado a tomar decisiones y afrontar los retos que se me han presentado a lo largo de mi vida.

A mis abuelos, tíos, y primos quienes me ayudaron, dando su cariño y comprensión para fortalecer mi personalidad y saber que cuento con grandes personas que me apoyan en los proyectos que me proponga.

A mis amigos Elkin Rojas, Mauricio Posada y Julián González por compartir momentos de alegría, fe y demostrar con confianza que siempre estarán en mi corazón.

Al Ing. José David Alvarado, Ing. Edwin Gómez, Ing. Eduardo Castellanos y Ing. Feliz Gutiérrez por toda la colaboración brindada, su asesoría, valiosas enseñanzas y conocimientos durante el desarrollo de este proyecto y mi carrera profesional.

A los locales de ropa y así mismo a sus respectivos administrados quienes fueron ficha fundamental en el desarrollo de esta tesis, por el préstamo de sus instalaciones para poder llevar a cabo uno de los procesos de mi proyecto.

A Esteban Fonseca por haber sido un excelente compañero de tesis y amigo, por haberme tenido la paciencia y la motivación para seguir adelante en todos momentos.

Cristian Felipe de Jesús Casallas Prieto

Resumen:

Este proyecto de grado se orientó al desarrollo de un prototipo con el fin de reconocer las prendas en una tienda de ropa para las personas con discapacidad visual, utilizando el sistema embebido ODROID XU4. Este se encarga de ejecutar el algoritmo de adquisición de imágenes, procesamiento de imágenes y clasificación de prendas. Antes de esto, se examinaron las fuentes de adquisición de imágenes y técnicas de reconocimiento de objetos, con el fin de evaluar e implementar el método más adecuado y que cumple con los objetivos del proyecto.

Este sistema realizó la prueba de capturar las imágenes de prendas en un ambiente controlado, con el propósito de analizar y verificar los resultados de los valores de los descriptores de ropas — descripción de su forma y contorno. Con los resultados anteriores, el proyecto creó una clasificación de dichas prendas que será utilizada para realizar las pruebas en la tienda de ropa con el fin de conocer y verificar el funcionamiento del prototipo.

TABLA DE CONTENIDO

INTRODUCCIÓN	1
OBJETIVOS	5
<i>Objetivo general</i>	5
<i>Objetivos específicos</i>	5
ALCANCES Y LIMITACIONES DEL PROYECTO	5
<i>Alcances</i>	5
<i>Limitaciones</i>	5
1. DESARROLLO DE PROTOTIPO.....	7
1.1. SISTEMA EMBEBIDO	7
1.2. INSTALACIÓN DEL SISTEMA OPERATIVO	11
1.2.1. <i>Instalación de sistema operativo Android.</i>	12
1.2.2. <i>Instalación de sistema operativo Linux</i>	15
1.3. CONFIGURACIÓN DE DISPOSITIVO PARA ADQUISICIÓN DE IMÁGENES	18
1.4. ADQUISICIÓN DE LA IMAGEN EN C++	25
1.4.1. <i>Configuración de pines</i>	25
1.4.2. <i>Lectura de sensor de distancia</i>	29
1.4.3. <i>Reproducción de audio</i>	34
1.4.4. <i>Captura de fotograma</i>	37
2. PROCESAMIENTO DIGITAL DE IMÁGENES.....	40
2.1. TÉCNICAS DE PROCESAMIENTO DE IMAGEN	40
2.1.1. <i>Escala de grises</i>	40
2.1.2. <i>Ajuste de contraste</i>	41
2.2. SEGMENTACIÓN USANDO MÉTODOS DE UMBRALIZACIÓN (THRESHOLDING):	45
2.2.1. <i>Binarización</i>	46
2.2.2. <i>Método de umbralización Otsu</i>	49
2.2.3. <i>Resultados de la segmentación usando métodos de umbralización</i>	51
2.3. SEGMENTACIÓN POR COLOR USANDO CRECIMIENTO DE REGIONES	55
2.3.1. <i>Imagen segmentación usando textura y color (ISTC)</i>	55
2.3.2. <i>Crecimiento de regiones</i>	59
2.4. CONSTRUCCIÓN DEL DISPOSITIVO	68
3. EXTRACCIÓN Y CLASIFICACIÓN DE CARACTERÍSTICAS	70
3.1. RESTAURACIÓN Y RECONSTRUCCIÓN	70
3.1.1. <i>Dilatación</i>	71
3.1.2. <i>Erosión</i>	71
3.1.3. <i>Apertura y cierre</i>	72

3.2.	EXTRACCIÓN DE CARACTERÍSTICAS	75
3.3.	ANÁLISIS DE DATOS OBTENIDOS POR LOS DESCRIPTORES Y LOS ESQUEMAS DE REPRESENTACIÓN	81
3.3.1.	<i>Método de clasificación</i>	94
3.3.2.	<i>Sistema de clasificación</i>	99
4.	ANÁLISIS Y RESULTADOS	107
4.1.	REVISIÓN DEL ESPACIO DE TRABAJO Y CONFIGURACIÓN DEL SISTEMA	107
4.2.	PRUEBAS DEL SISTEMA EN AMBIENTE CONTROLADO	109
4.2.1.	<i>Resultados de clasificación para prendas de hombre</i>	110
4.2.2.	<i>Resultados de clasificación para prendas de mujeres</i>	113
4.3.	RESULTADO GLOBAL DE LAS PRUEBAS REALIZADAS	115
4.3.1.	<i>Resultados de clasificación para prendas de hombre</i>	115
4.3.2.	<i>Resultados de clasificación para prendas de mujer</i>	119
5.	CONCLUSIONES	122
	BIBLIOGRAFÍA.....	127
	ANEXO A	130
	ANEXO B	134
	ANEXO C	135
	ANEXO D	136
	ANEXO E	137

LISTADO DE FIGURAS

Figura 1. Componentes que conforman a un sistema embebido.....	7
Figura 2. Esquema de componentes del sistema embebido: ODROID XU4.....	11
Figura 3. Comparación de velocidad de lectura y escritura de diferentes tipos de memoria.....	12
Figura 4. Listado de sistemas operativos Android disponibles para ODROID.....	13
Figura 5. Interfaz de usuario del programa Etcher.....	14
Figura 6. Parte superior del ODROID XU4 señalando el puerto para la tarjeta de memoria micro SD.....	14
Figura 7. Fotografía donde se muestra la pantalla principal al iniciar el sistema operativo Android.....	15
Figura 8. Listado de sistemas operativos basados en Linux disponibles para el ODROID XU4.....	16
Figura 9.(a) Puertos de conexión entre adaptador micro SD y memoria eMMC. (b) Tarjeta eMMC conectada a adaptador micro SD. (c) Conexión de adaptador micro SD a puerto USB 2.0.....	16
Figura 10. Parte posterior del ODROID XU4 señalando el puerto para la tarjeta de memoria eMMC.....	17
Figura 11. Pantalla principal del sistema operativo Ubuntu Mate 16.04.....	18
Figura 12. Módulo oCam 5MP.....	20
Figura 13. Instalación de programa "subversion" desde la terminal de Linux.....	20
Figura 14. Confirmación para la instalación de programas dependientes.....	21
Figura 15. Interfaz gráfica del programa oCam Viewer.....	23
Figura 16.(a) Vista previa proveniente del módulo de cámara una vez conectado al programa. (b) Lista de formatos y resoluciones compatibles de la cámara. (c) Controles para ajustar las propiedades de la imagen adquirida por la cámara.....	24
Figura 17. Módulo Shifter Shield acoplado al ODROID XU4.....	26
Figura 18. Confirmación para instalación del software.....	27
Figura 19. (a) Diagrama de pines del ODROID XU4. (b) Sistema embebido ODROID XU4. (c) Circuito del obturador para la adquisición de la imagen.....	28
Figura 20. Sensor ultrasónico HC-SR04.....	29
Figura 21. Ángulo de medición y distancia medida por el sensor.....	30
Figura 22. Diagrama de bloques correspondiente al programa de medición de distancia usando el sensor ultrasónico.....	31
Figura 23. Conexión del sensor ultrasónico al Shifter Shield del sistema embebido ODROID.....	32
Figura 24. (a) Resultado de medición hecha por el sensor desde un punto A y mostrado en la terminal Linux. (b) Medición hecha con un flexómetro desde un punto A. (c) Resultado de medición hecha por el sensor desde un punto B y mostrado en la terminal Linux. (d) Medición hecha con un flexómetro desde un punto B, más cercano al sensor. (e) Resultado de medición hecha por el sensor desde un punto C y mostrado en la terminal Linux. (d) Medición hecha con un flexómetro desde un punto C, más lejano al sensor.....	33
Figura 25. (a) Módulo adaptador de audio por protocolo USB. (b) Componentes principales presentes en el adaptador de audio.....	34
Figura 26. Diagrama de bloques de la función de reproducción de sonidos.....	35
Figura 27. (a) Terminal Linux donde muestra distancia actual y reproducción de audio indicando que la distancia es correcta. (b) Medición entre sensor y un objeto en un punto neutral.....	36
Figura 28. (a) Terminal Linux donde muestra distancia actual y reproducción de audio indicando que debe alejarse. (b) Medición entre sensor y un objeto en un punto más cercano a este.....	36
Figura 29. (a) Terminal Linux donde muestra distancia actual y reproducción de audio indicando que debe acercarse. (b) Medición entre sensor y un objeto en un punto más lejano a este.....	37

Figura 30. Diagrama de bloques del programa principal.....	38
Figura 31. Imagen de captura de la cámara.	39
Figura 32. Imagen original en escala de grises.	41
Figura 33. (a) Imagen original con +10 de contraste, (b) Imagen original, (c) Imagen original con -10 de contraste.....	42
Figura 34. Ejemplo de aumento lineal de contraste.	42
Figura 35. Funciones de transformación de píxeles.....	43
Figura 36. Imagen original con bajo contraste (arriba) y la aplicación de dos instancias de la función de aumento de contraste (medio y abajo).	44
Figura 37. Imagen original (a). Histograma de la imagen original (b). Imagen binarizada con un umbral de 80 (c).	47
Figura 38. Ejemplo del algoritmo de Otsu.....	51
Figura 39. (a) Fotografía original de un pantalón para hombre. (b) Resultado obtenido del método de binarización estándar con valor de umbral 100. (c) Resultado obtenido usando el algoritmo de Otsu.	52
Figura 40. (a) Fotografía original de un pantalón para mujer. (b) Resultado obtenido del método de binarización estándar con valor de umbral 100. (c) Resultado obtenido usando el algoritmo de Otsu.	52
Figura 41. (a) Fotografía original de un buso. (b) Resultado obtenido del método de binarización estándar con valor de umbral 100. (c) Resultado obtenido usando el algoritmo de Otsu.	53
Figura 42. (a) Fotografía original de una camisa. (b) Resultado obtenido del método de binarización con valor de umbral 100. (c) Resultado obtenido usando el algoritmo de Otsu.	53
Figura 43. Diagrama de bloques del método ISTC.....	56
Figura 44. (a) Imagen original. (b) Resultado obtenido usando el método ISTC.	57
Figura 45. (a) Imagen de una camisa con un diseño en su textura. (b) Resultado de la segmentación usando el método ISTC.	57
Figura 46. (a) Fotografía de un pantalón. (b) Resultado de la segmentación de la imagen original.....	58
Figura 47. (a) Fotografía de un pantalón. (b) Resultado de la segmentación realizada con ISTC.....	58
Figura 48. Posicionamiento del "seed pixel" para la región de crecimiento (a). Proceso repetitivo para la región en crecimiento (b).	60
Figura 49. (a, e) Fotografía original con recorte. (b, f) Resultado obtenido con valor de umbral 0.10. (c, g) Resultado obtenido con valor de umbral 0.15. (d, h) Resultado obtenido con valor de umbral 0.20.	62
Figura 50. Conectividad 4 en el punto centro de la imagen (Círculo rojo) en la matriz de escala grises. ...	63
Figura 51. (a) Imagen de la camiseta roja. (b) Escala de gris de la imagen original. (c) Canal R de original. (d) Canal G de original. (e) Canal B de original. (f) Valor de pixel presente en centro de la imagen. (g) Crecimiento de regiones de escala de gris. (h) Crecimiento de regiones de canal R. (i) Crecimiento de regiones de canal G. (j) Crecimiento de regiones de canal B.	64
Figura 52. (a) Imagen de la camiseta verde. (b) Escala de gris de la imagen original. (c) Canal R de original. (d) Canal G de original. (e) Canal B de original. (f) Valor de pixel presente en centro de la imagen. (g) Crecimiento de regiones de escala de gris. (h) Crecimiento de regiones de canal R. (i) Crecimiento de regiones de canal G. (j) Crecimiento de regiones de canal B.	64
Figura 53. (a) Imagen de la camiseta roja. (b) Escala de gris de la imagen original. (c) Canal R de original. (d) Canal G de original. (e) Canal B de original. (f) Valor de pixel presente en centro de la imagen. (g) Crecimiento de regiones de escala de gris. (h) Crecimiento de regiones de canal R. (i) Crecimiento de regiones de canal G. (j) Crecimiento de regiones de canal B.	65
Figura 54. Círculo cromático.	66

Figura 55. Diagrama de bloques del método de segmentación crecimiento de regiones.....	67
Figura 56. (a) Batería de 12 voltios a 9000 miliamperios-hora. (b) Regulador de voltaje con entrada de 12 voltios y salida a 5 voltios.....	68
Figura 57. Componentes que conforman el sistema.	69
Figura 58. (a) Conjunto bidimensional A. (b) Elemento estructurado B, cuatro veces más pequeño que el conjunto A. (c) Resultado del proceso de la dilatación entre el conjunto A y B.	71
Figura 59. (a) Conjunto original A. (b) Elemento de estructura B. (c) Resultado de la erosión entre el conjunto A y el elemento estructurado B.....	72
Figura 60. (a, f, k) Imagen original de prenda. (b, g, l) Resultados de la segmentación. (c, h, m) Resultado del operador erosión de la segmentación. (d, i, n) Resultados del operador dilatación de la segmentación. (e, j, o) Resultados del operador cierre de la segmentación.....	73
Figura 61. (a) Imagen de operador cierre. (b) Resultados de los contornos que existen en la segmentación de operador cierre. (c) Resultado del relleno de regiones.	74
Figura 62. (a) Firma del círculo, (b) Firma del cuadrado.....	76
Figura 63. (a) Conjunto S y Conjunto H-S (sombreada); (b) Contorno dividido.	77
Figura 64. (a) Imagen segmentada de la camiseta, (b) Firma de la camiseta segmentada, (c) Imagen segmentada de la pantaloneta, (d) Firma de la pantaloneta segmentada.....	80
Figura 65. (a) Pantalón “Abierta”, (b) Pantalón “Cerrada”.	82
Figura 66. Resultados de grafica de Firmas en Matlab: (a) Buso, (b) Camisa, (c) Camiseta, (d) Chaqueta, (e) Pantaloneta “Abierta”, (f) Pantaloneta “Cerrada”, (g) Pantalón “Abierta”, (h) Pantalón “Cerrada”. ...	86
Figura 67. Resultados de distribución de los valores de las prendas para hombre en tres dimensiones.....	88
Figura 68. Resultados de grafica de Firmas en Matlab: (a) Blusa, (b) Falda, (c) Vestido.	90
Figura 69. Resultados de grafica de Firmas de Convex Hull en Matlab: (a) Blusa sin manga, (b) Blusa con manga, (c) Minifalda, (d) Falda, (e) Pantalón, (f) Vestido.	93
Figura 70. Resultados de distribución de los valores de las prendas para mujer en dos dimensiones.....	94
Figura 71. Ejemplo de clasificación lineal de dos clases.	95
Figura 72. Partición del árbol de decisiones.	97
Figura 73. Clasificación árbol de decisiones.....	97
Figura 74. Resultados de distribución de los valores de las prendas en dos dimensiones (a) Plano cartesiano de valor mínimo de radio y μ_{02} , (b) Plano cartesiano de m_{01} y μ_{02} , (c) Plano cartesiano de valor mínimo de radio y μ_{02}	100
Figura 75. Árbol de decisiones para el sistema de clasificación para prendas de hombre.....	103
Figura 76. Resultados de clasificación para prendas de mujeres con sus hiperplanos.....	104
Figura 77. Árbol de decisiones para sistema de clasificación de prendas para mujer.	105
Figura 78. Ejemplo de organización de las prendas para que el sistema lo analice correctamente.	107
Figura 79. Representación gráfica de los factores ambientales controlados para obtener una prueba exitosa.	110
Figura 80. Ejemplo de muestras recolectadas y usadas en el sistema.....	111
Figura 81. Ejemplo de muestras recolectadas de prendas para mujer, que fueron usadas en el sistema. .	114
Figura 82. Ejemplo de muestras recolectadas de prendas para hombre, que fueron usadas en el sistema.	116
Figura 83. (a) Imagen original del pantalón, (b) Imagen segmentada de pantalón, (c) Resultado no deseado de relleno de la imagen segmentada.	117

Figura 84. (a) Imagen original del pantalón, (b) Imagen segmentada de pantalón, (c) Resultado exitoso de relleno de la imagen segmentada.	118
Figura 85. Resumen de la segmentación de las prendas para hombre.	119
Figura 86. Resumen de la segmentación de las prendas para mujer.	121

LISTADO DE TABLAS

Tabla 1. Listado de sistemas embebidos seleccionados con las especificaciones características de CPU, GPU y memoria.	8
Tabla 2. Listado de sistemas embebidos seleccionados con las descripciones de puertos periféricos.	9
Tabla 3. Especificaciones técnicas del módulo de cámara oCam.	19
Tabla 4. Parámetros de funcionamiento e información adicional.	30
Tabla 5. Resultados de variables “Momentos” para las prendas camiseta y pantaloneta.	81
Tabla 6. Resultados de variables “Momentos centrales” para las prendas camiseta y pantaloneta.	81
Tabla 7. Resultado de variables “Momentos normalizados” para las prendas camiseta y pantaloneta.	81
Tabla 8. Estadísticas de las prendas para hombre que se obtuvieron en la prueba de ambiente controlado.	111
Tabla 9. Matriz de confusión de la primera clasificación para el ambiente controlado.	112
Tabla 10. Matriz de confusión de la segunda clasificación para el ambiente controlado.	112
Tabla 11. Estadísticas de las prendas para mujer que se obtuvieron en la prueba de ambiente controlado.	114
Tabla 12. Matriz de confusión de las prendas de mujeres en el ambiente controlado.	114
Tabla 13. Estadísticas de las prendas para hombre que se obtuvieron en la prueba de tienda de ropa.	116
Tabla 14. Matriz de confusión de la primera clasificación para la tienda ropa.	116
Tabla 15. Matriz de confusión de segunda clasificación para la tienda ropa.	117
Tabla 16. Comparación de los valores de los descriptores de Buso y Pantalón.	118
Tabla 17. Estadísticas de las prendas para mujer que se obtuvo en la prueba de tienda de ropa.	120
Tabla 18. Matriz de confusión de clasificación para prendas de mujer en la tienda ropa.	120

INTRODUCCIÓN

En Colombia —de acuerdo con el registro para la localización y caracterización de las personas con discapacidad, hecho por el Departamento Administrativo Nacional de Estadística (DANE) en 2010—, de los 2.018.078 de ciudadanos censados que presentan alguna discapacidad permanente, el 17,7% de ellos (348.620) padecen de algún tipo de limitación visual, ya sea agudeza visual baja o ceguera total (DANE Departamento Administrativo Nacional de Estadísticas, 2010). Esto les dificulta realizar actividades cotidianas, educativas o laborales, por lo que deben usar elementos o dispositivos que faciliten el desarrollo de estas actividades.

De acuerdo con el Instituto Nacional para Ciegos, esta población presenta necesidades en el desarrollo de actividades como reconocer objetos dentro de un entorno específico, por ello en el mercado actual existen dispositivos que ayudan a las personas con discapacidad visual en actividades como identificar productos, leer textos en objetos y conocer la denominación de un billete, entre otras. Los valores de estos dispositivos pueden estar entre los 1.000 y 2.500 dólares (USD). En Colombia, la mayoría de personas que requieren esta tecnología no poseen los suficientes recursos económicos para adquirirlos, ya que el 77% de los encuestados (de un total de 118.957) no poseen un ingreso financiero estable, además viven en estratos 1, 2 y 3 (DANE Departamento Administrativo Nacional de Estadísticas, 2010).

Por otro lado, existen compañías desarrolladoras de software que se centran en la creación de aplicaciones móviles con el fin de aportar una solución a las necesidades anteriormente mencionadas. Un ejemplo de estas es la aplicación móvil *TapTapSee*, la cual identifica objetos en una fotografía y le informa al usuario de manera auditiva el nombre de estos. La limitante de dicha aplicación es que el procesamiento para la identificación de la imagen no se realiza en tiempo real,

además requiere de una conexión a internet para comparar la imagen entrante de la cámara con la base de datos con el fin de determinar qué tipo de objeto es. Otra desventaja con el uso de teléfonos inteligentes (*smartphone*) es que estos no fueron diseñados para ser usados por personas con discapacidad visual. Aun así, existen aplicaciones de accesibilidad que permiten ajustar el tipo de interacción entre el usuario y el dispositivo de acuerdo con su discapacidad —como lectores de pantalla o lectores de gestos para la ejecución de acciones dentro de programas—, pero para su activación, este tipo de usuario debe solicitar la ayuda de otra persona que no tiene esta clase de discapacidad.

Por otro lado, existen instituciones académicas y científicas que abordan estos temas, con el fin de solucionar otro tipo de problemas que abarcan el mismo tema, por ejemplo, la investigación realizada por (Gallagher & Chen, 2008), la cual consiste en segmentar ropa por medio de la identificación de personas. El proyecto consiste en: Primero, identificar la cara de la persona con la prenda, y allí se recorta usando el método de *Graph Cut*, a continuación, mediante la búsqueda e implementación de una máscara global para las prendas, se identifica la máscara de la prenda presente en la imagen que se esté procesando y la recorta nuevamente, utilizando el método de *Graph Cut*. Para aumentar la calidad de la máscara, se toma varias imágenes de la misma persona, que luego servirán para construir un mejor modelo de la prenda.

Los autores (Yang & Yu, 2011) también presentan un proyecto similar al anterior, este se basa en un sistema de reconocimiento de prendas en tiempo real, con el objetivo de vigilar y resguardar la seguridad dentro de establecimientos. El sistema obtiene el fotograma del video en tiempo real, el cual detecta y rastrea la cara de la persona, posteriormente, identifica el contorno del cuerpo de la persona y realiza una segmentación por color a las prendas, este proceso se realiza para unificar el color de las prendas y, a su vez, eliminar el fondo de la imagen. Para el reconocimiento de las

prendas, el sistema divide en dos partes la imagen, para identificar la prenda, tanto en la parte superior como en la inferior del cuerpo de la persona, luego, mediante el método histograma orientado en gradientes y descriptores *BoW*, genera un valor para cada prenda, que servirá como valor de entrada en un clasificador basado en *SVM*, para categorizar cada una de las prendas.

Con base en las investigaciones realizadas, se plantea el desarrollo de un sistema que funcione como una herramienta de apoyo dirigida a las personas con discapacidad visual, que tenga como propósito reconocer prendas en una tienda de ropa, y así ayudar a este tipo de personas a realizar una actividad cotidiana como es escoger su vestuario.

Este proyecto se realiza en cuatro (4) etapas. La primera consiste en consultar distintas fuentes de información relacionada con la adquisición y almacenamiento de fotogramas provenientes de un módulo de cámara para su posterior procesamiento. La siguiente etapa se enfoca en analizar e identificar las técnicas de procesamiento de imágenes adecuadas para el reconocimiento de prendas de vestir —tales como buso, camisa, camiseta, chaqueta, pantalón y pantaloneta. En tercer lugar, se procede con el desarrollo de un sistema de clasificación basado en descriptores de contorno y de regiones, que permita diferenciar las formas de las prendas presentes; del mismo modo se procede con un sistema de advertencia auditivo que informa al usuario acerca de los objetos reconocidos en la imagen procesada. Después de desarrolladas las tres (3) primeras etapas, se instala al sistema una fuente de alimentación portátil (batería) con el fin de que pueda ser transportado a cualquier lugar.

Para obtener una retroalimentación de la calidad y el tiempo del procesamiento realizado, se validará y verificará el funcionamiento del sistema por medio de pruebas dentro de una tienda de

ropa, además el desarrollo de este proyecto contribuirá a futuras investigaciones que abarquen temas relacionados a los expuestos en este documento.

Objetivos

Objetivo general

Desarrollar un prototipo para el reconocimiento de objetos en una tienda de ropa para personas con discapacidad visual.

Objetivos específicos

- Implementar un algoritmo de adquisición de imágenes en un sistema embebido.
- Evaluar e implementar las técnicas de procesamiento de imágenes en un sistema embebido para el reconocimiento de objetos.
- Desarrollar un sistema de clasificación de imágenes en un sistema embebido para reconocer los objetos.
- Verificar y validar el funcionamiento del sistema de procesamiento de imágenes.

Alcances y limitaciones del proyecto

Alcances

- El dispositivo tendrá un sistema de advertencia auditiva de los objetos dentro del entorno.
- Se seleccionará las técnicas de procesamiento más eficaces y que muestre buenos resultados para la implementación en el sistema.

Limitaciones

- El dispositivo será portátil pero no se enfocará en la parte ergonómica.

- El alcance que tendrá el dispositivo para reconocer los objetos será máximo de un metro de distancia.
- La batería durará máximo 1 hora.
- Para el prototipo se usará el sistema embebido ODRIOD XU4 con una cámara integrada de 5 megapíxeles.
- El sistema será capaz de procesar el contorno de ropa para hombre tales como camisas, camisetas con y sin mangas, pantalones, chaquetas y sacos, para las mujeres, este solo reconocerá prendas tales como faldas, vestidos, blusas con y sin mangas.
- El sistema no va reconocer las prendas por color ni textura y textos impresos en las prendas

Capítulo 1

DESARROLLO DE PROTOTIPO

1.1. Sistema embebido

Los sistemas embebidos poseen una unidad central de procesamiento con una arquitectura específica, cuentan con una memoria RAM y ROM. También poseen puertos de entrada y salida, los cuales pueden estar conectados a dispositivos periféricos. En la Figura 1 se muestra estos y otros componentes que conforman dicho sistema. Estos dispositivos pueden reconocer señales entrantes, desarrollar cálculos, generar señales salientes y ejecutar procesos en tiempo real, con el propósito de ejecutar una o varias tareas específicas (Valvano, 2014).

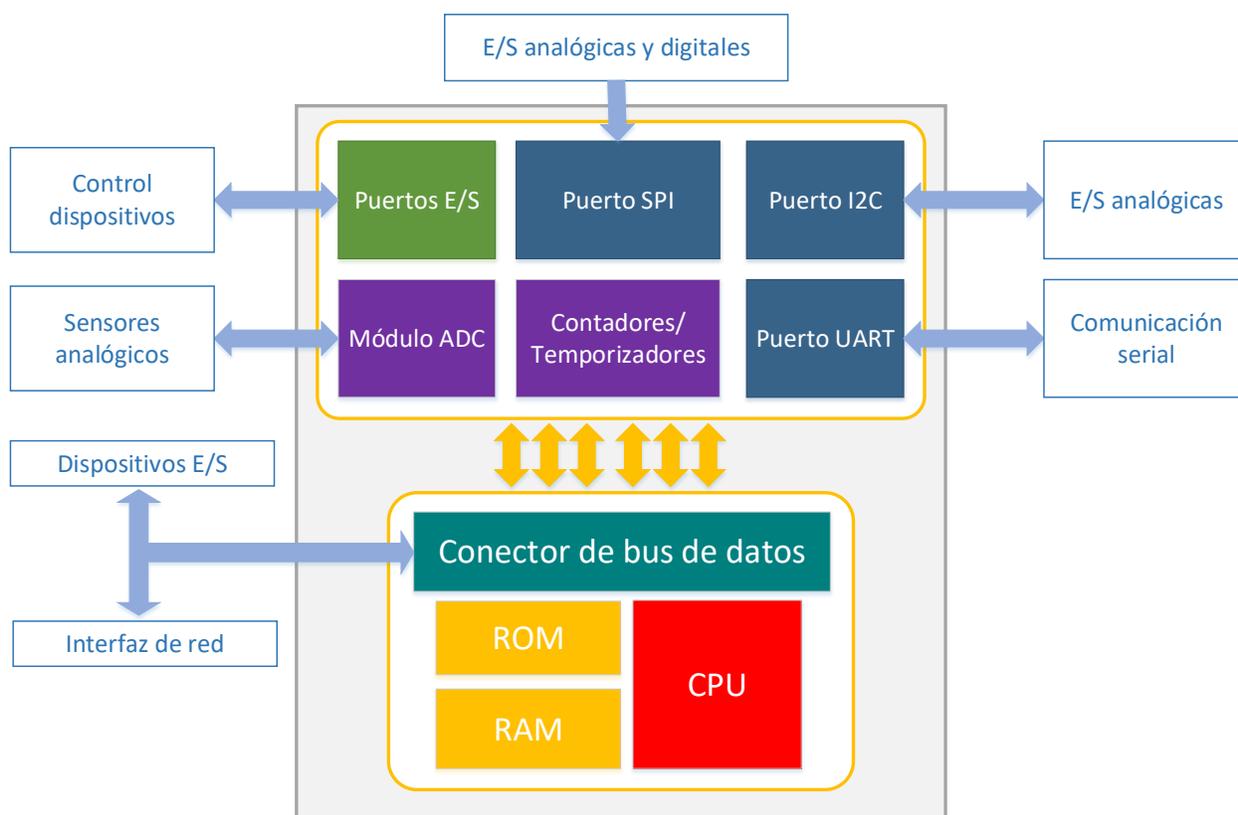


Figura 1. Componentes que conforman a un sistema embebido.
Fuente (Valvano, 2014)

En la actualidad existen dos tipos de sistemas embebidos: uno de ellos consiste en un microcontrolador que depende de un software externo para desarrollar, ya que este no tiene implementado un sistema operativo para realizar dicha tarea. La ventaja de estos dispositivos es que son compactos, no usan mucha energía y son de bajo costo. Por otra parte, están los sistemas embebidos con una unidad central de procesamiento de alto rendimiento mejor que un microcontrolador, estos sistemas utilizan un sistema operativo como plataforma de desarrollo con la opción de manipular, tanto su software como su hardware (Valvano, 2014).

Con el fin de determinar qué sistema embebido se adapta mejor a los objetivos del proyecto, se realiza una comparación acerca de las placas que hay en el mercado, en donde se analizaron aspectos como; la velocidad del procesador, la capacidad de la memoria RAM y la unidad gráfica de procesamiento que viene incorporada, los periféricos disponibles para este y su costo.

Los resultados de esta comparación se manifiestan en la Tabla 1 y

Tabla 2, donde se muestran las tres características de la unidad central de proceso que son: la arquitectura, el número de núcleos y la frecuencia de reloj; junto con la memoria RAM. Estas tablas permiten además realizar una comparación de los mencionados sistemas, y los puertos de entradas/salidas y los periféricos que están disponibles en estos dispositivos, respectivamente.

Tabla 1. Listado de sistemas embebidos seleccionados con las especificaciones características de CPU, GPU y memoria.

NOMBRE	CPU	GPU	RAM (GB)	Precio (\$USD)
UDOO X86 ULTRA	Intel N3710 x86 Quad-Core 2.56 GHz	Intel® HD 405 Graphics, 16 execution units up to 700 MHz	8	267
UDOO X86 ADVANCED PLUS	Intel N3710 x86 Quad-Core 2.56 GHz	Intel® HD 400 Graphics, 12 execution units up to 640 MHz	4	165
ODROID XU4	Exynos 5 ARM Octa-Core (A15 2.0 GHz + A7 1,4GHz)	ARM Mali-T628 @695MHz	2	58

NOMBRE	CPU	GPU	RAM (GB)	Precio (\$USD)
UDOO X86 BASIC	Inter x5 (X86) - E8000 Quad-Core 2.0 GHz	Intel® HD Graphics, 12 execution units up to 320 MHz	2	125
BANANA PI M3	Allwinner A83T ARM A7 Octa-Core 1.8 GHz	PowerVR SGX544MP	2	68
RADXA ROCK	Rockchip RK3188 ARM A7 Quad-Core 1.6 GHz	Mali-400MP4	2	100
ODROID C2	Amlogic S905 ARM A53 Quad-Core 1.5 GHz	Mali-450MP3 +2VS @700 MHz	2	46
ORANGE PI PLUS 2	Allwinner H3 ARM A53 Quad-Core 1.53 GHz	ARM Mali-400 MP2 @600MHz	2	49
RASPBERRY PI 3 MODEL 3	Broadcom BCM2837 ARM A53 Quad-Core 1.2 GHz	Broadcom VideoCore IV	1	35
DRAGONBOARD 410c	Qualcomm Snapdragon 410 ARM A53 Quad-Core 1.2 GHz	Qualcomm Adreno 306	1	75
BEAGLEBONE BLACK WIRELESS	TI Sitara AM335x ARM A8 Single-Core 1.0 GHz	PowerVR SGX530	0.51	55

Fuente (Kearney, 2016)

Tabla 2. Listado de sistemas embebidos seleccionados con las descripciones de puertos periféricos.

NOMBRE	USB		Almacenamiento			Conectividad (Ethernet / WiFi / BT)	Comunicación (I ² C / SPI)	Puertos I/O	
	2.0	3.0	eMMC	Ranura Flash	SATA			GPIO	Análogo
UDOO X86 ULTRA	No	3	Si (incorporado)	microSD	Si	Si / Si / Si	Si / Si	36	ADC, PWM
UDOO X86 ADVANCED PLUS	No	3	Si (incorporado)	microSD	Si	Si / Si / Si	Si / Si	36	ADC, PWM
ODROID XU4	1	2	Si	microSD	No	Si / No / No	Si / Si	40	ADC
UDOO X86 BASIC	No	3	No	microSD	Si	Si / Si / Si	Si / Si	36	ADC, PWM
BANANA PI M3	2	No	Si (incorporado)	microSD	Si	Si / Si / Si	Si / Si	40	12 bit ADC
RADXA ROCK	2	No	No	microSD	No	Si / Si / Si	Si / Si	80	ADC, PWM
ODROID C2	4	No	Si	microSD	No	Si / No / No	Si / Si	40	2x12 Bit ADC, PWM
ORANGE PI PLUS 2	4	No	No	microSD	Si	Si / Si / No	Si / Si	40	ADC, PWM
RASPBERRY PI 3 MODEL 3	4	No	No	microSD	No	Si / Si / Si	Si / Si	40	No
DRAGONBOARD 410c	2	No	Si (incorporado)	microSD	No	Si / Si / Si	Si / Si	12	No
BEAGLEBONE BLACK WIRELESS	1	No	Si (incorporado)	microSD	No	Si / Si / Si	Si / Si	66	12 bit ADC

Fuente (Kearney, 2016)

De acuerdo con este análisis, se puede concluir que el sistema embebido de UDOO x86 Ultra posee mayor frecuencia en su procesador e incluso una mayor capacidad en su memoria RAM, en

comparación con los demás sistemas: *ODROID XU4*, *Banana Pi M3*, *Radxa Rock*, *Orange Pi Plus 2*, *Raspberry Pi 3*, *Dragonboard 410c* y *Beaglebone Black Wireless*. Sin embargo, el precio de esta tarjeta es elevado con respecto a las demás, esto se debe a que posee un procesador de arquitectura superior y una cantidad de memoria RAM mejor que los otros sistemas, teniendo en cuenta lo anterior, se selecciona un embebido que tenga unas especificaciones técnicas similares, pero buscando un equilibrio entre rendimiento y costo.

Con base a estas condiciones se prefiere hacer uso del sistema embebido *ODROID XU4* (véase la Figura 2) por las siguientes razones:

- Posee un procesador ARM de ocho (8) núcleos y un procesador *GPU* también basado en ARM, con una frecuencia similar que el *UDOO*, además cuenta con una capacidad de memoria *RAM* de dos (2) GB.
- Es económico, compacto y posee los componentes necesarios para cumplir con los objetivos del proyecto, como lo son: puertos USB de alta velocidad de transmisión, 40 pines E/S para interacción con periféricos, entre otros (Hardkernel co. Ltd., 2015a).
- Al poseer puertos USB de alta velocidad (3.0), se puede percibir la calidad del video es mejor y de mejor rendimiento.
- Se encuentra entre uno de los cinco mejores sistemas embebidos que están en el mercado (Kearney, 2016).

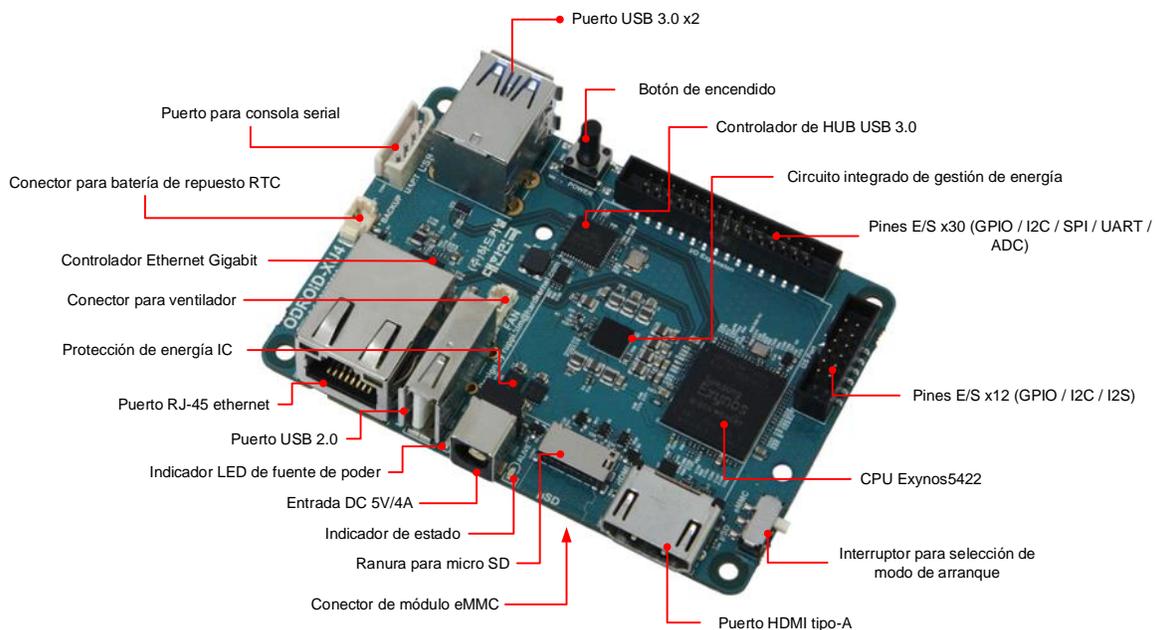


Figura 2. Esquema de componentes del sistema embebido: ODROID XU4.
Fuente (Roy, Bommanakanti, & Doiche, 2015)

1.2. Instalación del sistema operativo

Para este proceso es necesario contar con un soporte físico de arranque (disco duro), donde se almacenará el SO para su posterior ejecución. Luego de tener esto, se realiza el proceso de “*flashing*”, el cual consiste en grabar la imagen del sistema operativo en el soporte de arranque, usando otro sistema informático para efectuar dicha acción.

En el sistema embebido ODROID XU4 pueden usarse dos tipos de soportes físicos de arranque - la tarjeta micro SD (tarjeta de almacenamiento) o el módulo eMMC (*Embedded MultiMedia Card*, por sus siglas en inglés), en las cuales pueden almacenarse dos tipos de sistemas operativos diferentes, como lo son Linux y Android.

Para seleccionar el tipo de soporte de arranque, se tiene en cuenta la velocidad de lectura y escritura que posee estos, como lo indica la Figura 3, donde muestra las diferentes velocidades de lectura y escritura que posee una tarjeta micro SDHC clase 10 (*Secure Digital High Capacity*), una micro

SD-UHS1 (*SD Extreme Capacity* con estándar *UHS-I* y clase 10) y una eMMC versión 5.0. Los resultados fueron: - para la micro SD clase 10 su velocidad de escritura es de 8.5 MB/s (*Mega Bytes* por segundo) y una velocidad de lectura de 18.9 MB/s - para la SD-UHS1 sus velocidades fueron de 10.8 MB/s y 35.9 MB/s para lectura y escritura respectivamente y, - para la tercera opción se concluyó que sus velocidades de lectura y escritura fueron de 39.3 MB/s y 140 MB/s respectivamente. De acuerdo con estos resultados se puede señalar que la eMMC presenta mejores resultados que las SD-UHS1 y SDHC.

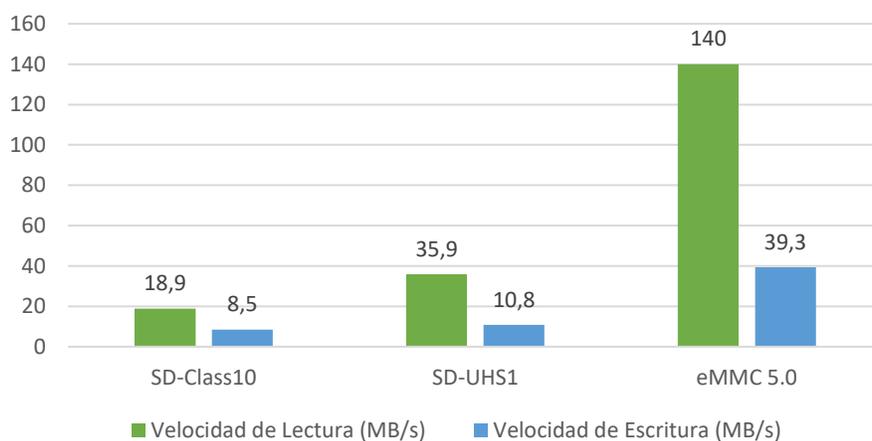


Figura 3. Comparación de velocidad de lectura y escritura de diferentes tipos de memoria.
Fuente (Hardkernel co. Ltd., 2015a)

1.2.1. Instalación de sistema operativo Android.

Para la instalación del sistema operativo Android se usa una tarjeta micro SD con capacidad de 16 GB, además se requerirá un adaptador SD para grabar la imagen del sistema operativo usando un ordenador de escritorio o portátil. A iniciar este proceso, primero se descarga la imagen de disco la cual contendrá el sistema operativo a instalar en el ODROID para ello, se ingresa a una de las direcciones web que ofrece el desarrollador para la descarga del SO Android¹; en esta página solo brindan desde la versión 1.4 hasta 5.0 del Android 4.4.4 “KitKat”.

¹ ODROID. *Software Release for Android*. URL [<http://bit.ly/1XwOatZ>]

Software Release for Android

Android 4.4.4 (v5.0)

- Update Images
- android-4.4.4-alpha-5.0-emmc_installer-odroidxu4-20170621.img.zip
(MD5SUM: e508dbc25d0b9ceb51ec0f7c4256571f)
- android-4.4.4-alpha-5.0-sd2emmc_installer-odroidxu4-20170621.img.zip
(MD5SUM: bc894fa466d36f63a47e234d04f5da5a)
- android-4.4.4-alpha-5.0-sd_installer-odroidxu4-20170621.img.zip
(MD5SUM: 72a77bfc572831c3ce0944ae2d133f6e)

Figura 4. Listado de sistemas operativos Android disponibles para ODROID.
Fuente autores.

Para este proceso se usa la versión más reciente la cual es 5.0, como lo indica la Figura 4, en este grupo se presentan tres opciones distintas para descargar la imagen de disco de acuerdo con el módulo que se vaya a utilizar (eMMC, SD to eMMC, SD). Para este caso se elige la tercera opción: “*android-4.4.4-alpha-5.0-sd_installer-odroidxu4-20170621.img.zip*”, ya que, como en su nombre lo indica, ese archivo está orientado para ser grabado en una tarjeta de memoria SD. Después de bajar el archivo de la página, procede el programa WinRAR para descomprimir esta descarga, luego monta el sistema operativo Android con el programa *Etcher* para realizar la escritura de la imagen en la SD. Este programa es sencillo de utilizar ya que permite al usuario, en unos simples pasos, grabar una imagen de disco dentro de cualquier unidad de almacenamiento extraíble.

En la Figura 5 se muestra la interfaz de usuario del programa, en este se muestran tres opciones:

- **Select image:** Se selecciona la imagen de disco a grabar.
- **Select drive:** Se selecciona la unidad de almacenamiento donde se grabará la imagen.
- **Flash:** Opción para iniciar el proceso de grabación y verificación.

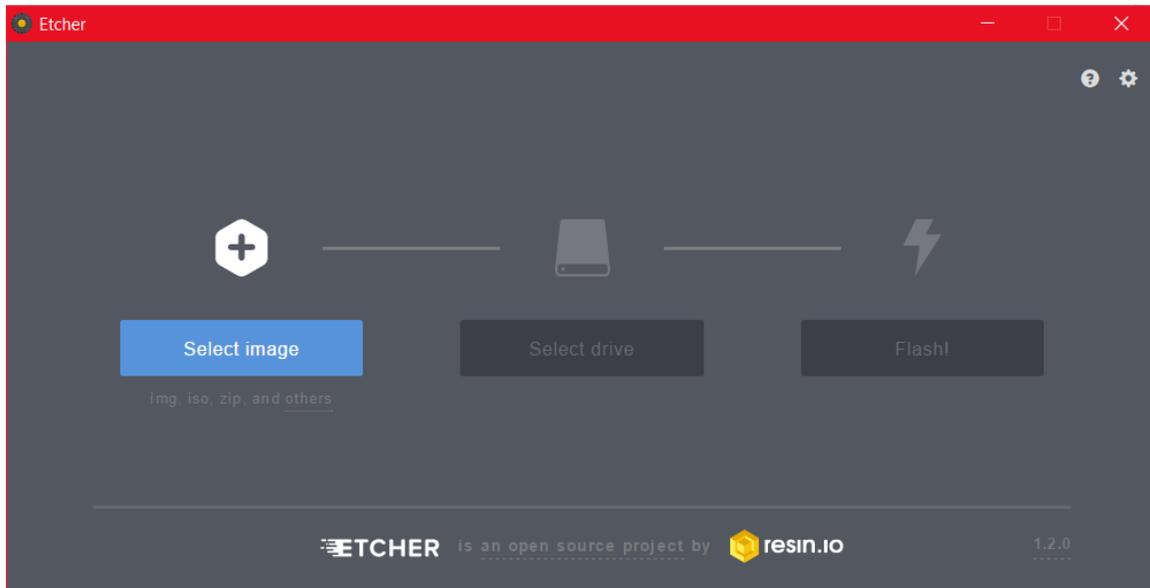


Figura 5. Interfaz de usuario del programa Etcher.

Fuente autores.

Al completar el proceso de grabación y verificación, se procede a extraer la memoria SD del ordenador e insertarla en el puerto micro SD del sistema embebido ubicado al costado derecho, como se muestra en la Figura 6:

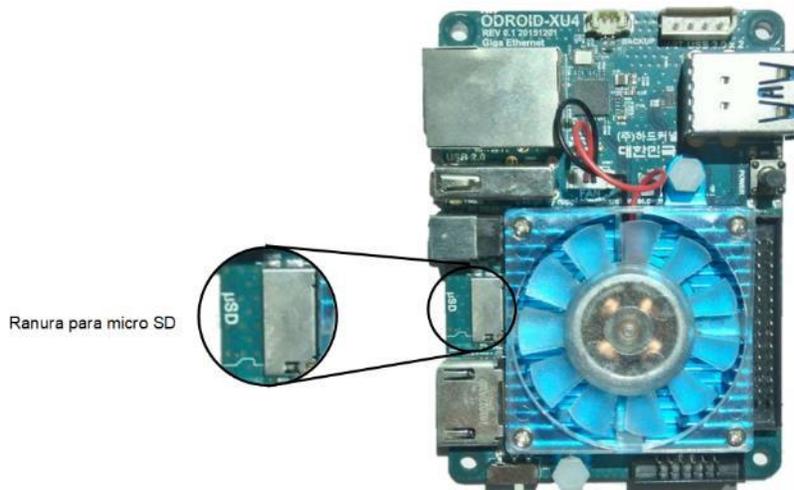


Figura 6. Parte superior del ODROID XU4 señalando el puerto para la tarjeta de memoria micro SD.

Fuente (Hardkernel co. Ltd., 2015a)

Con la tarjeta insertada en la ranura SD, se conecta el sistema embebido a una fuente de poder y se espera un intervalo de 10 a 20 minutos para que se complete el proceso de instalación del sistema operativo, durante este proceso el sistema no muestra ningún contenido gráfico a través del

monitor. Al completar la instalación, se muestra la página principal del sistema operativo, tal como se contempla en la siguiente Figura 7:

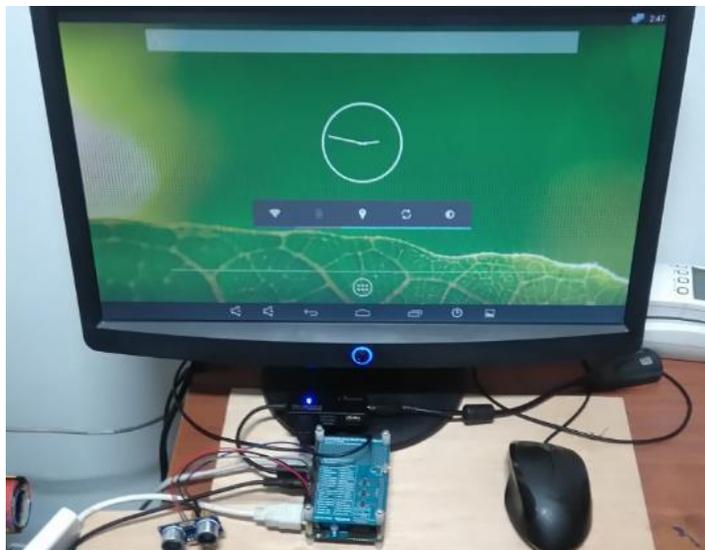


Figura 7. Fotografía donde se muestra la pantalla principal al iniciar el sistema operativo Android.
Fuente autores.

1.2.2. Instalación de sistema operativo Linux

Para la instalación de este sistema operativo se siguen los mismos pasos del proceso anterior, solo que esta vez se ingresa a la página de descarga de software Linux², aquí se descarga la versión de Linux Ubuntu 16.04. Al ingresar a esta opción, la página lo redireccionará a un directorio dentro del servidor de ODROID, donde se encuentran almacenados todas las versiones de Ubuntu, para la descarga de la imagen se selecciona el archivo con el nombre “*ubuntu-16.04-mate-odroid-u2u3-20160920.img.xz*”. En la Figura 8 se muestra la página web con la lista de SO Linux disponibles para descargar.

² ODROID. *Index of /ubuntu_16.04lts*. URL [https://odroid.in/ubuntu_16.04lts/]

Index of /ubuntu_16.04lts

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 old/	2017-11-26 21:04	-	
 ubuntu-16.04-mate-odroid-u2u3-20160920.img.md5sum	2016-09-21 07:52	77	
 ubuntu-16.04-mate-odroid-u2u3-20160920.img.xz	2016-09-21 08:08	1.1G	
 ubuntu-16.04-mate-odroid-u2u3-20160920.img.xz.md5sum	2016-09-21 07:56	80	
 ubuntu-16.04-mate-odroid-x-20160920.img.md5sum	2016-09-21 07:56	74	
 ubuntu-16.04-mate-odroid-x-20160920.img.xz	2016-09-21 08:27	1.1G	

Figura 8. Listado de sistemas operativos basados en Linux disponibles para el ODROID XU4.

Fuente autores.

Para su inicialización en el sistema operativo se usa una tarjeta de eMMC como soporte de arranque, para grabar la imagen anteriormente descargada se usa un módulo conversor de eMMC a micro SD, y luego de micro SD a puerto USB, como se muestra en la Figura 9:

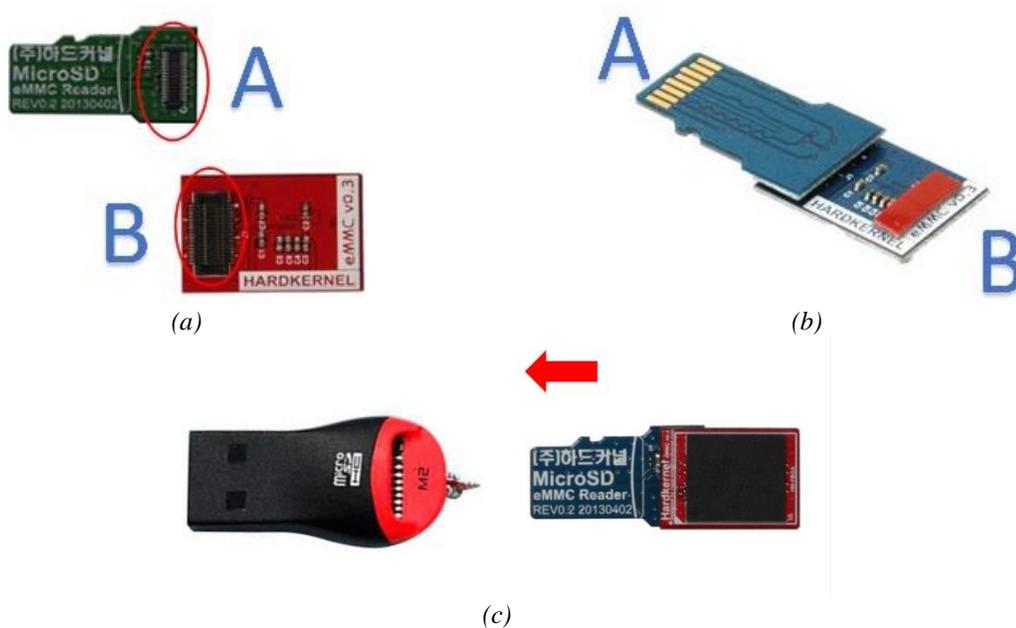
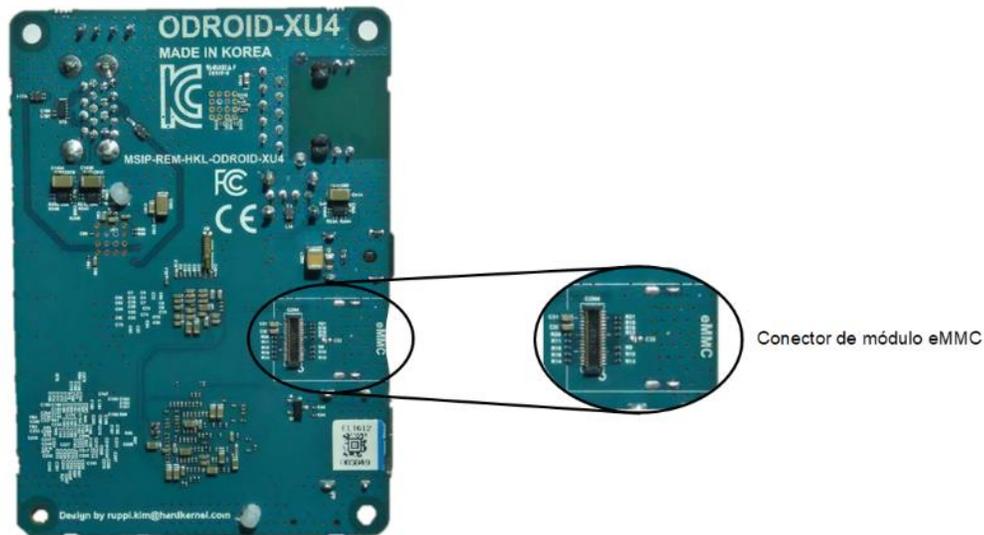


Figura 9.(a) Puertos de conexión entre adaptador micro SD y memoria eMMC. (b) Tarjeta eMMC conectada a adaptador micro SD. (c) Conexión de adaptador micro SD a puerto USB 2.0.

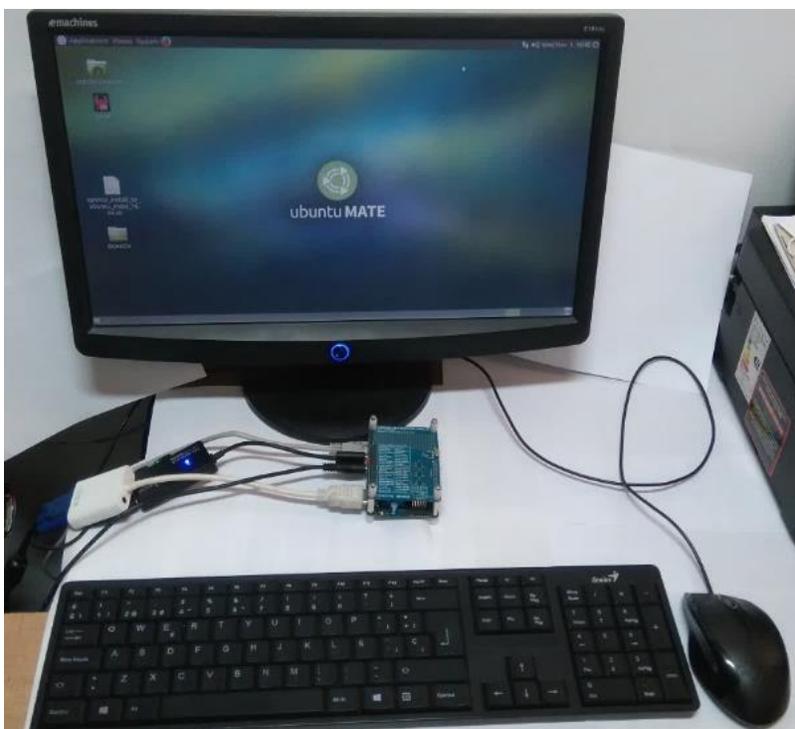
Fuente (Hardkernel co. Ltd., 2015a)

Después de que se culmina el proceso de descarga y grabación de la imagen con el sistema operativo en la memoria eMMC, se conecta el módulo en el puerto ubicado debajo del sistema embebido, tal cual como lo muestra la Figura 10:



*Figura 10. Parte posterior del ODROID XU4 señalando el puerto para la tarjeta de memoria eMMC.
Fuente (Hardkernel co. Ltd., 2015a)*

A continuación, se inicia el sistema embebido conectando este a una fuente de poder, se da espera para que el S.O. inicie y se acople al embebido, este proceso puede tardar entre 5 a 10 minutos. Al terminar se muestra la página de inicio de sesión del usuario, como se ilustra en la Figura 11.



*Figura 11. Pantalla principal del sistema operativo Ubuntu Mate 16.04.
Fuente autores.*

Se decide usar como sistema operativo principal el Ubuntu Mate; debido a la experiencia que se posee en el uso y configuración de este, además responde a las necesidades técnicas del proyecto — poder hacer uso de software de programación para el desarrollo prototipo, posee facilidad en la manipulación de sus periféricos y sus puertos de comunicación, además de la facilidad en la instalación y ejecución de programas en este entorno —.

1.3. Configuración de dispositivo para adquisición de imágenes

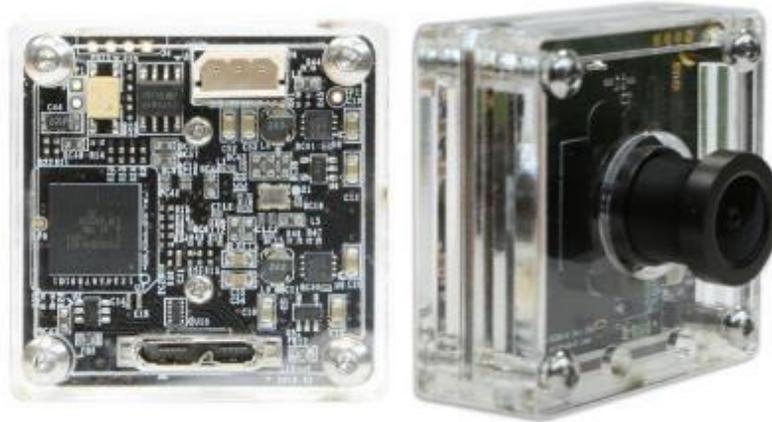
Para el proceso de adquisición de imágenes se usa un módulo de cámara suministrado por la misma compañía del sistema embebido, esto con el fin de que no se presentase algún tipo de incompatibilidad entre la cámara y el sistema embebido. Este módulo es conocido como *oCam* (véase Figura 12), una cámara de 5 megapíxeles que soporta tecnología Plug-and-Play mediante el protocolo UVC (USB Video Class, por sus siglas en inglés), usa un lente estándar de tipo M12 con una distancia focal de 3.6 milímetros. Una de las ventajas de este módulo es que al utilizar

tecnología USB de versión 3.0, permite un acceso directo a la memoria principal del sistema embebido para la escritura de datos, reduciendo significativamente la carga de trabajo en la CPU de este (Hardkernel co. Ltd., 2016). A continuación, se describe los detalles técnicos del módulo en la Tabla 3:

Tabla 3. Especificaciones técnicas del módulo de cámara oCam.

COMPONENTES	DESCRIPCIÓN
Sensor:	OmniVision OV5640 CMOS image sensor
Lente:	Lente estándar M12 con longitud focal de 3.6 mm
Campo de visión (FOV):	65 grados
Tamaño del sensor de imagen:	0.25 pulg. (3673.6 μm x 2738.4 μm)
Tamaño del pixel:	1.4 μm x 1.4 μm
Obturador:	Obturador enrollable eléctrico
Interfaz:	USB 3.0 Super-Speed
SO compatible:	Windows7/8/10, Linux, Plug-and-Play por protocolo UVC (USB Video Class).
Potencia:	USB BUS Power (Rating: 5V/280mA)
Temperatura de operación:	0°C ~ + 70°C
Control de cámara:	Brillo, contraste, matiz, saturación, balance de blancos.
Cuadros por segundo:	YUV: 2592x1944@7.5fps, 1920x1080@15fps, 1280x720@30fps, 640x480@120fps, 320x240@120fps MJPEG: 1920x1080@30fps, 1280x720@45fps, 640x480@30fps
Profundidad:	30 – 35 mm (Longitud variable para enfoque manual)
Peso:	35 gramos aprox. (Incluyendo marco)

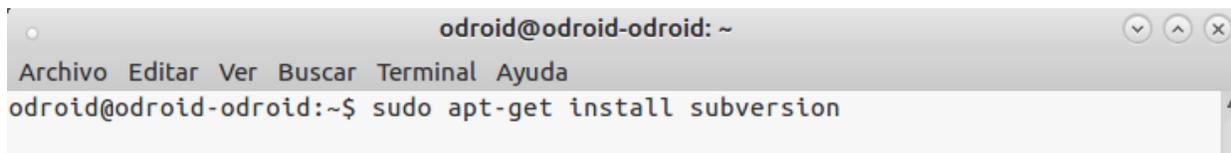
Fuente (Hardkernel co. Ltd., 2016)



*Figura 12. Módulo oCam 5MP.
Fuente (Hardkernel co. Ltd., 2016)*

Para poder usar este módulo en el sistema operativo Linux debe antes instalarse el software correspondiente, el cual se encuentra disponible para la descarga en la plataforma de GitHub³. Para la instalación de este primero se debe instalar el programa “*subversión*”, el cual permitirá la exportación del programa desde el repositorio web, para ello se ingresa a la terminal de Linux y se ingresa el siguiente comando, como se muestra la Figura 13:

```
$ sudo apt-get install subversion
```



*Figura 13. Instalación de programa "subversion" desde la terminal de Linux.
Fuente autores.*

Al ingresar el comando, el sistema comenzará a obtener todos los paquetes necesarios para su instalación, luego el mismo preguntará si se desea continuar con la instalación, aquí se escribe la letra “Y” indicando que sí:

Luego de instalado este programa, se ingresa a la carpeta de descargas:

³ GitHub. *oCam Viewer Linux*. URL
[https://github.com/withrobot/oCam/tree/master/Software/oCam_viewer_Linux]

```
$ cd Downloads
```

A continuación, se ingresa el siguiente comando para la exportación del software:

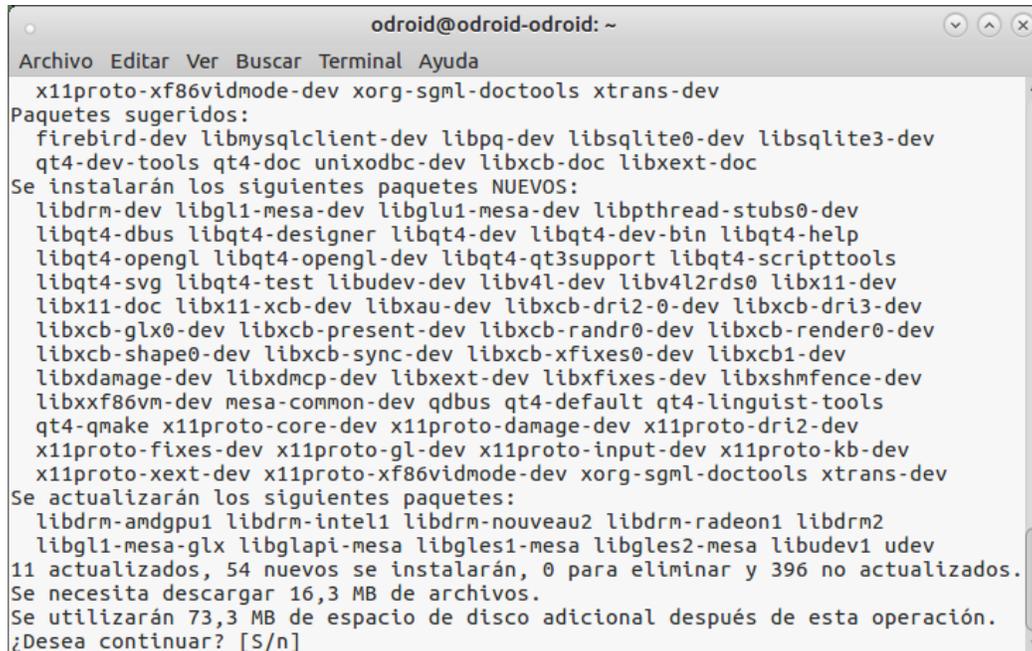
```
$ svn export
```

```
https://github.com/withrobot/oCam/trunk/Software/oCam_viewer_Linux
```

Aquí el sistema comenzará la descarga del software del directorio de GitHub, luego se procede a instalar los paquetes dependientes del software desde el *Linux Package Manager*, también desde la terminal:

```
$ sudo apt-get install qt4-default libv4l-dev libudev-dev
```

Nuevamente, el sistema comienza a adquirir todos los paquetes necesarios para la instalación, este pedirá una confirmación, se selecciona la opción “Y”, para continuar con el proceso, tal cual como lo muestra la Figura 14:



```
odroid@odroid-odroid: ~
Archivo Editar Ver Buscar Terminal Ayuda
x11proto-xf86vidmode-dev xorg-sgml-doctools xtrans-dev
Paquetes sugeridos:
firebird-dev libmysqlclient-dev libpq-dev libsqlite0-dev libsqlite3-dev
qt4-dev-tools qt4-doc unixodbc-dev libxcb-doc libxext-doc
Se instalarán los siguientes paquetes NUEVOS:
libdrm-dev libgl1-mesa-dev libglu1-mesa-dev libpthread-stubs0-dev
libqt4-dbus libqt4-designer libqt4-dev libqt4-dev-bin libqt4-help
libqt4-opengl libqt4-opengl-dev libqt4-qt3support libqt4-scripttools
libqt4-svg libqt4-test libudev-dev libv4l-dev libv4l2rds0 libx11-dev
libx11-doc libx11-xcb-dev libxau-dev libxcb-dri2-0-dev libxcb-dri3-dev
libxcb-glx0-dev libxcb-present-dev libxcb-randr0-dev libxcb-render0-dev
libxcb-shape0-dev libxcb-sync-dev libxcb-xfixes0-dev libxcb1-dev
libxdamage-dev libxdmcp-dev libxext-dev libxfixes-dev libxshmfence-dev
libxxf86vm-dev mesa-common-dev qdbus qt4-default qt4-linguist-tools
qt4-qmake x11proto-core-dev x11proto-damage-dev x11proto-dri2-dev
x11proto-fixes-dev x11proto-glx-dev x11proto-input-dev x11proto-kb-dev
x11proto-xext-dev x11proto-xf86vidmode-dev xorg-sgml-doctools xtrans-dev
Se actualizarán los siguientes paquetes:
libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2 libdrm-radeon1 libdrm2
libgl1-mesa-glx libglapi-mesa libgles1-mesa libgles2-mesa libudev1 udev
11 actualizados, 54 nuevos se instalarán, 0 para eliminar y 396 no actualizados.
Se necesita descargar 16,3 MB de archivos.
Se utilizarán 73,3 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Figura 14. Confirmación para la instalación de programas dependientes.

Fuente autores.

El paquete *qt4-default* refiere al software para ejecutar programas que usen interfaz gráfica de usuario, *libv4l-dev* a la API (Interfaz de programación de aplicaciones) para la captura de video en

Linux y *libudev-dev* a la API para la inspección de los dispositivos locales, dando de manera robusta y de distribución independiente acceder a los dispositivos desde su nodo raíz.

Para la compilación del programa en el sistema operativo, por medio de la terminal de Linux, se ingresa al directorio donde se encuentra ubicado este mediante el comando:

```
$ cd ./oCam_viewer_Linux
```

Una vez ingresado se crea una carpeta llamada *build* con el uso del comando *mkdir* de la siguiente forma:

```
$ mkdir build
```

Luego se accede a esta carpeta y se ejecuta la herramienta *qmake* para crear el *makefile* que será utilizado en la construcción del programa a partir de su código fuente:

```
$ cd ./build  
$ qmake ..
```

Para finalizar, se inicia el programa *make* para crear e instalar el ejecutable del software descargado usando el *makefile* anteriormente creado, esto mediante el comando:

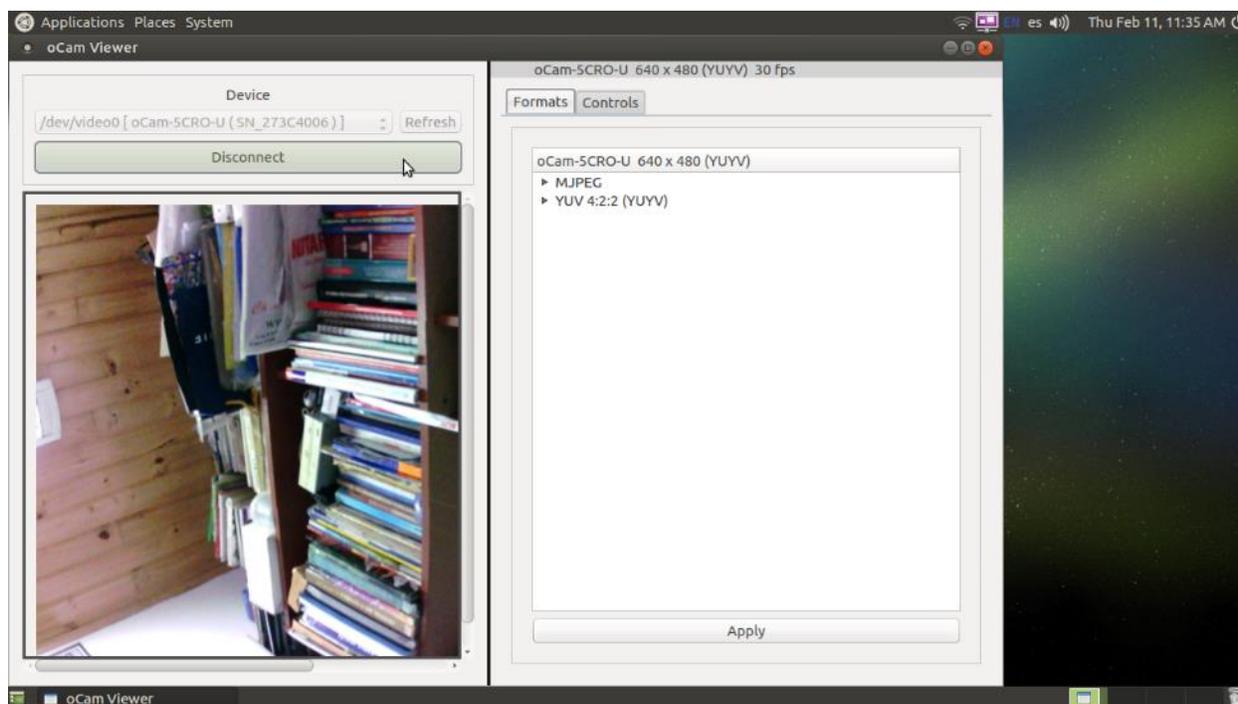
```
$ make release
```

Para la ejecución del software instalado, se usa la terminal de Linux ingresando a la carpeta donde este fue instalado y luego se ejecuta el comando que se describe a continuación:

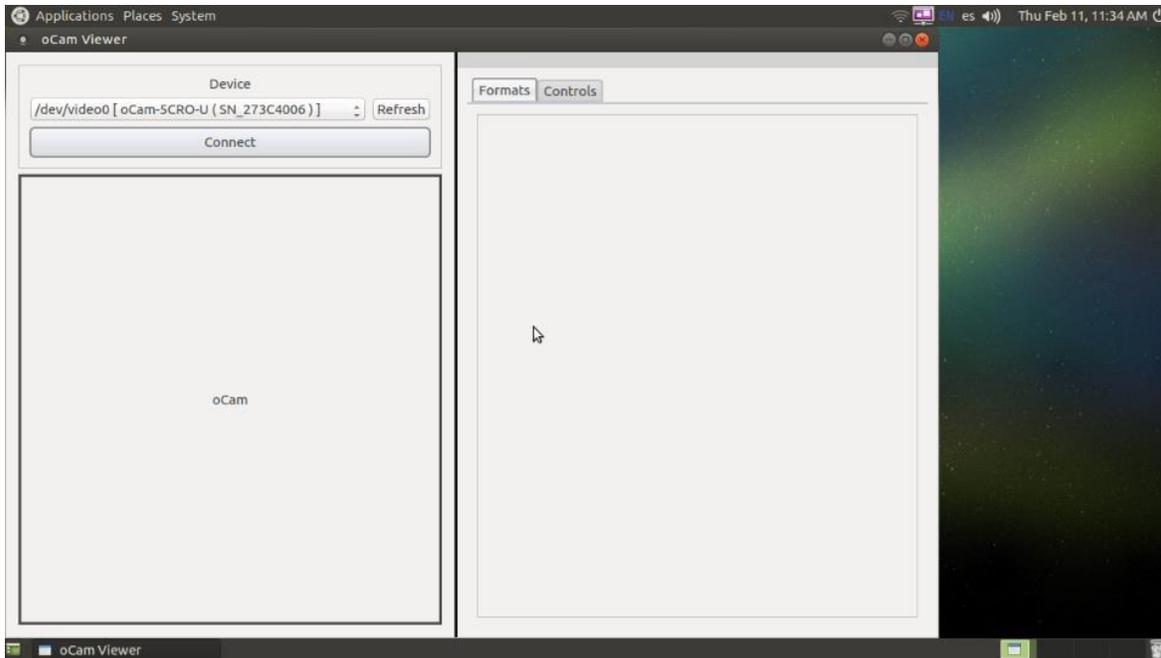
```
$ ./oCam-viewer
```

Dentro de la interfaz gráfica del programa primero se selecciona el dispositivo que se desea visualizar, una vez elegido se oprime la opción *Conectar* para vincular el módulo con el programa, como se ve en la Figura 16(a), este inmediatamente mostrará la vista previa proveniente del

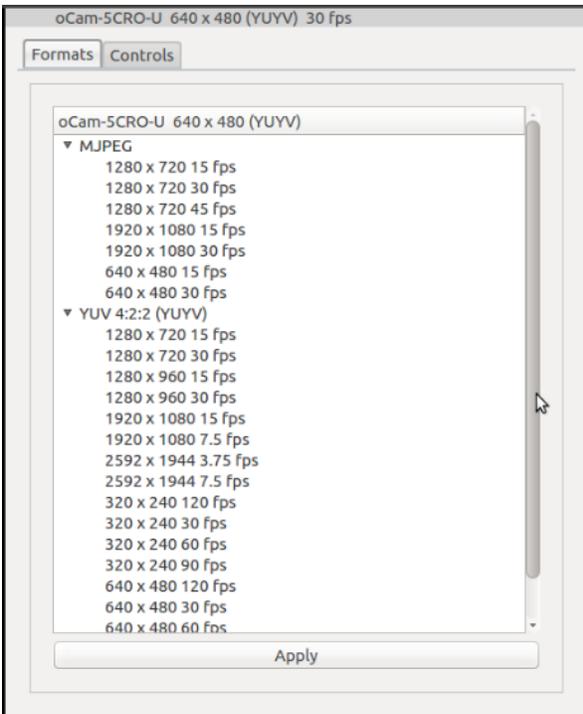
módulo, como se observa en la Figura 15, además se pueden encontrar diferentes parámetros de configuración disponibles en el programa como lo son: en una pestaña se encuentran los formatos de entrada (MJPEG o YUV) y una lista de varias resoluciones compatibles con el módulo además de los cuadros por segundo que puede visualizar, en otra pestaña se encuentra los controles básicos para el ajuste de las propiedades de la imagen, como lo es brillo, contraste, exposición, tono y saturación, estas opciones se muestran en las Figura 16(b) y (c):



*Figura 15. Interfaz gráfica del programa oCam Viewer.
Fuente autores.*



(a)



(b)



(c)

Figura 16.(a) Vista previa proveniente del módulo de cámara una vez conectado al programa. (b) Lista de formatos y resoluciones compatibles de la cámara. (c) Controles para ajustar las propiedades de la imagen adquirida por la cámara.

Fuente autores.

1.4. Adquisición de la imagen en C++

Para la adquisición de la imagen se codifica un programa en lenguaje de C++ implementando una de las librerías ofrecidas por el software libre OpenCV además de una librería para la lectura y escritura de los pines del sistema embebido ODROID, este último con el fin de adaptar dos actuadores externos: un botón que cumplirá con la función de apagar el sistema y un sensor que medirá la distancia adecuada para que el módulo de cámara obtenga todos los objetos con una misma proporción o tamaño, esto ayudará en el pre-procesamiento y en los datos de entrada para el proceso de clasificación. Adicional a esto se implementará un algoritmo para la reproducción de audio con el fin de orientar al usuario de la distancia entre el dispositivo y el objeto a reconocer.

Dentro del programa se crean cuatro funciones principales las cuales permitirán una buena ejecución dentro del *Main Program*:

- Configuración de pines
- Lectura del sensor de distancia
- Reproducción de audio
- Captura de fotograma

1.4.1. Configuración de pines

Esta función se realiza la habilitación de los pines que serán usados por el programa para leer y modificar las entradas de los puertos GPIO en el sistema embebido.

Para lograr la instalación del cableado del sensor y el obturador fue necesario la adquisición de una placa de circuito modular, ya que los pines que vienen instalados en la placa del sistema embebido son incompatibles físicamente con los pines de los circuitos. Otra razón para la

adquisición de esta es para prevenir algún tipo de corto circuito que se pueda producir entre las conexiones entre el embebido y los sensores.

A continuación, se muestra en la Figura 17 como se ve este módulo instalado en el sistema embebido:



Figura 17. Módulo Shifter Shield acoplado al ODROID XU4.
Fuente (Hardkernel co. Ltd., 2015a)

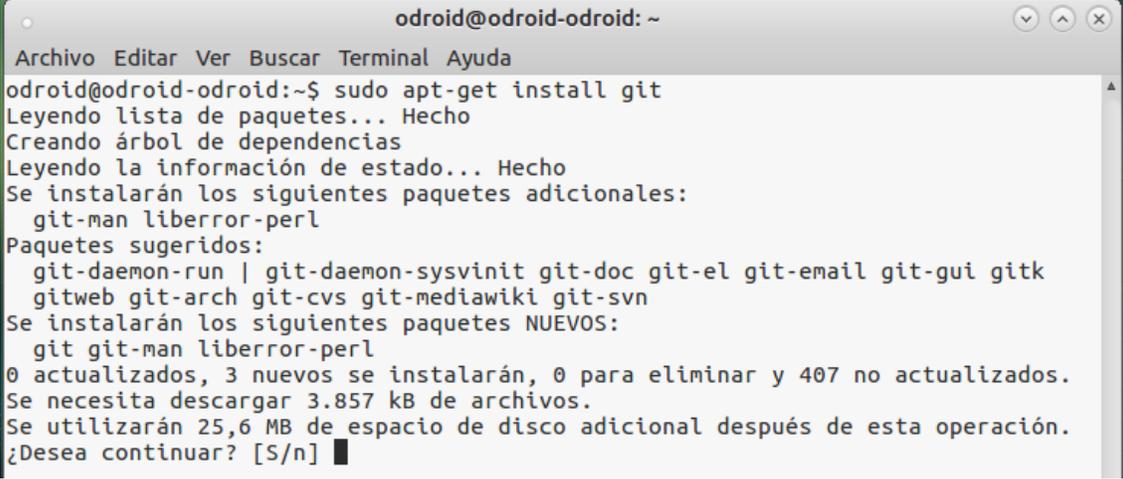
1.4.1.1. Instalación de librería para acceso a los pines

Para manipular los pines del sistema embebido debe instalarse la librería *WiringPi*, esta permite el control de los GPIO desde el software desarrollador. La librería *WiringPi* original fue, en primer lugar, diseñada para usarse en microprocesadores BCM2835 incorporados a las Raspberry PI, fue escrita en lenguaje C y licenciada bajo la GNU LGPLv3 para usarse en lenguajes como C, C++, Python, entre otras. Hardkernel, proveedor de la ODROID XU4, ofrece una librería derivada de la original para ser usada en dispositivos ODROID.

Para la instalación de *WiringPi*, se inicia la terminal Linux y se introducen los siguientes comandos:

Primero se debe instalar el programa *GIT* para clonar la librería disponible el repositorio en la memoria del sistema embebido, en el proceso el sistema pedirá confirmación para proceder con la instalación, como se indica en la Figura 18:

```
$ sudo apt-get install git
```



```
odroid@odroid-odroid: ~
Archivo Editar Ver Buscar Terminal Ayuda
odroid@odroid-odroid:~$ sudo apt-get install git
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  git-man liberror-perl
Paquetes sugeridos:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-arch git-cvs git-mediawiki git-svn
Se instalarán los siguientes paquetes NUEVOS:
  git git-man liberror-perl
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 407 no actualizados.
Se necesita descargar 3.857 kB de archivos.
Se utilizarán 25,6 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

*Figura 18. Confirmación para instalación del software.
Fuente autores*

Terminado el proceso de instalación, se procede a clonar la librería, esto mediante el siguiente comando:

```
$ git clone https://github.com/hardkernel/wiringPi
```

Luego se inicia el proceso de compilación de la librería, el cual dura entre 1 a 3 minutos, para ello se usan los comandos que se muestran a continuación:

Comando para ingresar a la carpeta donde fue clonado el archivo:

```
$ cd wiringPi
```

Comando para iniciar el proceso de compilación:

```
$ sudo ./build
```

Cuando termine este proceso, la librería estará lista para ser utilizada.

Para la adecuación de un botón de apagado seguro del sistema, se sitúa un pulsador el cual va conectado al pin 13 (GPIO2 según tabla de pines) del embebido de la siguiente forma:

Se tiene un interruptor tipo pulsador con dos pines, uno de estos pines se conecta al pin 1 del embebido, el cual proveerá 5 voltios al circuito, el otro pin se conecta al pin 13 (GPIO2) el cual obtendrá la señal del pulsador, además se conecta a este pin una resistencia de 220 ohmios, el propósito de este es contrarrestar alguna corriente no deseada que pueda provocar una falla eléctrica en el sistema embebido, luego del otro extremo de la resistencia se conecta al pin Tierra del embebido (pin 2), y así completar el circuito. En la Figura 19 se ilustra dicho circuito.

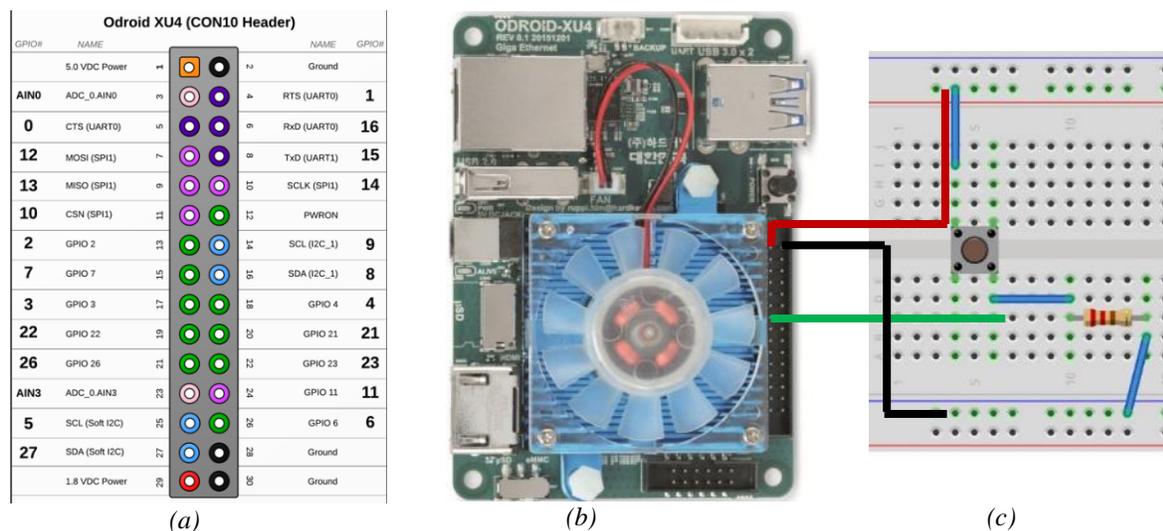


Figura 19. (a) Diagrama de pines del ODROID XU4. (b) Sistema embebido ODROID XU4. (c) Circuito del obturador para la adquisición de la imagen.

Fuentes: (a) Pi4j.com, (b) (Hardkernel co. Ltd., 2015a), (c) autores.

Después de esto se desarrolla un algoritmo en lenguaje C++ que permita leer la señal entrante del pin 13 y luego ejecutar una acción específica, en este caso que se ejecute una función para el apagado seguro del sistema embebido, dentro de dicho algoritmo se importa la librería que permite la manipulación de los pines en el embebido, además de la librería donde se encuentra el comando del sistema para apagar el equipo.

1.4.2. Lectura de sensor de distancia

Para el procesamiento de la imagen se debe tomar en cuenta que los objetos dentro de esta tengan aproximadamente el mismo tamaño, esto se puede conocer por medio de la distancia, en la cual se captura el fotograma, para ello se instala un sensor ultrasónico el cual indicará si la imagen a procesar se encuentra a la distancia adecuada, además minimiza los errores en el programa de reconocimiento.

El sensor para utilizar es el HC-SR04, este es muy común en aplicaciones para medir la distancia de un objeto ya que es fácil de implementar y porque su medición es aproximada a la real, el funcionamiento del sensor consiste en enviar un pulso ultrasónico a través del transductor, luego se espera a que este rebote sobre cualquier objeto o superficie y sea captado por el receptor.

A continuación, se describe las características técnicas del sensor:



*Figura 20. Sensor ultrasónico HC-SR04.
Fuente (ITEAD Intelligent Systems Co.Ltd., 2010)*

El módulo se compone de un transmisor ultrasónico, su correspondiente receptor y el circuito de control. Como se muestra en la Figura 20, este sensor posee 4 pines:

VCC: Alimentación +5 Voltios (4.5V min – 5.5V max)

TRIG: Disparador (Trigger) de entrada (input) del sensor (TTL)

ECHO: Señal eco de salida (output) del sensor (TTL)

GND: Pin tierra

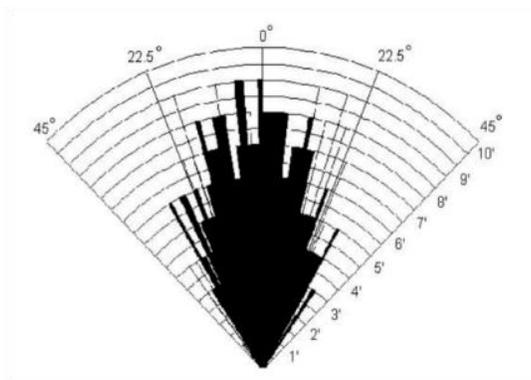


Figura 21. Ángulo de medición y distancia medida por el sensor.
Fuente (ITEAD Intelligent Systems Co.Ltd., 2010)

Como lo demuestra la Figura 21, el sensor es capaz de medir distancias no mayores a 4 metros desde su punto inicial y detectar dentro de un ángulo de apertura efectivo de 15° en cada lado desde su origen. A continuación, en la Tabla 4 se muestran otros parámetros de funcionamiento del sensor:

Tabla 4. Parámetros de funcionamiento e información adicional.

PARÁMETRO	VALOR
Corriente en estado de reposo:	< 2 miliamperios (mA)
Corriente de trabajo:	15 miliamperios (mA)
Ángulo de medición:	30°
Ángulo de medición efectivo:	15°
Detección:	De 2 cm a 400 cm o 1 a 13 pies
Precisión:	±1 cm
Dimensiones:	45 mm x 20 mm x 15 mm
Frecuencia de trabajo:	40 KHz

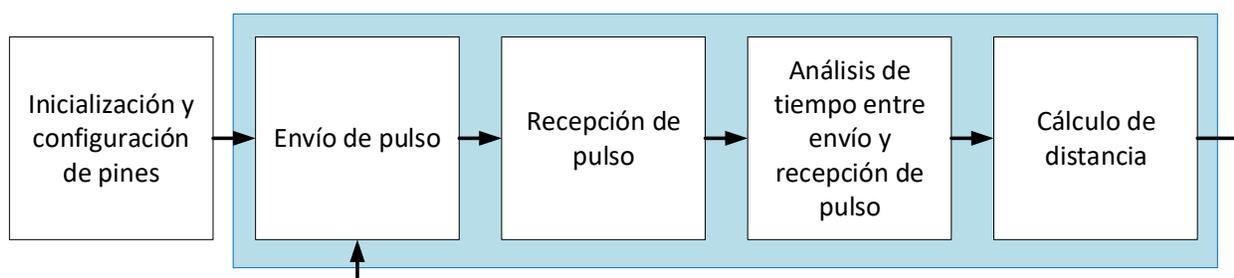
Fuente (ITEAD Intelligent Systems Co.Ltd., 2010)

Como se describió anteriormente, el sensor envía y recibe un pulso ultrasónico, en el proceso se toma el tiempo que esta demora en ir y volver al sensor, y de este se calcula la distancia aproximada, este valor puede variar entre más o menos tres (3) centímetros de la medida actual.

De acuerdo con la hoja técnica del sensor para medir con el dispositivo primero se envía un pulso corto de 10 microsegundos (uS) en la entrada del disparador o *Trigger* para comenzar la medición, el módulo inmediatamente enviará una ráfaga de 8 ciclos ultrasónicos de 40 KHz, al mismo tiempo se configura el pin *Echo* en estado alto y se dará inicio al conteo de tiempo, esta se mantendrá hasta recibir la señal reflejada por el obstáculo la cual pondrá su pin *Echo* en bajo y terminará el conteo, con este intervalo de tiempo recorrido se obtiene la distancia recorrida usando la fórmula que se describe a continuación (ITEAD Intelligent Systems Co.Ltd., 2010).

$$distancia [cm] = \frac{tiempo\ recorrido [uS]}{58} \quad (1)$$

Para la implementación del sensor en el algoritmo de programación, se debe tomar dichos parámetros de configuración, y la Ecuación 1 para la conversión en valores de distancia, todo esto se ingresa a este código en el orden descrito en el diagrama de bloques de la Figura 22 :



*Figura 22. Diagrama de bloques correspondiente al programa de medición de distancia usando el sensor ultrasónico.
Fuente autores.*

Primero se inicializan los pines mediante la inclusión de la librería *WiringPi* dentro del programa, luego, dentro de una función, se configuran los pines de entrada y salida para el sensor ultrasónico, para este caso se configura el pin 24 del sistema embebido como salida, este va conectado al pin *Trigger* y el pin 29 como entrada donde se conecta el pin *Echo*, todos directamente al sensor. Ya configurados los pines, se crea una nueva función para operar el sensor. La primera parte consiste

en enviar el pulso desde el pin de salida 24, durante 10 microsegundos. Luego se crean dos funciones: una donde detecte cuando el pin 29 (*Echo*) se encuentre en estado alto e inicie un cronómetro; y otra que al estar el pin 29 en estado bajo, detenga dicho cronómetro. Con estas funciones se permite hallar el tiempo recorrido entre el sensor y el obstáculo, y consecuentemente hallar la distancia entre estos.

Con el valor de tiempo anteriormente hallado, mediante la ecuación 1, se halla el valor de la distancia entre el sensor y el objeto, que más tarde servirá para determinar si esta distancia es la recomendada o no para comenzar a hacer el procesamiento de imágenes.

Para la instalación del sensor en el sistema embebido se tomó en cuenta el circuito electrónico descrito en la Figura 23:

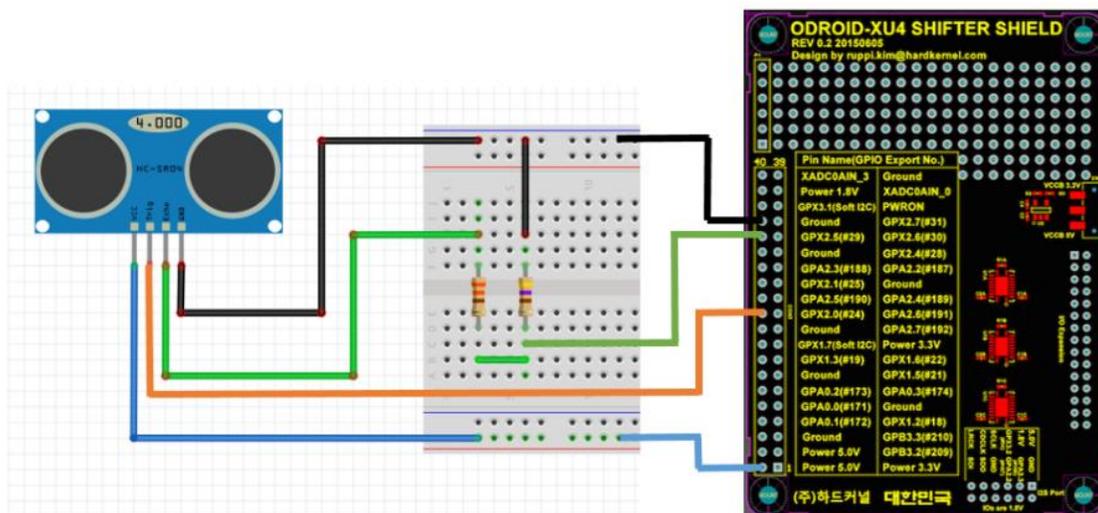


Figura 23. Conexión del sensor ultrasónico al Shifter Shield del sistema embebido ODROID.
Fuente autores.

En la Figura 24 se muestra una comparación entre los resultados obtenidos de la medición hecha por el sensor ultrasónico, y la distancia medida por un flexómetro en diferentes puntos. Como se ve en las figuras, el valor medido entre el sensor y el instrumento de medición varía entre 1 a 2 centímetros (cms) en su precisión, lo cual lo hace apto para ser usado como referencia.

```

Terminal
File Edit View Search Te
Distancia: 34.40 cm
Distancia: 33.97 cm
Distancia: 33.95 cm
Distancia: 33.95 cm
Distancia: 34.38 cm
Distancia: 33.98 cm
Distancia: 33.95 cm
Distancia: 33.98 cm
Distancia: 33.95 cm
Distancia: 33.93 cm
Distancia: 33.97 cm
Distancia: 33.97 cm
Distancia: 33.97 cm
Distancia: 33.93 cm

```

(a)



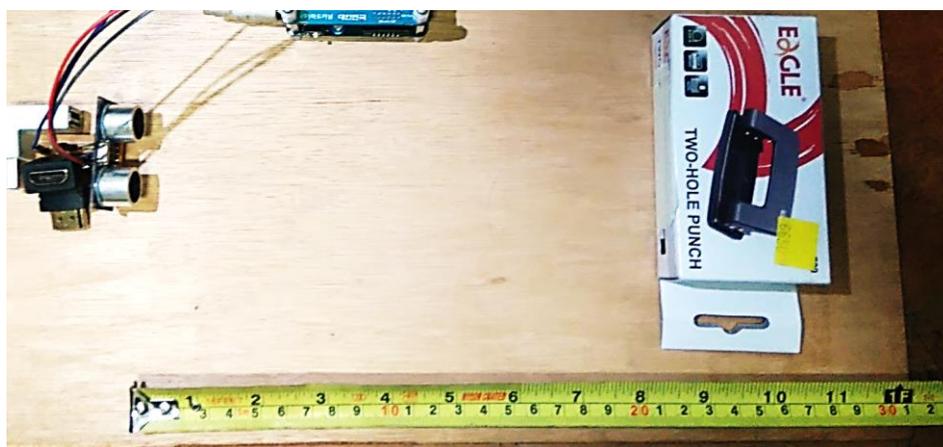
(b)

```

Terminal
File Edit View Search Te
Distancia: 23.55 cm
Distancia: 23.97 cm
Distancia: 23.98 cm
Distancia: 23.97 cm
Distancia: 23.98 cm
Distancia: 23.97 cm
Distancia: 23.57 cm
Distancia: 24.00 cm
Distancia: 24.00 cm
Distancia: 23.98 cm
Distancia: 23.98 cm
Distancia: 23.57 cm
Distancia: 1.66 cm
Distancia: 23.55 cm
Distancia: 28.86 cm
Distancia: 23.97 cm

```

(c)



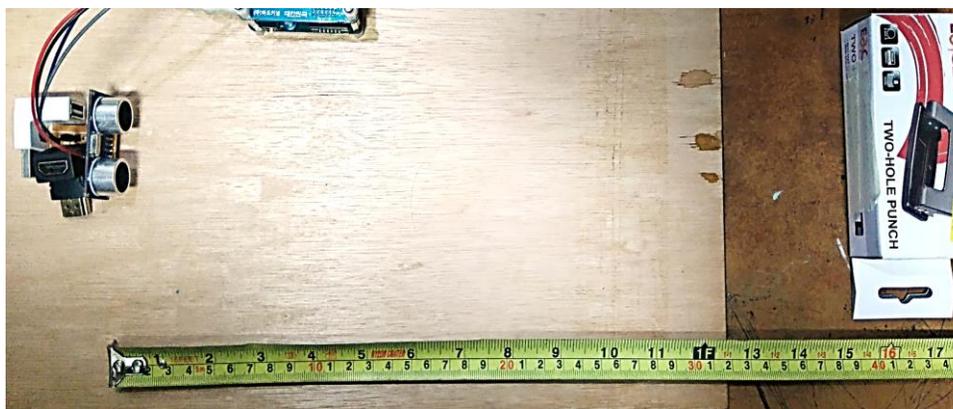
(d)

```

Terminal
File Edit View Search Te
Distancia: 41.12 cm
Distancia: 41.60 cm
Distancia: 41.62 cm
Distancia: 41.12 cm
Distancia: 41.59 cm
Distancia: 41.59 cm
Distancia: 41.17 cm
Distancia: 41.16 cm
Distancia: 41.60 cm
Distancia: 41.57 cm
Distancia: 41.59 cm
Distancia: 41.16 cm
Distancia: 40.74 cm
Distancia: 40.74 cm

```

(e)



(f)

Figura 24. (a) Resultado de medición hecha por el sensor desde un punto A y mostrado en la terminal Linux. (b) Medición hecha con un flexómetro desde un punto A. (c) Resultado de medición hecha por el sensor desde un punto B y mostrado en la terminal Linux. (d) Medición hecha con un flexómetro desde un punto B, más cercano al sensor. (e) Resultado de medición hecha por el sensor desde un punto C y mostrado en la terminal Linux. (f) Medición hecha con un flexómetro desde un punto C, más lejano al sensor.

Fuente autores.

1.4.3. Reproducción de audio

Debido a que el prototipo va desarrollado hacia las personas con discapacidad visual, uno de los medios para transmitir las acciones del dispositivo es por medio del audio, por lo tanto, se hace uso de un módulo transmisor de audio a través del protocolo USB en el sistema embebido, el adaptador usado es elaborado por la misma compañía del sistema embebido —Hardkernel—, a continuación, se describe este módulo:

El adaptador de audio USB (véase Figura 25) está conformado por un microchip modelo CM108AH desarrollado por la compañía C-Media Electronics, el elaborador de este adaptador (Hardkernel) afirma que este chip es “altamente integrado para una solución de audio por protocolo USB”, ya que todos los módulos análogos esenciales se encuentran integrados en este chip, incluyendo: un conversor digital – analógico dual, un amplificador en la salida, un conversor analógico – digital, un amplificador en el micrófono, un PLL (*Phase-Lock Loop*, por sus siglas en inglés), un regulador de voltaje y un transceptor USB (Hardkernel co. Ltd., 2015b).



Figura 25. (a) Módulo adaptador de audio por protocolo USB. (b) Componentes principales presentes en el adaptador de audio.

Fuente (Hardkernel co. Ltd., 2015b)

A continuación, se muestran las características pertenecientes a este módulo:

- Conexión USB 2.0 de alta velocidad (12 Mbps)
- *USB Audio Device Class Specification 1.0*
- Alimentación por USB
- Conectores:
 - *USB Type-A*
 - Salida estéreo de 3.5 mm
 - Entrada para microfono mono de 3.5 mm
- Dimensiones: 60 x 30 x 15 mm aprox.
- Peso: 33 g.

Para el algoritmo, la función toma el valor dado por el sensor de distancia y pasa por una serie de condiciones, y de acuerdo a la condición la función reproduce un sonido alertando al usuario si se encuentra en una posición adecuada para comenzar a procesar la imagen, en la Figura 26 se explica este algoritmo.

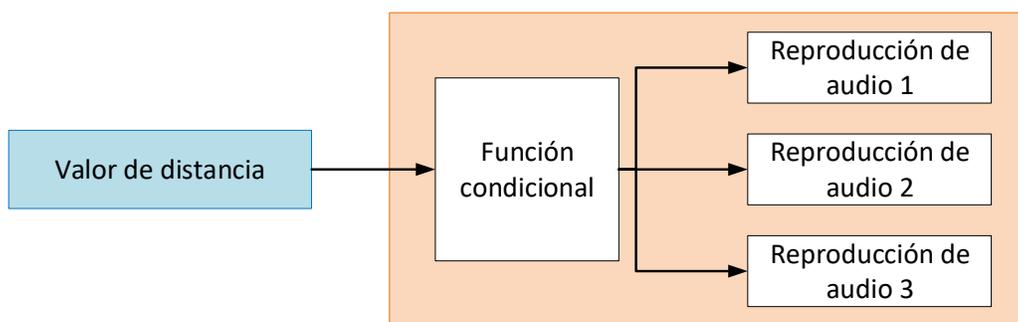


Figura 26. Diagrama de bloques de la función de reproducción de sonidos.
Fuentes autores.

A continuación, se muestran los resultados de la prueba realizada para confirmar el funcionamiento de este, donde se configura que, de acuerdo con el valor de la distancia, reproduzca un archivo de audio distinto, para ello se programa la función de acuerdo con la Ecuación 2:

$$distancia = \begin{cases} \text{Audio 1(Alejarse)} & \text{si } distancia \leq 30 \\ \text{Audio 2(Detengase)} & \text{si } 30 \leq distancia \leq 35 \\ \text{Audio 3(Acecarsse)} & \text{si } distancia \geq 30 \end{cases} \quad (2)$$

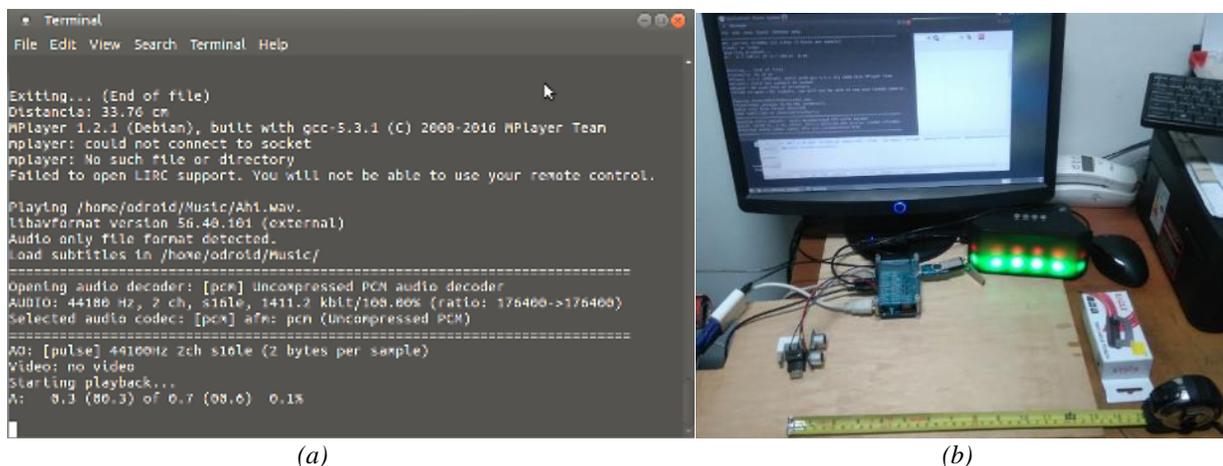


Figura 27. (a) Terminal Linux donde muestra distancia actual y reproducción de audio indicando que la distancia es correcta. (b) Medición entre sensor y un objeto en un punto neutral.
Fuente autores.

En la Figura 27(a) se muestra la terminal de comandos de Linux, en ella se indica la distancia entre el sensor y el objeto, que para este caso es de 33.76 centímetros (cms) aproximadamente; además está la información referente a la reproducción del archivo de audio correspondiente para este valor. En la Figura 27(b) se muestra el sensor y el objeto (caja de cartón) en un punto neutral o en el origen, en este el sistema emite la alerta sonora indicando que se encuentra en la posición correcta.

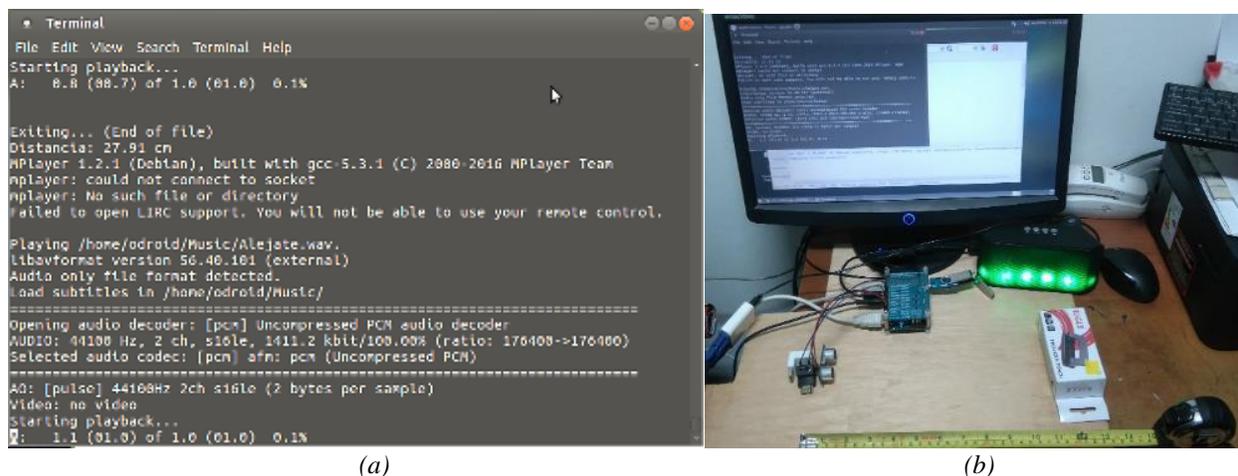


Figura 28. (a) Terminal Linux donde muestra distancia actual y reproducción de audio indicando que debe alejarse. (b) Medición entre sensor y un objeto en un punto más cercano a este.
Fuente autores.

Luego, en la Figura 28(a) se ilustra nuevamente la terminal de Linux, pero esta vez indicando una distancia diferente (27.91 cms), y consecuentemente, el sistema reproduce un archivo de audio totalmente distinto, indicando al usuario que debe alejarse, ya que este no se encuentra a una distancia adecuada, como lo demuestra la Figura 28(b) donde el objeto está más cerca al sensor.

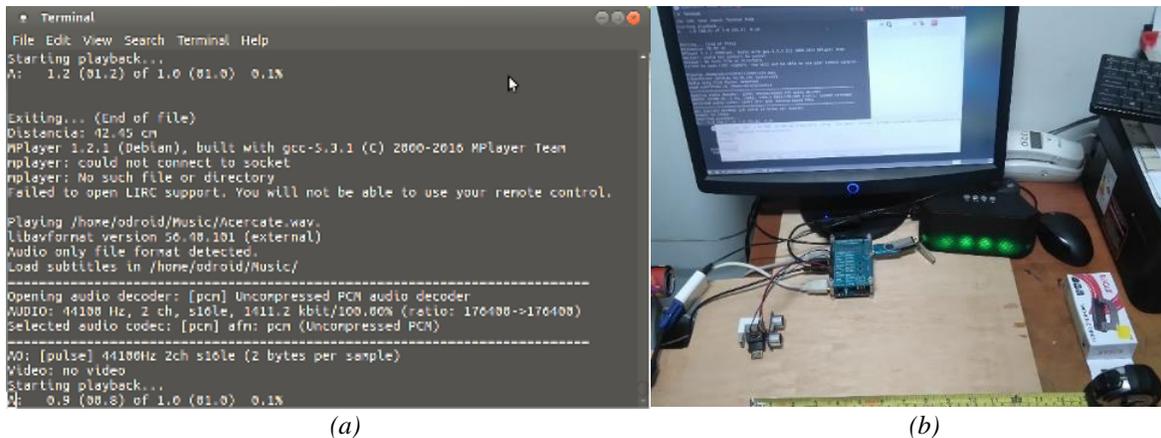


Figura 29. (a) Terminal Linux donde muestra distancia actual y reproducción de audio indicando que debe acercarse. (b) Medición entre sensor y un objeto en un punto más lejano a este.
Fuente autores.

Por otro lado, la Figura 29(a) muestra todo lo contrario de la anterior, dentro de la terminal de Linux, el sistema indica que la distancia en la que se encuentra el objeto es mucho mayor (42.45 cms), es decir que, el usuario debe acercarse al objeto para estar en una distancia adecuada. En la Figura 29(b) se observa como el objeto se encuentra más alejado del sensor, por lo tanto, el programa indica al usuario, por medio del sistema de alerta sonora, que debe acercarse más al objeto.

1.4.4. Captura de fotograma

Para el proceso de captura de la imagen, se hace uso de todas las funciones descritas anteriormente y se acoplan en un programa principal o *Main Program*, el funcionamiento de este programa va desde la configuración de los pines hasta la captura de los fotogramas presentes en el video. En la Figura 30 se ilustra la composición de este algoritmo de programación.

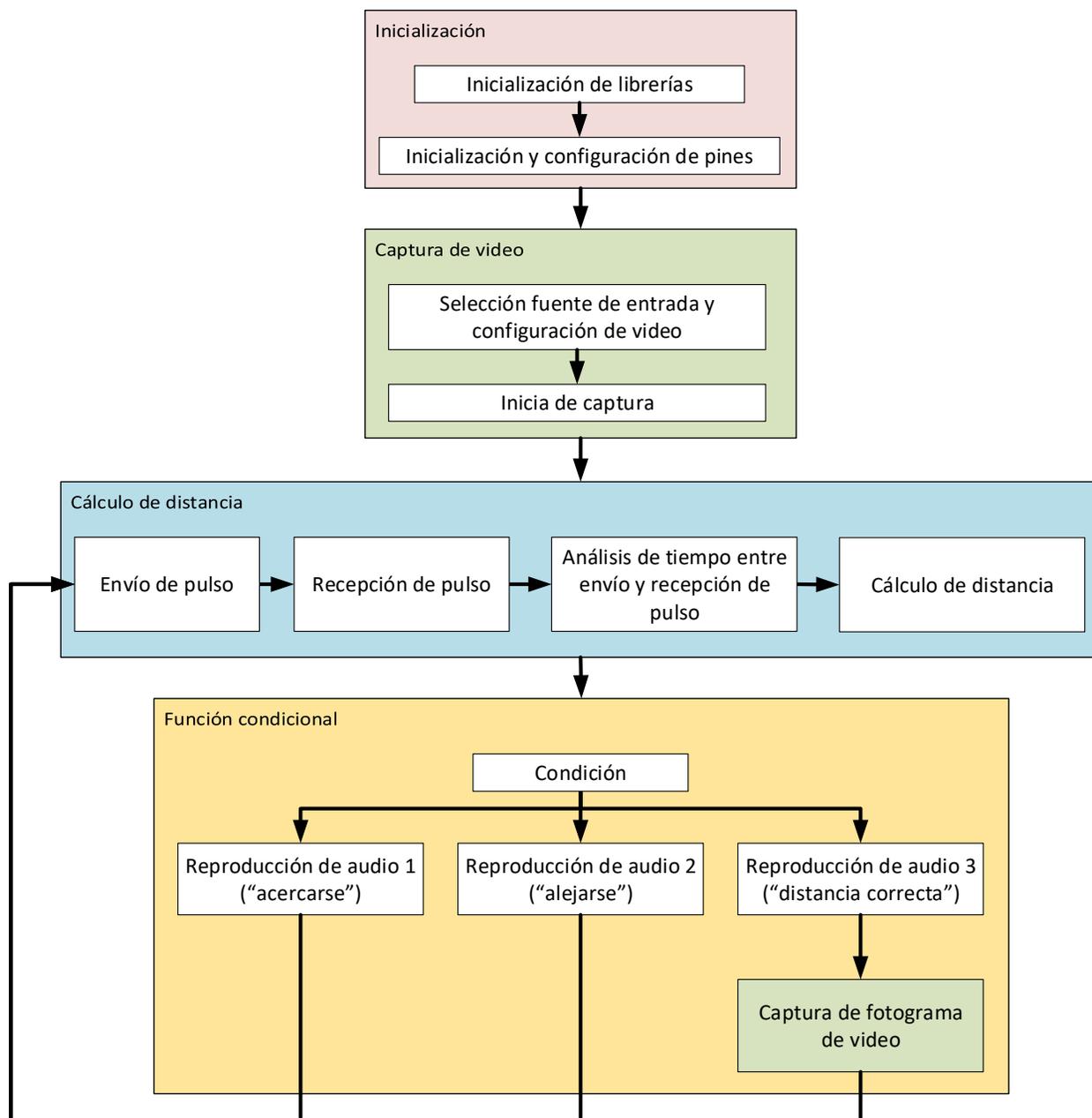
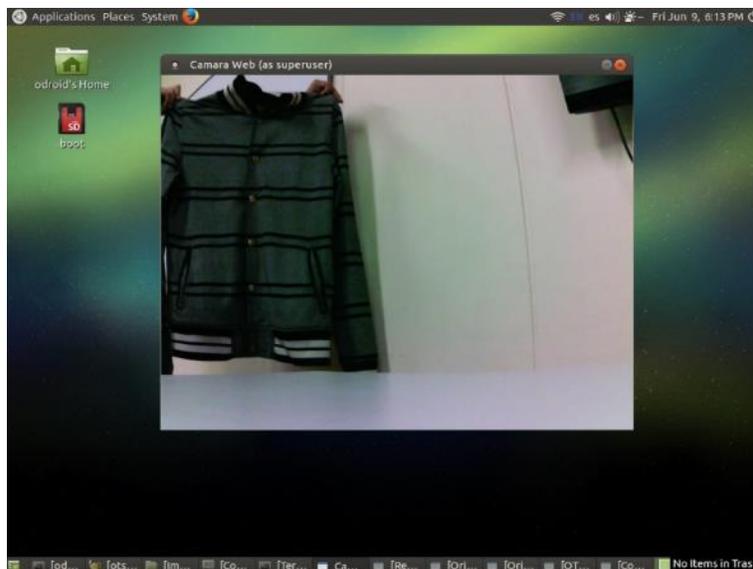


Figura 30. Diagrama de bloques del programa principal.
Fuente autores.

Para iniciar el proceso de captura de la imagen, se configura desde el programa el acceso al puerto donde se encuentra ubicada la cámara, luego se establece los fotogramas por segundo y la resolución de la imagen. Seguido a esto se guardan en una variable los valores representativos del fotograma, con el propósito de leer y manipular este de manera individual sin afectar la composición o las propiedades de la imagen que proviene directamente del módulo de cámara.

Paralelo a esto, se habilita el acceso a los pines del sistema embebido, que luego serán usados por el sensor para comenzar con la medición, y así obtener el valor de la distancia. Después de adquirir este valor, este servirá como entrada en la función de audio, que identificará si la distancia en la que se encuentra en ese momento es la correcta, o si debe acercarse o alejarse, mientras tanto, el programa entra en estado de espera (o *Stand-By*) hasta que cumpla la condición presente en la función de audio, y así, pueda iniciarse el proceso de captura del fotograma.

Adquirido el fotograma (véase Figura 31) este pasa a la función de preprocesamiento y a su vez, el sistema hace una retroalimentación para confirmar si la distancia no ha cambiado, y así continuar con el proceso.



*Figura 31. Imagen de captura de la cámara.
Fuente autores.*

Capítulo 2

PROCESAMIENTO DIGITAL DE IMÁGENES

2.1. Técnicas de procesamiento de imagen

Esta consiste en una transformación que debe pasar la imagen con el fin de destacar los detalles que serán procesados usando la técnica de segmentación seleccionada, en este caso se adecuará la imagen para que resalte detalles presentes en las prendas de vestir dentro de la imagen, lo que ayudará a obtener un mejor resultado. Para lograr dicho objetivo, dentro del proceso se combinarán diferentes técnicas para trabajar.

De acuerdo con lo anterior se implementan las siguientes operaciones puntuales:

2.1.1. Escala de grises

La escala de grises consiste en transformar la representación de una imagen a color a una escala que se extiende entre el color blanco y el negro, donde en cada píxel se representa con un valor numérico indicando el grado de luminiscencia. En el procesamiento digital de imágenes, “el término "escala de grises" se aplica al modo de color en el que cada elemento (píxel o de otro tipo) se describe dentro de una serie limitada de valores de un mismo tono neutro. En el caso de imágenes con una profundidad de 8 bits, los valores posibles teóricos son 256” (Cedano Olvera, Rubio González, & Vega Gutiérrez, 2014).

Este procedimiento se realiza para analizar ciertas áreas en la imagen adquirida que pueden ser descartadas en el procesamiento, para la conversión se usa uno de los comandos presentes en la librería de OpenCV en lenguaje de programación C++.

La función utilizada para la conversión del color es *cvtColor*, esta convierte cualquier imagen entrante de un espacio de color a otro. En caso de transformación “a RGB” o “desde RGB” el orden de los canales debe especificarse explícitamente (RGB o BGR).

Una de las recomendaciones que hacen los desarrolladores de OpenCV es que el formato de color predeterminado en el software es a menudo referido como RGB (Rojo – Verde – Azul), pero en realidad es BGR, es decir que las capas se invierten. Así que la primera capa en una imagen de color estándar (24 bits) será un componente azul de 8 bits, la segunda capa será verde y la tercera será rojo (OpenCV, 2014).

Para este proceso se toma el fotograma adquirido del módulo de cámara como entrada en la función *cvtColor*, aquí la función transforma el formato de color a tonos de grises de acuerdo con sus niveles de luminosidad o brillo, el resultado de esta transformación se presenta en la Figura 32.



*Figura 32. Imagen original en escala de grises.
Fuente autores.*

2.1.2. Ajuste de contraste

Esta técnica consiste en que, mediante el histograma de la imagen, se realiza el ajuste de contraste para cambiar la intensidad entre los tonos claros y oscuros con el fin de obtener un mejor enfoque y claridad en la imagen. Se efectúa este ajuste a la imagen mediante la modificación los valores de

intensidad en sus puntos mínimos y máximos, para luego obtener una imagen con calidad suficiente para ser procesada, en la Figura 33 se demuestra el funcionamiento de esta técnica.

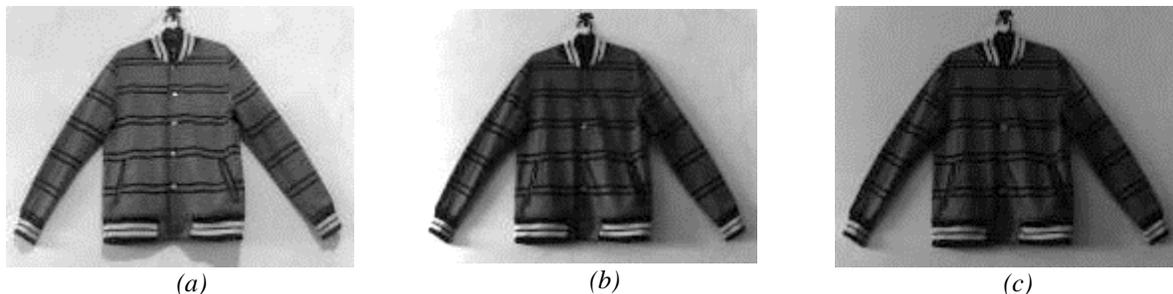


Figura 33. (a) Imagen original con +10 de contraste, (b) Imagen original, (c) Imagen original con -10 de contraste.
Fuente autores.

Esta técnica se encarga de ampliar el rango de los valores presentes en el histograma, por ejemplo, se obtiene el histograma de una imagen a escala de grises entre los valores 20 y 150, el aumento lineal de contraste efectúa, como su nombre lo indica, un aumento en el rango original al valor mínimo de 0 y al valor máximo de 255.

Las imágenes con poco contraste se presentan debido a varios aspectos, ya sea, por la configuración del diafragma al momento de la captura o por poca iluminación en el espacio donde se realiza la captura, el método consiste en incrementar el rango de los niveles de grises presentes en el histograma, tal cual como se muestra en la Figura 34.

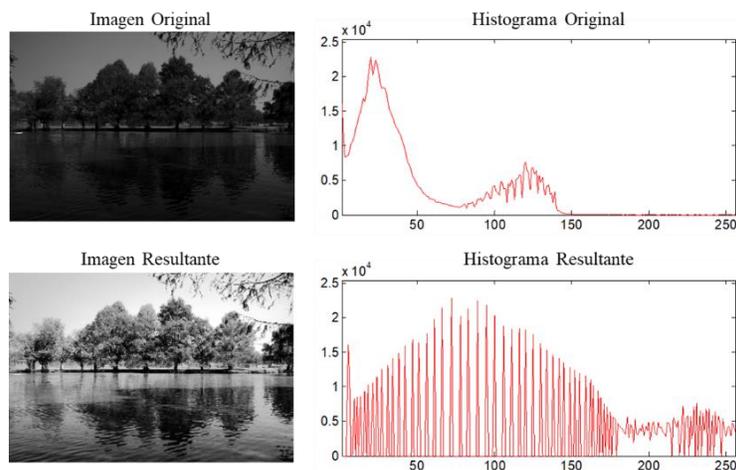


Figura 34. Ejemplo de aumento lineal de contraste.
Fuente Autores

El autor expresa en la página 92 del documento “Capítulo 3. Técnicas de Realce de Imágenes” el uso de una transformación para la mejora del contraste en la imagen:

Una transformación típica empleada para la mejora de contraste se muestra en la Figura 35. La ubicación de los puntos (r_1, s_1) y (r_2, s_2) controla la forma de la función de transformación. Por ejemplo, si $r_1 = s_1$ y $r_2 = s_2$ se genera una función lineal que no produce cambios en los niveles de gris. Si por el contrario $r_1 = r_2$, $s_1 = 0$ y $s_2 = L - 1$ la transformación se convierte en una función umbral (*Thershold*) que genera una imagen binaria. Se supone que $r_1 \leq r_2$, y si $s_1 \leq s_2$ de tal forma que la función sea de valor único y monótonamente creciente. La Figura 36 muestra una imagen con bajo contraste y la aplicación de la función de la Figura 35 con dos valores diferentes (Graffigna, 2016b).

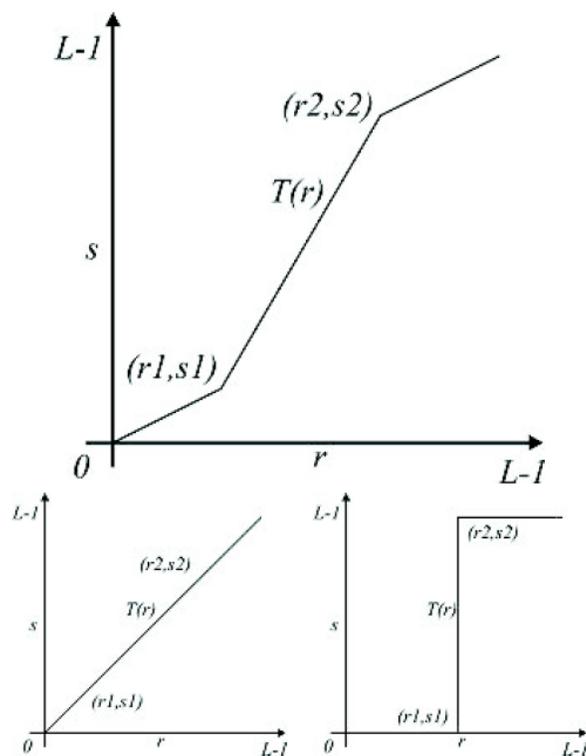


Figura 35. Funciones de transformación de pixeles.
Fuente (Graffigna, 2016b)

Los resultados de esta transformación se muestran en la Figura 36, donde en la primera figura, se muestra la imagen original opacada, luego, al aplicar el aumento de contraste, esta imagen se aclara, dando una mejor definición y nitidez a los objetos presentes en la imagen, para este, caso el paisaje con las cebras, esto se muestra en la segunda y tercera imagen.



*Figura 36. Imagen original con bajo contraste (arriba) y la aplicación de dos instancias de la función de aumento de contraste (medio y abajo).
Fuente (Graffigna, 2016b)*

Para la implementación de este operador en el algoritmo de programación se usa como entrada la imagen original la cual es sometida a un proceso de cambio de representación de RGB a escala de grises. Luego de esta conversión se procede a adquirir su histograma. Para ello se obtiene primero el valor en cada uno de los píxeles y se almacena en un vector. El siguiente paso es analizar el histograma y averiguar cuál es su valor máximo y su valor mínimo, con estos valores resultantes se procede a realizar una normalización a cada píxel mediante la siguiente fórmula:

$$valor\ normalizado = \frac{valor\ píxel - valor\ mínimo}{valor\ máximo - valor\ mínimo} \quad (3)$$

El valor normalizado (vn) es multiplicado por 255 con el fin de ampliar su rango de 0 a 1 (donde 0 es negro y 1 es blanco) a un rango de 0 a 255 y así poder guardarlo en una matriz para su posterior visualización, este resultado se usa como entrada en una función condicional expresada de la siguiente forma:

$$píxel = \begin{cases} 255 & \text{si } píxel > 255; \text{ sino } píxel = vn \\ 0 & \text{si } píxel < 0; \text{ sino } píxel = vn \end{cases} \quad (4)$$

Si el valor del píxel es mayor que 255 su nuevo valor será 255 sino no se modifica, pero si este es menor que 0 su nuevo valor será 0 de lo contrario sigue con su mismo valor. El resultado de esta condición se ingresa a una nueva matriz con las mismas dimensiones que la imagen original y en su correspondiente coordenada.

2.2. Segmentación usando métodos de umbralización (Thresholding):

Se utiliza esta técnica con el fin de separar nuestro objeto de interés, del resto de la imagen, para ello se hace una transformación no-lineal de la imagen de entrada para así convertirla en una imagen binaria, donde cada píxel puede tomar alguno de los dos valores: 0 o 1, negro o blanco, 0

o 255, esto dependiendo del valor del umbral que se desea tomar, este valor puede variar entre 0 hasta 255, tomando en cuenta la intensidad de luz deseada.

Para obtener automáticamente el valor del umbral, se utiliza el histograma de la imagen en escala de grises para luego obtener una imagen binaria.

2.2.1. Binarización

Un método básico para diferenciar un objeto presente en el primer plano del fondo de la imagen es mediante una simple binarización. A través del histograma obtenemos una gráfica donde se muestran el número de píxeles por cada nivel de gris que aparece en la imagen, tal cual como se muestra en la Figura 37, donde de una imagen original, se obtiene el histograma y de este se selecciona el valor de umbral con el cual se modificará la imagen.

La binarización de una imagen consiste en comparar niveles de gris presentes en la imagen con un valor (umbral) predeterminado. Si el nivel de gris de la imagen es menor que el umbral predeterminado, al píxel de la imagen binarizada se le asigna el valor 0 (negro), y si es mayor, se le asigna un 1 (blanco). De esta forma se obtiene una imagen en blanco y negro. Generalmente se utiliza un umbral de 128 si se trabaja con 256 niveles de gris, pero en algunas aplicaciones se requiere de otro umbral. La expresión matemática para esta técnica se expresa en la Ecuación 5 (Esqueda Elizondo & Maestre Palafox, 2005).

$$dst(x, y) = \begin{cases} if \ src(x, y) > T(x, y) = 1 \\ if \ src(x, y) < T(x, y) = 0 \end{cases} \quad (5)$$

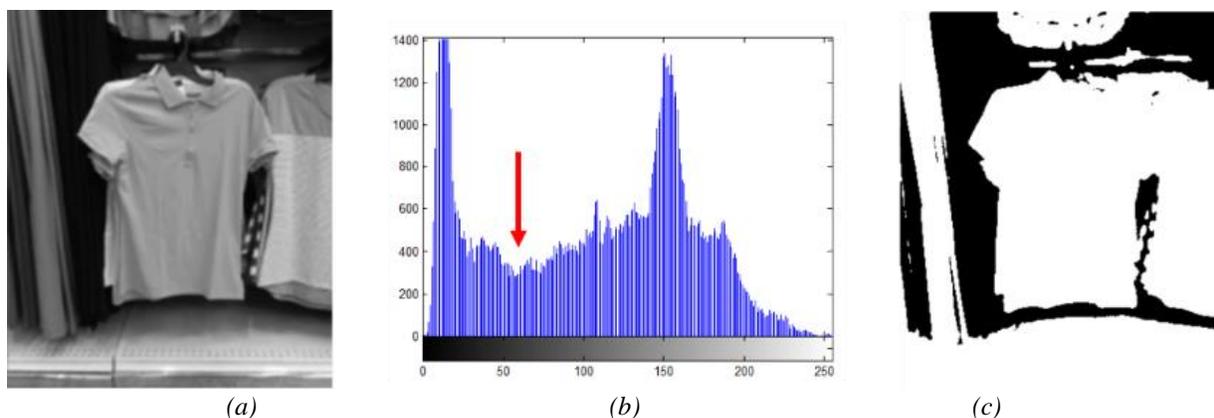


Figura 37. Imagen original (a). Histograma de la imagen original (b). Imagen binarizada con un umbral de 80 (c).
Fuente Autores

El método de binarización consiste en identificar los píxeles que posean un valor particular dentro un rango específico, este con el fin de encontrar en la imagen los objetos presentes en ella mediante sus niveles de brillo (véase la Figura 37 (c)), el autor Raúl Enriquez explica en su tesis que existen dos métodos para realizar la binarización: La Uniforme y la Adaptable.

La binarización uniforme consiste en modificar los píxeles que se encuentren por debajo del umbral a color negro y los que se encuentren por encima de este nivel se modifican a color blanco, para ello es necesario conocer los niveles de grises que posee la imagen, pero si estos niveles son desconocidos, se puede utilizar el histograma para identificarlos con facilidad, con la desventaja de disminuir el rendimiento del procesamiento de la imagen. Dentro de esta existen métodos más avanzados que, por medio de ecuaciones matemáticas, permiten determinar el nivel más adecuado para obtener una binarización óptima. Dentro de estas técnicas, la más usada y reconocida es el algoritmo de Otsu. Este usa la distribución de probabilidad para elegir el nivel del umbral que logre separar los objetos presentes en el primer plano con el fondo de la imagen.

Esto es logrado mediante la selección de un umbral que proporcione una mejor separación de clases, para todos los píxeles en la imagen. La base de este método es la utilización del histograma normalizado, donde el número de puntos en cada nivel es dividido entre el

número de puntos totales de la imagen. Así, la distribución de probabilidad de los niveles de intensidad está dada por la Ecuación 6:

$$P(l) = \frac{N(l)}{N^2} \quad (6)$$

Donde:

- $P(l)$ = Distribución de probabilidad
- $N(l)$ = Número de puntos por nivel de la imagen nueva.
- N^2 = Puntos de entrada

El método adaptable es más específico con respecto a su uso y es más utilizado en la binarización de documentos con texto con el fin de localizar caracteres específicos, además en el reconocimiento de patrones estáticos (Enriquez Vásquez, 2013).

Para la implementación del método de binarización dentro del algoritmo se usa un proceso similar como el explicado anteriormente en el ajuste de contraste:

Primero la imagen debe pasar por un proceso de cambio en su espacio de color RGB a escala de grises. Esto con el fin de obtener el valor de tono para cada pixel y luego guardarlo en una variable que será usado como entrada en una función condicional, la cual está conformada por la Ecuación 7:

$$valor\ pixel = \begin{cases} 255, & si\ pixel > 128 \\ 0, & si\ pixel < 128 \end{cases} \quad (7)$$

Este valor se guarda en una matriz nueva en la misma coordenada que fue indexada en la imagen original en escala de grises.

2.2.2. Método de umbralización Otsu

El algoritmo divide el histograma de la imagen en dos clases (*foreground* y *background*), usando un valor de umbral como la variación entre las clases. Con esto cada clase podrá ser compacta de lo mejor posible. La relación espacial entre píxeles no es tomada en cuenta, por lo tanto, sus regiones tendrán una similitud en el valor de los píxeles, pero estos estarán en lugares completamente diferentes en la imagen y serán unidas al calcular el histograma, esto significa que el algoritmo tomará dichos valores como iguales.

Asumiendo que los píxeles están categorizados en dos clases, el algoritmo trata de minimizar la magnitud de la variación ponderada dentro de las clases $\sigma_w^2(t)$ definida en la expresión de la Ecuación 8. La variable T es el umbral el cual es, comúnmente, un valor entre 0 y 255.

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \quad (8)$$

El proceso para calcular $\sigma_w^2(t)$ se describe a continuación. Una función de probabilidad de aparición P (La probabilidad de ocurrencia o aparición de un evento determinado, consiste en el nivel de certeza que tenemos de que ocurra dicho suceso, es la razón entre el número de veces en que ocurrió dicho evento y el número de repeticiones del experimento. A esta razón se le denomina frecuencia relativa) se obtiene de acuerdo con el valor de cada uno de los píxeles. Primero, la distribución de los valores del histograma es calculada, y luego se realiza una normalización con el fin de garantizar que siga una distribución de la probabilidad, esto consiste en dividir el valor de intensidad de cada píxel sobre el número de píxeles que posee la imagen. Después de esto, los valores de los píxeles son divididos en dos clases, C_1 y C_2 , de acuerdo con el valor del umbral utilizando las funciones de probabilidad acumulada definidas en las Ecuaciones 9 y 10:

$$q_1(t) = \sum_{i=1}^t P(i) \quad (9)$$

$$q_2(t) = \sum_{i=t+1}^I P(i) \quad (10)$$

La clase C1 representa los pixeles con niveles de intensidad desde 1 hasta el valor del umbral, y la clase C2 representa aquellos pixeles con niveles de intensidad en el rango $[T + 1, I]$ donde I es el valor máximo del pixel (comúnmente es 255), luego se calcula la media (acumulada) de la clase C1, μ_1 , y de la clase C2, μ_2 , que se obtienen de:

$$\mu_1(t) = \sum_{i=1}^t \frac{iP(i)}{q_1(t)} \quad (11)$$

$$\mu_2(t) = \sum_{i=t+1}^I \frac{iP(i)}{q_2(t)} \quad (12)$$

Luego de esto, las varianzas para las clases C1 y C2 se calculan por medio de las ecuaciones 13 y 14:

$$\sigma_1^2(t) = \sum_{i=1}^t [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \quad (13)$$

$$\sigma_2^2(t) = \sum_{i=t+1}^I [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)} \quad (14)$$

Las anteriores ecuaciones definen la varianza ponderada dentro de las clases para C1 y C2 respectivamente, estos son los valores que el algoritmo de *Otsu* intenta minimizar. Esta varianza es la medida de cuan compacta se encuentra cada clase, es decir que, si el método elige un valor de umbral erróneo, el valor de la varianza para cada clase va a ser mayor.

Usando las Ecuaciones 9, 14 y 15, la varianza total puede ser definida por la Ecuación 9 y 10, como la suma de la clase interna y la varianza entre clases. El valor de $\sigma_2(t)$ es constante, ya que

no depende del valor del umbral (la varianza de una imagen es siempre un valor constante), esto significa que el algoritmo debe enfocarse en minimizar $\sigma_w^2(t)$ o maximizar $\sigma_b^2(t)$ (Balarini & Nesmachnow, 2016).

$$\sigma^2 = \sigma_w^2(t) + \sigma_b^2(t), \text{ donde } \sigma_b^2(t) = q_1(t)q_2(t)[\mu_1(t) - \mu_2(t)]^2 \quad (15)$$

En la Figura 38 que se muestra a continuación se observa un ejemplo de los resultados que se pueden obtener el método Otsu:

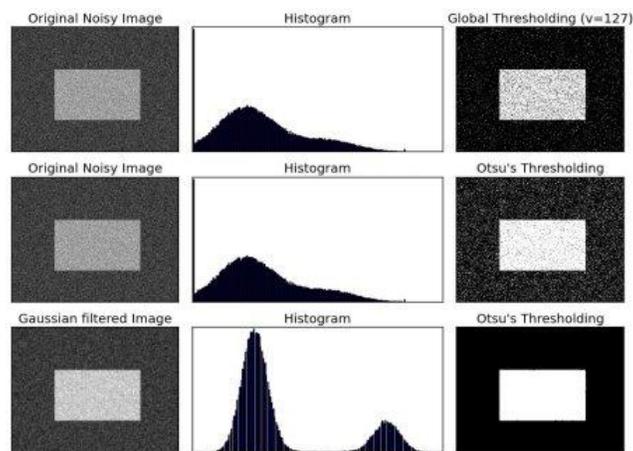


Figura 38. Ejemplo del algoritmo de Otsu.
Fuente (OpenCV, 2017)

En la primera columna se aprecia una imagen con ruido y a la cual se aplica una umbralización estándar (binarización) con valor de 127 en el umbral, en la siguiente columna se muestra la misma imagen que luego es procesada con el algoritmo de *Otsu*, dando un resultado muy similar al anterior, en la tercera y última se muestra la imagen original después de haber pasado por un filtro gaussiano, obteniendo un resultado similar al usar el método de binarización.

2.2.3. Resultados de la segmentación usando métodos de umbralización

Los métodos de umbralización abordados en este capítulo demuestran que estos poseen diferentes formas de aplicarse. Por ejemplo, el método de binarización estándar es solo usa una función condicional que, de acuerdo con un valor de umbral definido por el usuario, establece que valor

debe tener el pixel (0 ó 255) para su segmentación. En cambio, el algoritmo de *Otsu* emplea funciones de probabilidad (varianza, covarianza, media, mediana, entre otras) para dividir la imagen en dos distintas capas -primer plano y fondo de la imagen-, de este resultado, el algoritmo determina el valor del umbral óptimo, y así lograr una mejor segmentación de la imagen.

A continuación, se presenta una comparación entre los resultados obtenidos del método de binarización estándar y el método de umbralización automática usado el algoritmo de *Otsu*, con estos resultados se pretende demostrar las diferencias en cuanto a su funcionamiento y los resultados que se obtienen de estos.

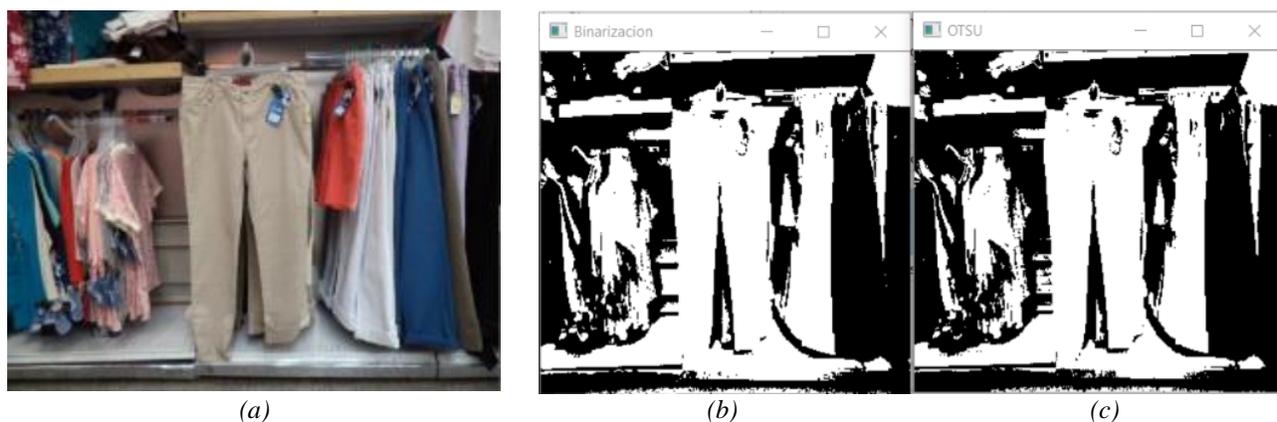


Figura 39. (a) Fotografía original de un pantalón para hombre. (b) Resultado obtenido del método de binarización estándar con valor de umbral 100. (c) Resultado obtenido usando el algoritmo de *Otsu*.
Fuente autores.

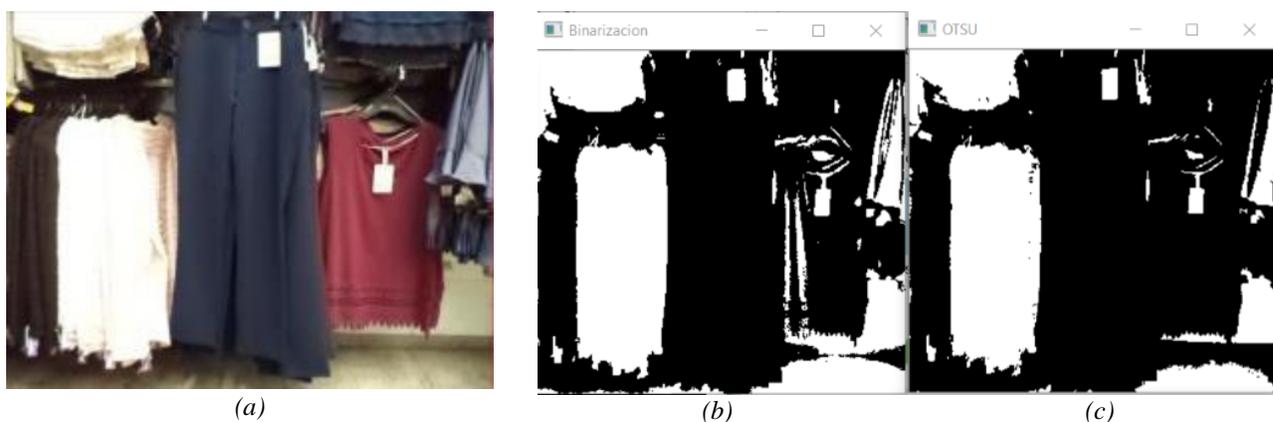


Figura 40. (a) Fotografía original de un pantalón para mujer. (b) Resultado obtenido del método de binarización estándar con valor de umbral 100. (c) Resultado obtenido usando el algoritmo de *Otsu*.
Fuente autores.



Figura 41. (a) Fotografía original de un buso. (b) Resultado obtenido del método de binarización estándar con valor de umbral 100. (c) Resultado obtenido usando el algoritmo de Otsu.

Fuente autores.



Figura 42. (a) Fotografía original de una camisa. (b) Resultado obtenido del método de binarización con valor de umbral 100. (c) Resultado obtenido usando el algoritmo de Otsu.

Fuente autores.

De acuerdo con estos resultados, se puede observar que el problema principal que se presenta al usar esta clase de segmentación es que este solo analizan los valores claros y oscuros dentro de la imagen, esto hace que sea difícil diferenciar los objetos que están contiguos y los cuales posean valores similares, haciendo que el método los reconozca como un solo objeto, por ejemplo, en la Figura 40, donde en el centro de la sección (a) se encuentra un pantalón de color azul y a su lado derecho una blusa roja, cuando se realizan los métodos de umbralización en escala de grises, dos se unen con parte del fondo, haciéndolos irreconocibles, como se muestra en las secciones (b) y (c). Este problema también sucede en la Figura 39.

Por otro lado, existen casos donde el objeto de interés es diferenciado de los demás objetos, ya que estos son de diferente tono, tal y como se demuestra en la Figura 41 (a), donde el objeto a procesar es un buso de color blanco, y en sus costados hay prendas de tonos oscuros. Al ejecutar los métodos, se obtiene que las prendas de tonos claros resaltan, como se muestra en las secciones (b) y (c). El buso está en la parte central y las otras prendas, en la parte derecha de la imagen, y pueden ser diferenciadas y procesadas. Otro caso similar sucede con la Figura 42, donde el objeto de interés posee un tono similar al fondo de la imagen (sección (a)). En este ejemplo en particular, se puede observar que el método binarización y el algoritmo de Otsu proporcionan resultados totalmente diferentes. En la sección (b) se demuestra como el objeto en el primer plano se combina con el fondo de la imagen, haciendo que no pueda usar en el procesamiento. En cambio, en la sección (c), se muestra una segmentación clara del objeto.

En conclusión, al usar métodos de segmentación basados en umbralización, deben tomarse en cuenta los factores que afectan el ambiente donde se realiza el procesamiento, como la iluminación que recibe, la posición o el orden en que se encuentra el objeto a procesar, además de los niveles de tonos de los colores presentes en los objetos contiguos o en el ambiente cercano, que afecten al primer plano.

Para el caso del proyecto, estos son factores que no pueden ser fácilmente modificados, ya que no pueden ser predeterminados en el espacio donde se realiza el procesamiento. Para modificarlos es necesario de permisos especiales para ajustar el ambiente y así tener el espacio adecuado para el procesamiento. Por lo tanto, el uso de estos métodos se hace inviable como herramienta principal para el cumplimiento de los objetivos del proyecto.

2.3. Segmentación por color usando crecimiento de regiones

2.3.1. Imagen segmentación usando textura y color (ISTC)

Este método permite hallar objetos de interés en el primer plano de una imagen, pero al realizar este proceso, se pueden encontrar otros objetos que perjudican la búsqueda de un objeto en particular o extraer este, dificultando la ejecución de los clasificadores. La segmentación ayuda a identificar la región donde se encuentra el objeto de interés, para que pueda ser usado como entrada para el clasificador e identificarlo. Al lograr disminuir las diferentes variables presentes en la imagen se podrá saber que clasificadores usar para extraer el objeto.

La mayoría de librerías presentes en los softwares de procesamiento de imágenes, ofrecen funciones que permiten segmentar de manera automática. Pero estos solo se enfocan en segmentar los colores y no las texturas. Este método, como lo indica en su nombre, toma tanto las características del color como de la textura para obtener una segmentación más clara.

Se implementa un algoritmo, que se describe en la Figura 43, para la agrupación de colores que posean valores similares. Antes se realiza una conversión de formato de color RGB a HSV, con el fin de facilitar la medición de los valores de los píxeles. Se distribuyen puntos de manera aleatoria, con el propósito de medir los valores de los píxeles e ir agrupando los que posean un valor similar, de allí se obtendrá una segmentación de color aproximada.

Después de obtener la segmentación por color se procede a hallar la dirección del gradiente a cada uno de los píxeles, luego estos se agrupan formando sectores de 20x20 píxeles con superposiciones. Estos sectores definirán el patrón de textura de la región tomando en cuenta las direcciones del gradiente. Se usan medidas estadísticas como la media, la varianza y la densidad, para cuantificar la textura y así obtener una mejor comparación. Lo siguiente que se hace es tomar

las regiones que posean una textura similar y aplicar el mismo método anterior para segmentar el color (Dhall, 2015).

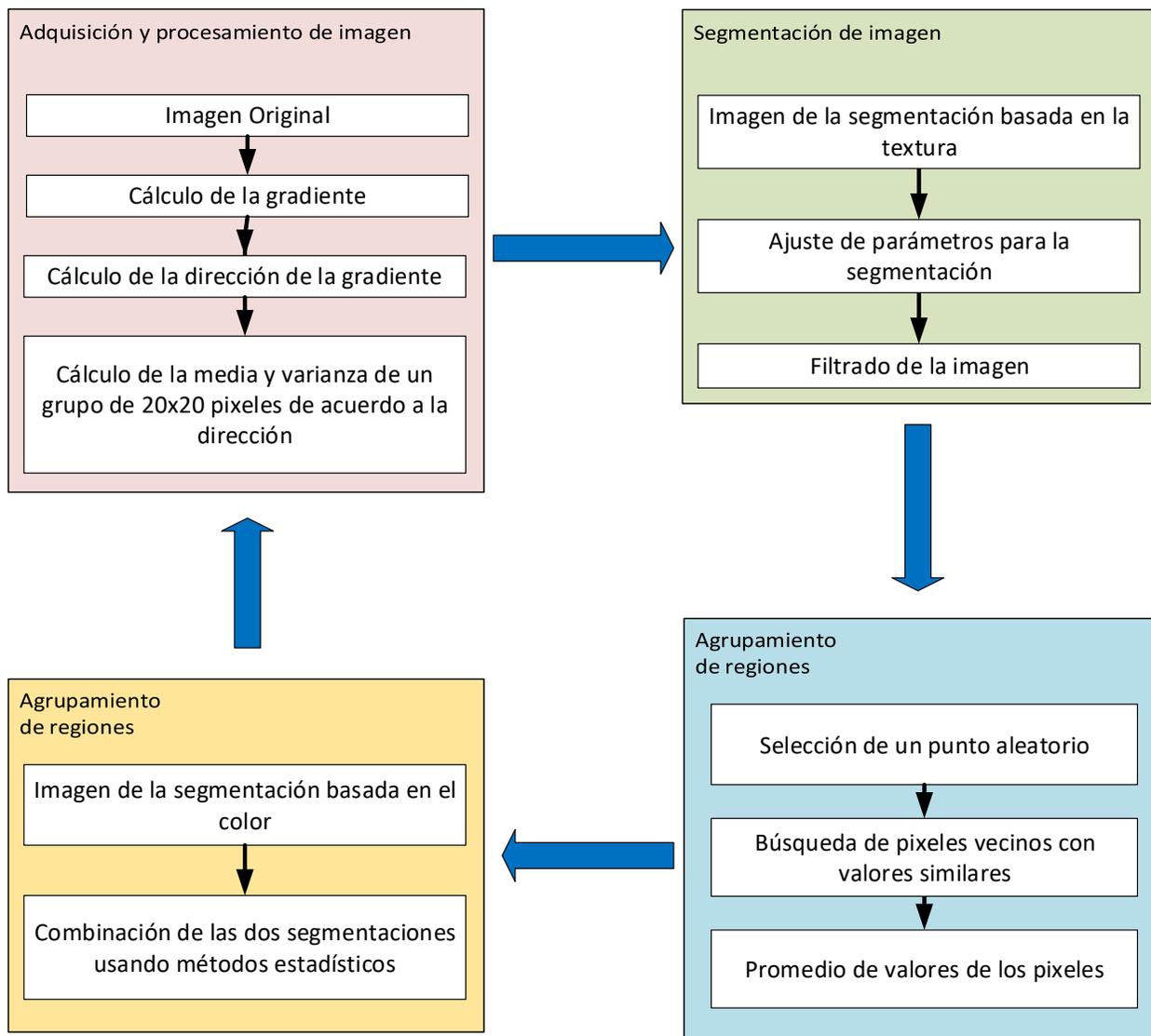


Figura 43. Diagrama de bloques del método ISTC.
Fuente autores.

Para finalizar, se combinan los dos resultados y se obtiene una segmentación más clara de la imagen, la desventaja de usar una técnica como estas es que no se logra definir en qué momento es mejor usar solo la segmentación por color o por textura, además esta técnica puede usarse para mejorar el resultado de otras técnicas (Dhall, 2015).

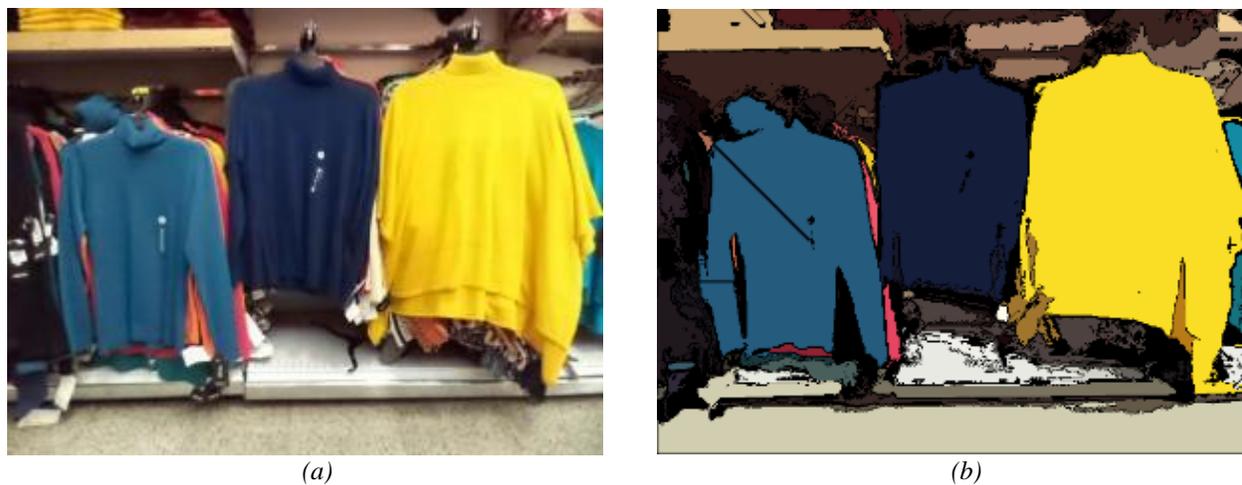


Figura 44. (a) Imagen original. (b) Resultado obtenido usando el método ISTC.
Fuente autores.



Figura 45. (a) Imagen de una camisa con un diseño en su textura. (b) Resultado de la segmentación usando el método ISTC.
Fuente autores.



Figura 46. (a) Fotografía de un pantalón. (b) Resultado de la segmentación de la imagen original.
Fuente autores.

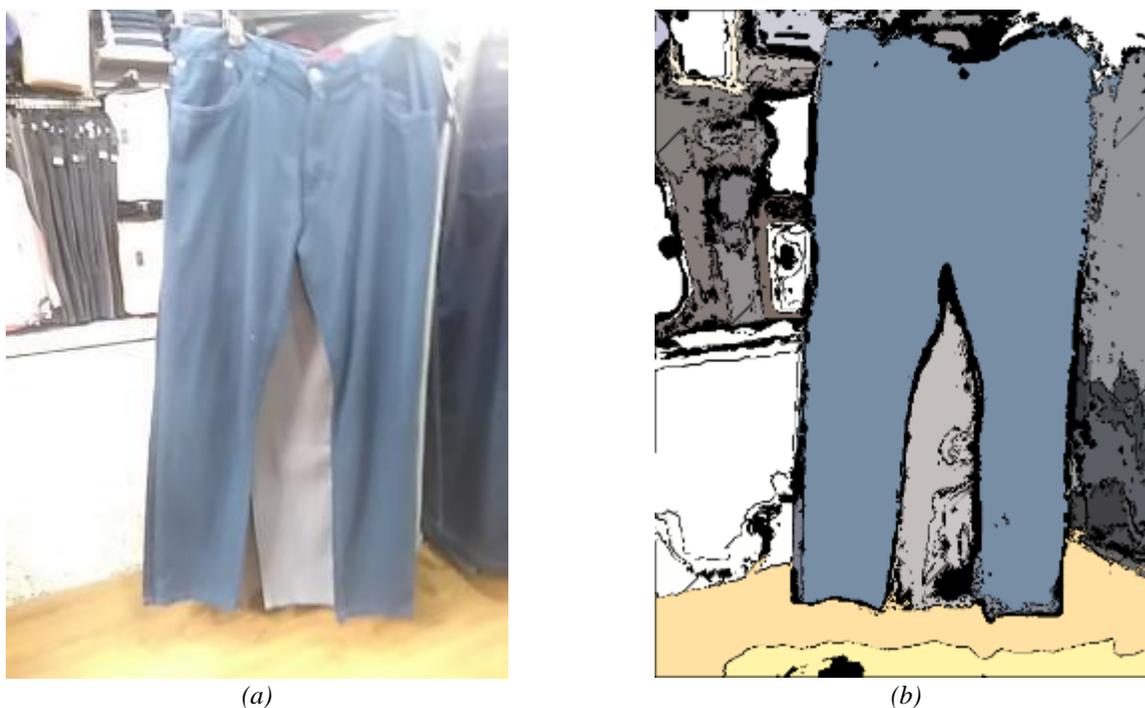


Figura 47. (a) Fotografía de un pantalón. (b) Resultado de la segmentación realizada con ISTC.
Fuente autores.

De acuerdo con los resultados obtenidos, el método de segmentación no resulta ser eficiente cuando el objeto a procesar (la prenda de vestir), posee algún tipo de estampado o diseño en su textura. El algoritmo no pueda determinar el color dominante presente en la prenda, como se muestra en la Figura 45 (b). Otro caso sucede cuando el objeto de interés tiene un color similar a los objetos contiguos, haciendo que el sistema los reconozca como uno solo y los agrupe, tal es el

caso de la Figura 46 (b), donde el tono de la prenda es similar al tono del piso de la repisa. Existen otros casos cuando los colores dentro de la imagen favorecen el procesamiento, como sucede en la Figura 44(a). Los objetos de interés poseen un tono de color sólido y vivo, que resalta más que los colores presentes en el fondo de la imagen, permitiendo ser más fáciles de identificar y segmentar por el algoritmo, tal cual como se muestra en las Figura 44(b) y Figura 47(b).

Otro aspecto para tomar en cuenta es que, como en la mayoría de los casos presentados, la iluminación debe ser uniforme para que no presente alguna variación en los valores de los píxeles y así obtener una buena segmentación.

De acuerdo con lo anterior, no se opta por establecer esta técnica como solución al objetivo propuesto en el proyecto, debido a las diferentes deficiencias que se presenta al usar este método. Además, la duración del procesamiento es bastante alta ya que debe elegir píxeles al azar, lo cual puede variar dependiendo del tamaño de la imagen.

2.3.2. Crecimiento de regiones

Este método inicia con un conjunto de pequeñas áreas que se fusiona con sus vecinos cercanos que posean un valor del píxel similar. Se selecciona de manera aleatoria el píxel semilla y el valor de este se compara con los píxeles vecinos para formar una región, este va creciendo a medida que los valores de los píxeles vecinos sean similares al píxel semilla (véase la Figura 48), de ahí su nombre de crecimiento de regiones, cuando los valores de los píxeles vecinos cambien y detengan el crecimiento de la región, el algoritmo elige otro píxel semilla que tenga un valor distinto y empieza nuevamente el proceso, este es un método iterativo por lo que este se repite hasta que todas las regiones cubran el tamaño de la imagen.

Sin embargo, este método presenta una desventaja cuando, en la mayoría de la imagen, los valores de los píxeles son bastante similares, haciendo que el primer píxel semilla se extienda por toda la imagen sin dejar que otro píxel semilla sea medido, y así segmentando toda la imagen con la región que se calculó al principio.

Esto puede causar efectos indeseables como lo son:

- La región que se esté procesando domine todo el proceso de crecimiento, obteniendo que otras regiones no se procesen correctamente.
- Diferentes opciones de los píxeles de inicio (píxel semilla) puedan dar diferentes resultados en su segmentación.
- El píxel de inicio (píxel semilla) se posicione en un borde.

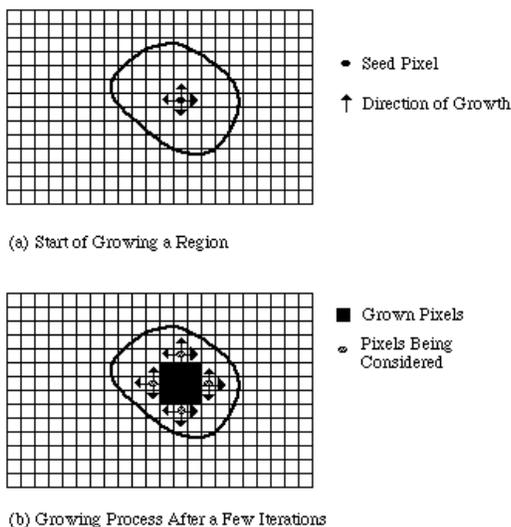


Figura 48. Posicionamiento del "seed pixel" para la región de crecimiento (a). Proceso repetitivo para la región en crecimiento (b).

Fuente (Marshall & Martin, 1992)

Para reducir estos efectos, se han desarrollado técnicas para obtener un crecimiento de región simultánea (*simultaneous region growing*), donde:

- Regiones similares en su valor se toman en cuenta para el proceso de crecimiento

- Evitar que una sola región domine todo el resultado del procedimiento
- Un número de regiones pueden procesarse al mismo tiempo, y al llegar a ser similares se acoplan para formar una sola región (Marshall & Martin, 1992).

Para desarrollar este método, teniendo en cuenta el fundamento del tema de crecimiento de regiones, se creó una función para esta técnica con cuatro argumentos de entrada; la matriz de la imagen normalizada, la coordenada del punto centro de la imagen (x, y) y el valor de umbral, el cual representa la máxima diferencia de nivel de pixel, como se ve en la Ecuación 16:

$$|P(x, y) - P_{seed}(x, y)| < T \quad (16)$$

Antes de implementar este método, se ejecutó una conversión del color original (RGB) a escala de grises, luego se normalizaron los valores de sus pixeles siguiendo la Ecuación 17:

$$\text{Valor normalizado} = \frac{\text{valor pixel}}{255} \quad (17)$$

Luego se selecciona la coordenada que ubique el centro de la imagen ya que es donde se encuentra la prenda. Dentro de la función de dicho método, se creó una variable para almacenar el dato del pixel en el centro de la imagen y un arreglo para almacenar los valores que cumple con la condición, la cual se describe en la Ecuación 18:

$$P(R_i) = \text{Verdadero: si } |P(x, y) - P_{seed}(x, y)| < 0.15 \quad (18)$$

Donde:

- $P(R_i)$ refiere a la región en crecimiento
- $P(x, y)$ refiere al pixel vecino el cual se está analizando
- $P_{seed}(x, y)$ refiere al pixel que se toma en el centro de la conectividad

Si $P(R_i)$ resulta verdadero esto quiere decir que el pixel z puede tomarse para la región en crecimiento, de lo contrario no se toma en cuenta y se repite el proceso con otro pixel vecino. Se

toma como valor de umbral 0.15 ya que este demuestra ser un valor óptimo para la segmentación, esto se comprueba comparando con diferentes umbrales, como se muestra en las siguientes figuras:

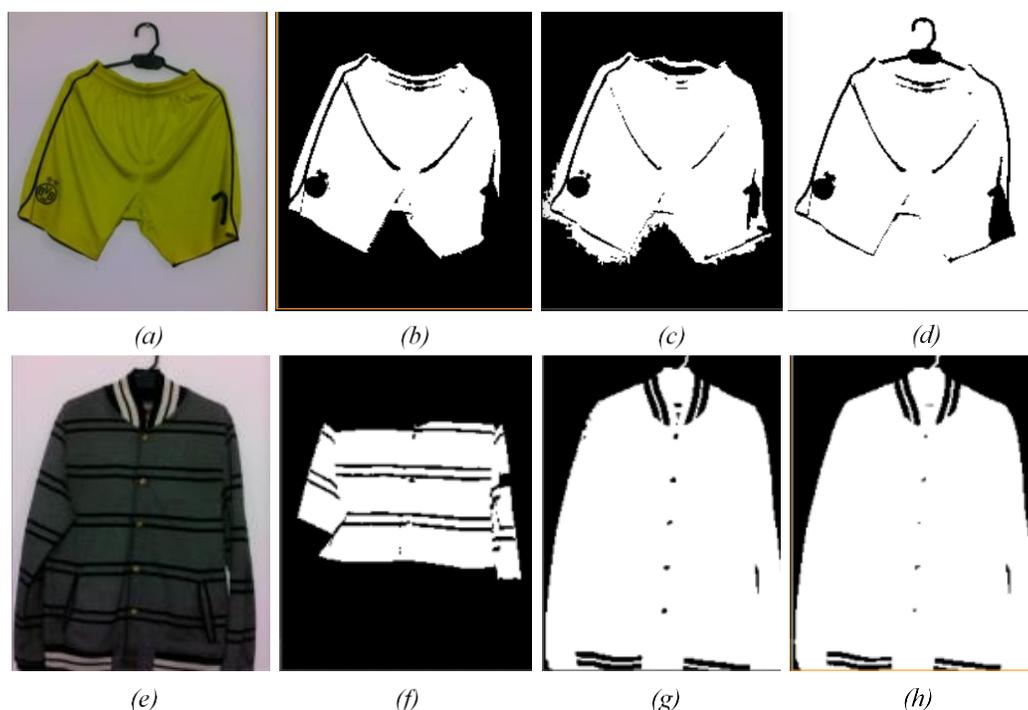
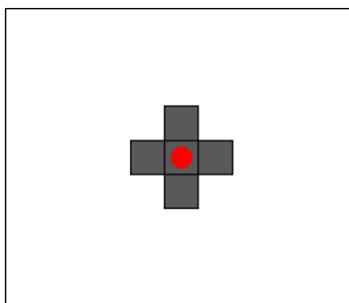


Figura 49. (a, e) Fotografía original con recorte. (b, f) Resultado obtenido con valor de umbral 0.10. (c, g) Resultado obtenido con valor de umbral 0.15. (d, h) Resultado obtenido con valor de umbral 0.20.
Fuente autores.

Como se muestra en la Figura 49 entre menor sea el valor del umbral, se formarán más regiones pequeñas, pero se mantendrán la homogeneidad de la figura como el caso de la sección (f), en cambio al aumentar el valor del umbral, las regiones serán pocas, pero más grandes, afectando la homogeneidad y provocando una mayor desviación típica como se ve la región (d) (Graffigna, 2016a).

Una parte importante del crecimiento de regiones es mapear los pixeles, en este caso se aplica la conectividad 4 o vecindad 4, esto significa que se analizará solo a los cuatros vecinos en la forma cruz como se ve en la Figura 50.



*Figura 50. Conectividad 4 en el punto centro de la imagen (Círculo rojo) en la matriz de escala grises.
Fuente autores.*

Tomando los datos de los píxeles vecinos para hallar el resultado de la diferencia entre estos valores y el del centro de la conectividad, se busca cuáles de ellas cumplen con la condición del umbral. Si cumple esta condición, se agrega un bit con valor “1” a un arreglo previamente creado, luego se busca el valor mínimo del resultado de la operación anterior, con el fin de ubicar el nuevo centro de la conectividad, y así mapear los nuevos vecinos presentes en la siguiente secuencia hasta que no cumpla con la condición o hasta hallar los píxeles vecinos que estén fuera del umbral.

En las Figura 51, Figura 52, Figura 53 se muestran los diferentes resultados del crecimiento de regiones según los valores del píxel presente en centro de la matriz. En la (b) de todas las figuras representan la conversión de escala de grises de la imagen original mientras que (c-e) pertenecen a los canales R, G y B respectivamente. Estas fueron distribuidas con el fin de conocer los diferentes valores presentes en el píxel central en cada capa, y así determinar que estos no son homogéneos, como se puede ver en la numeración (f) donde se indica los datos de cada uno de los canales, incluyendo los datos de la conversión de HSL. Este modelo de color se representa en tres componentes diferentes como tono, saturación y luminosidad, el primero componente se encarga en denotar el color base mediante números o grados. El segundo describe la relación entre el radio de tonalidad y la luminosidad. El tercer componente se define como el brillo relativo emitido por una iluminación blanca (Mary Jenitha & Bashashaik, 2015). Sin embargo, existe resultados los

cuales no obtuvieron buena segmentación como el caso de (g) de la Figura 51, (g-h) de la Figura 52 y (h) de la Figura 53 debido a que hay factores como la iluminación o características presentes en la figura objetivo que generan ruido y no permiten una buena segmentación (objetos con colores similares).

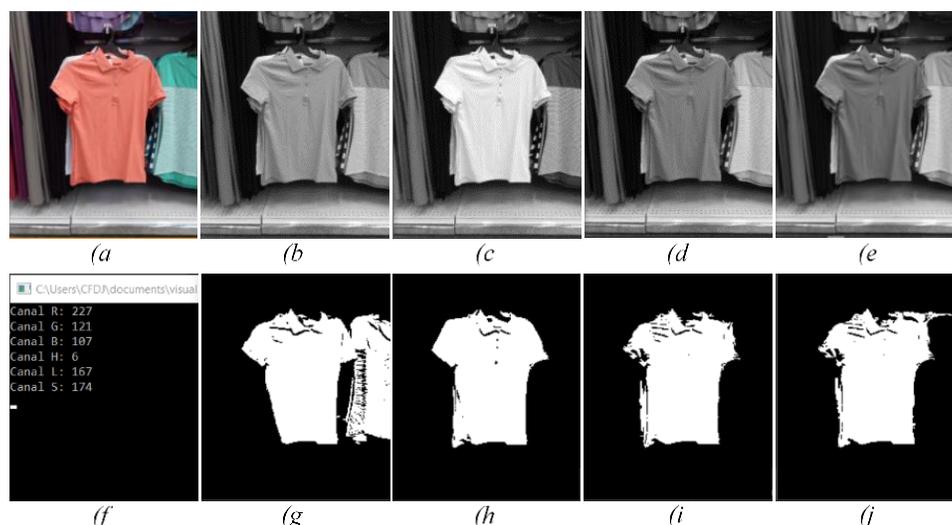


Figura 51. (a) Imagen de la camiseta roja. (b) Escala de gris de la imagen original. (c) Canal R de original. (d) Canal G de original. (e) Canal B de original. (f) Valor de pixel presente en centro de la imagen. (g) Crecimiento de regiones de escala de gris. (h) Crecimiento de regiones de canal R. (i) Crecimiento de regiones de canal G. (j) Crecimiento de regiones de canal B.

Fuente autores.

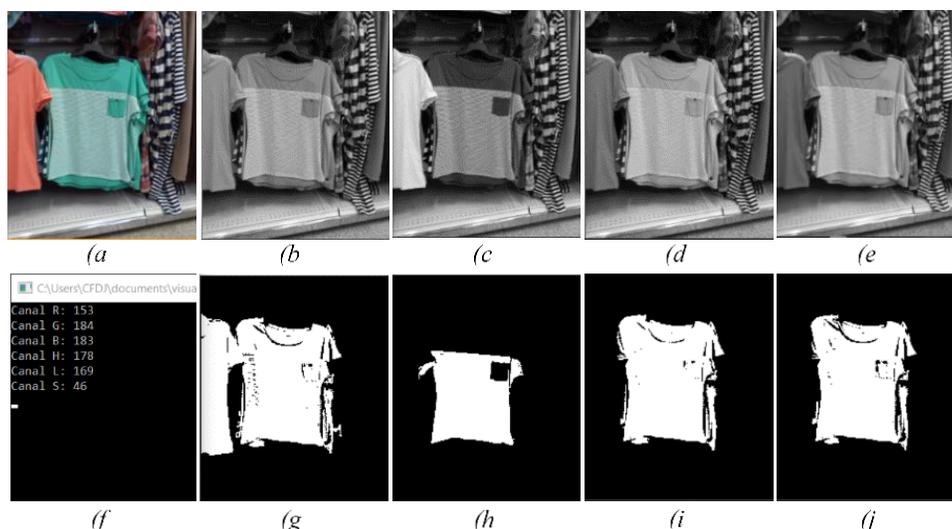


Figura 52. (a) Imagen de la camiseta verde. (b) Escala de gris de la imagen original. (c) Canal R de original. (d) Canal G de original. (e) Canal B de original. (f) Valor de pixel presente en centro de la imagen. (g) Crecimiento de regiones de escala de gris. (h) Crecimiento de regiones de canal R. (i) Crecimiento de regiones de canal G. (j) Crecimiento de regiones de canal B.

Fuente autores.

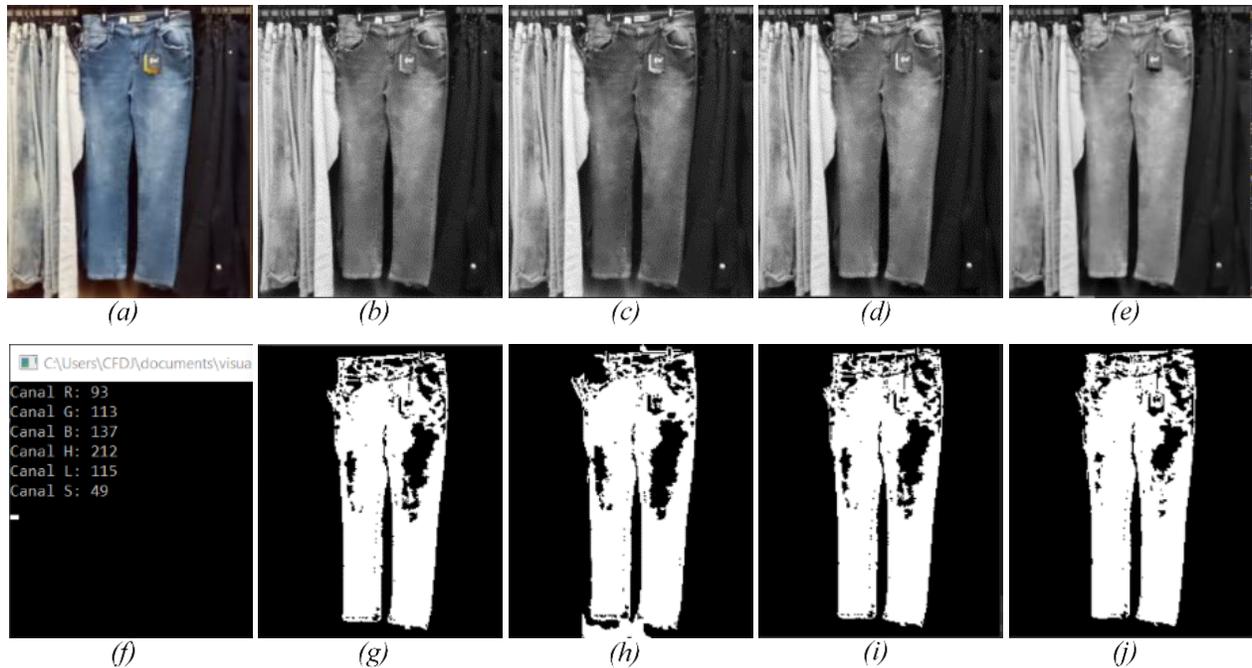


Figura 53. (a) Imagen de la camiseta roja. (b) Escala de gris de la imagen original. (c) Canal R de original. (d) Canal G de original. (e) Canal B de original. (f) Valor de pixel presente en centro de la imagen. (g) Crecimiento de regiones de escala de gris. (h) Crecimiento de regiones de canal R. (i) Crecimiento de regiones de canal G. (j) Crecimiento de regiones de canal B.

Fuente autores.

Para corregir este problema, se implementa una conversión de RGB a HSL, con el fin de obtener el valor del tono presente en el canal H (*Hue*). Se crea una función que, de acuerdo con el valor del tono tomado, determina en que canal R, G o B es recomendable realizar el crecimiento de regiones, esto se hace por medio del círculo cromático, como se muestra en la Figura 54, el cual define los valores de los colores como ángulos. Para definir el canal dominante de acuerdo con el matiz se aplica la Ecuación 19.

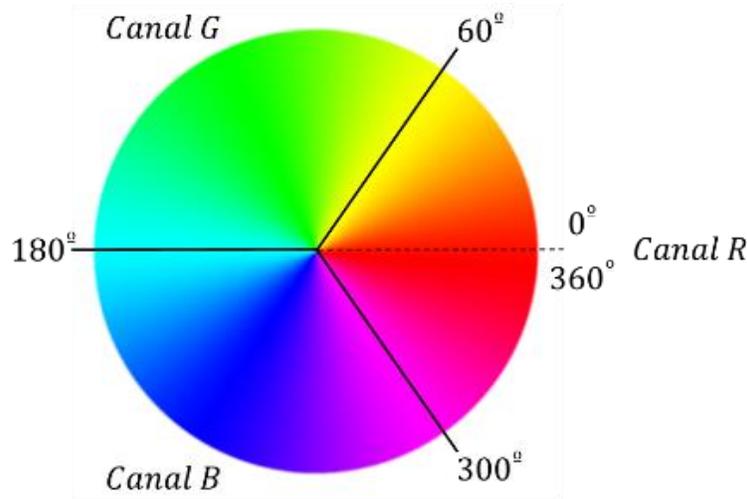


Figura 54. Círculo cromático.
Fuente autores.

$$H = \text{Canal}_H(x, y)$$

$$\text{Posicion de canal RGB} = \begin{cases} \text{Si } H > 300 \text{ y } H \leq 60 & = 2 \text{ Canal Rojo} \\ \text{Si } H > 60 \text{ y } H \leq 180 & = 1 \text{ Canal Verde} \\ \text{Si } H > 180 \text{ y } H \leq 300 & = 0 \text{ Canal Azul} \end{cases} \quad (19)$$

El valor de tono H ingresa dentro de una función condicional y de acuerdo con su valor, la función se encarga en seleccionar uno de los tres canales de RGB, el cual mejor se adapta para realizar la segmentación. Al seleccionar dicha capa, el programa nuevamente ejecuta el algoritmo de región en crecimiento, y así obtener satisfactoriamente una nueva segmentación.

En la Figura 55 se presenta el funcionamiento del algoritmo, donde se usa la imagen proveniente del módulo de cámara como entrada en la función, luego esta imagen es transformada del formato de color original (RGB) a HLS, con el fin de dividir la imagen por capas y usar la capa de tono para analizar y deducir que valor de tono es dominante en el primer plano de la imagen procesada, esto para obtener una segmentación más detallada de esta.

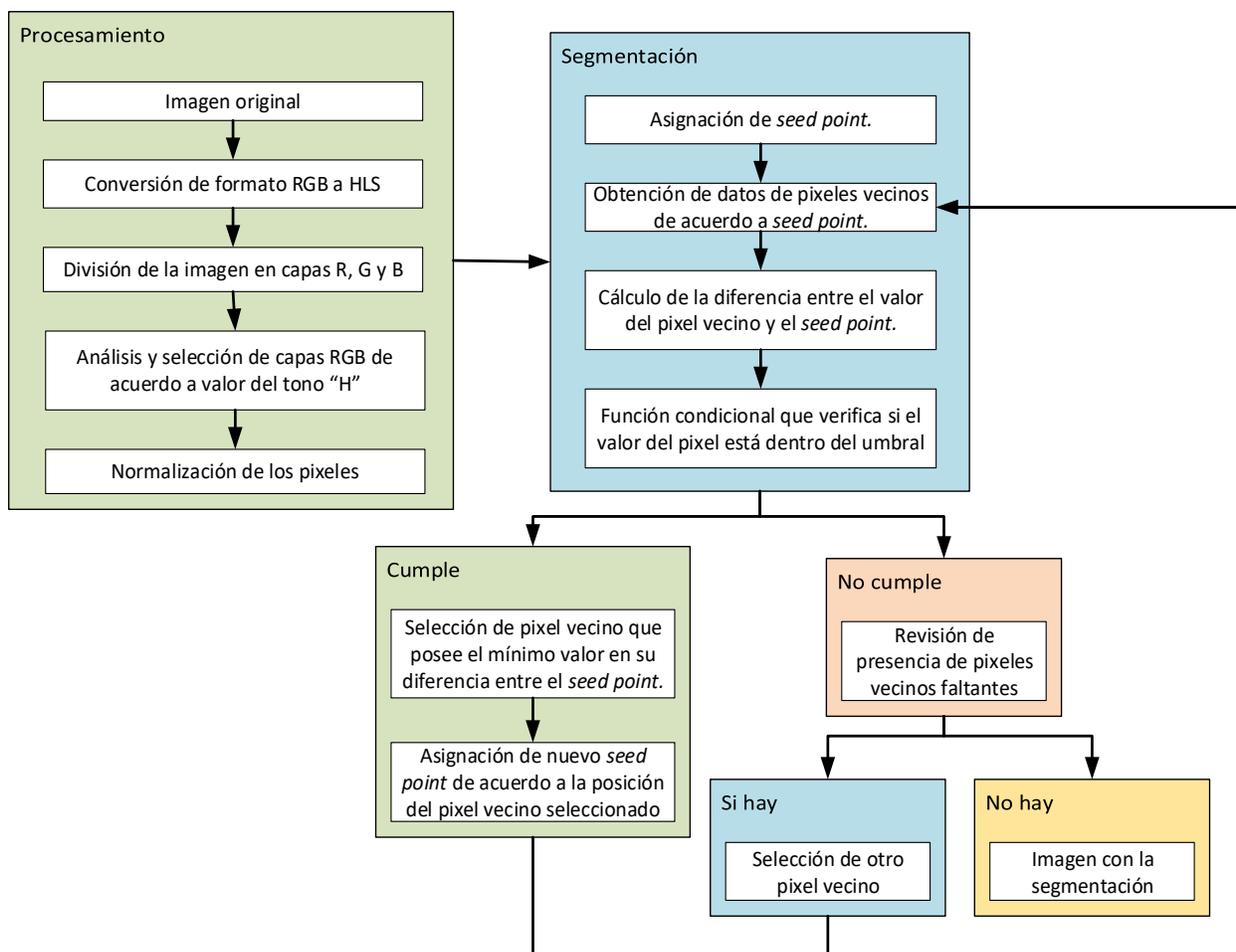


Figura 55. Diagrama de bloques del método de segmentación crecimiento de regiones.

Fuente autores.

De acuerdo con los resultados obtenidos usando este método de segmentación por color, se puede determinar que este es el apropiado para usarse como técnica de procesamiento, ya que permite reducir la mayoría de factores que afectan la imagen, y así obtener el objeto de interés presente en el primer plano de la imagen. Un factor que es difícil de reducir es el brillo, el cual es provocado por la iluminación en el espacio donde se procesa la imagen, para quitar este es necesario de un procesamiento más complejo y dedicado a solucionar este problema. Este proceso no se realiza debido a que excede las limitaciones en proyecto del pregrado.

2.4. Construcción del dispositivo

Una vez organizada toda la parte software del sistema, se procede con la elaboración física del prototipo. Esto consiste en la instalación de los componentes necesarios para el funcionamiento del sistema. Estos componentes son: el sistema embebido, el módulo de cámara, el sensor de distancia y el adaptador de audio. Para lograr que el prototipo sea portable, se le incorpora una batería de 12 voltios a 9000 miliamperios-hora. Además, se instala un regulador de voltaje, debido a que el sistema embebido consume solo 5 voltios de energía. En la Figura 56 se muestra la fuente de energía y el regulador respectivamente.

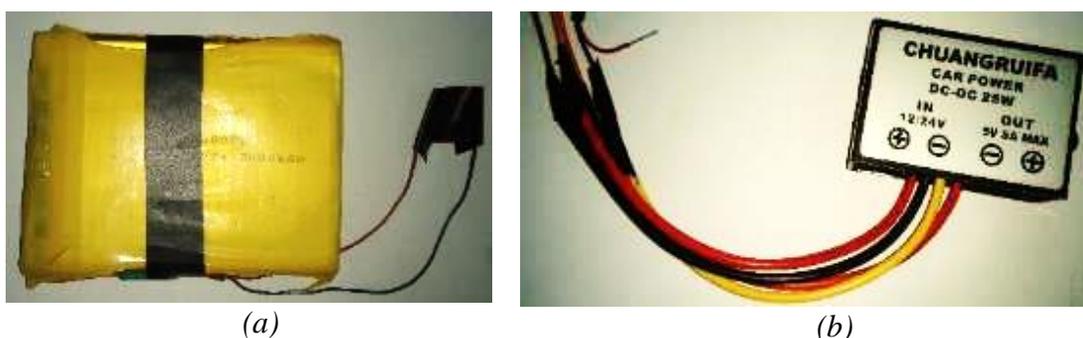


Figura 56. (a) Batería de 12 voltios a 9000 miliamperios-hora. (b) Regulador de voltaje con entrada de 12 voltios y salida a 5 voltios.
Fuente autores.

Teniendo en cuenta que los materiales a usar para el prototipo ocupan mucho espacio, se decidió dividir el dispositivo en dos módulos. El primero consiste en la parte de procesamiento, donde se encuentra ubicado el sistema embebido, el adaptador de audio y la fuente de energía. En el segundo módulo se encuentra toda la parte de adquisición de datos, el cual está compuesto por la cámara y el sensor de distancia. A continuación, Figura 57 se muestra el estado final de construcción del prototipo:

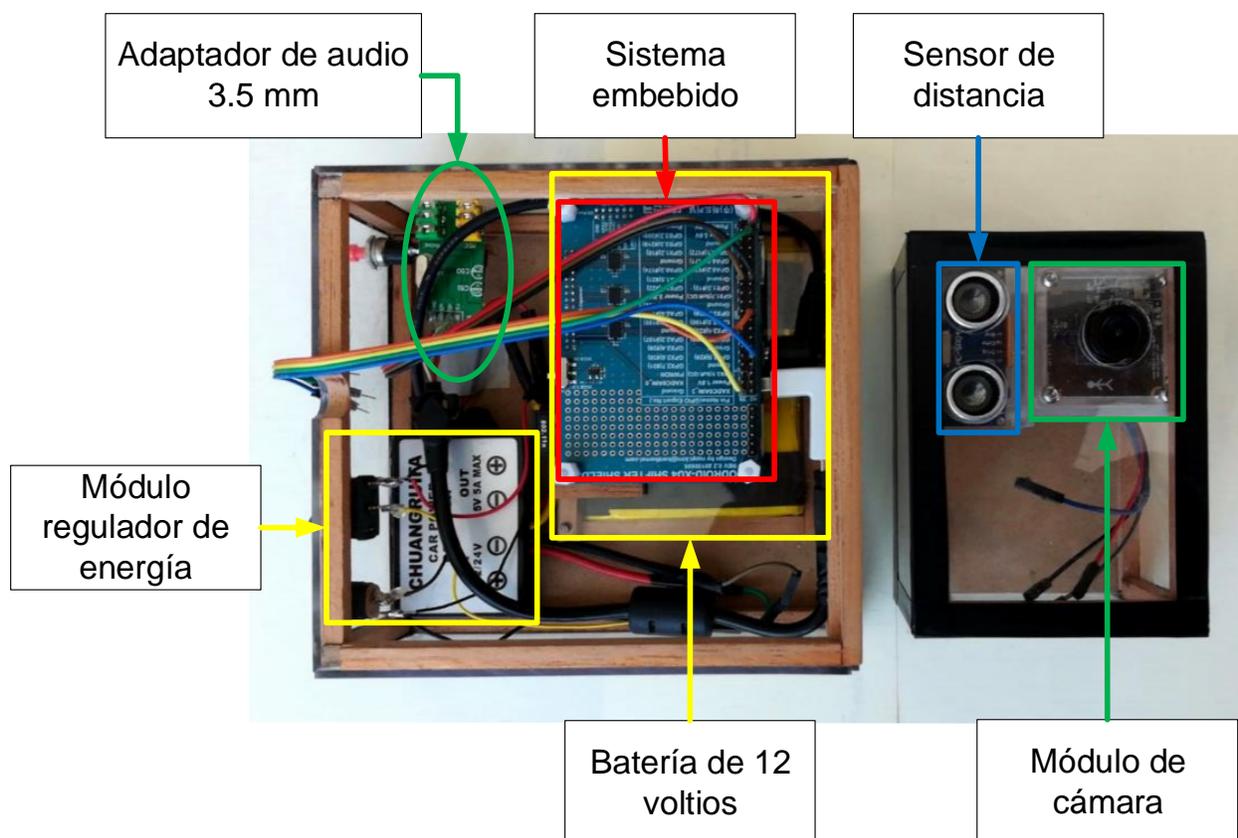


Figura 57. Componentes que conforman el sistema.
Fuente autores.

Capítulo 3

EXTRACCIÓN Y CLASIFICACIÓN DE CARACTERÍSTICAS

3.1. Restauración y reconstrucción

Después de realizar el proceso de segmentación en la imagen, se puede observar que algunos de estos resultados poseen algún tipo de interferencia, lo que puede generar errores en el proceso de clasificación. Por ello antes de continuar, se implementan métodos de restauración y reconstrucción que ayudarán a corregirlos, y así reducir dichos errores de la imagen para no perturbar la clasificación de estas imágenes.

Para este proceso, se usan operadores morfológicos como la dilatación, erosión, apertura, cierre, complemento, diferencia, reflexión, entre otros. Estos corrigen los errores presentes en la imagen segmentada que interfieren en el proceso de clasificación. Pueden ser aplicados como funciones para extraer o realzar algunas de las características presentes en la imagen.

A continuación, se presentan algunas definiciones básicas dentro de la morfología:

Se posee dos conjuntos bidimensionales A y B, con componentes $a = (a_1, a_2)$ y $b = (b_1, b_2)$ respectivamente. La reflexión de un conjunto A se define en la Ecuación 20:

$$\hat{B} = \{x | x = -a, \text{ para } a \in A\} \quad (20)$$

El complemento de un conjunto se denota de la Ecuación 21:

$$A^c = \{x | x \notin A\} \quad (21)$$

3.1.1. Dilatación

De acuerdo con (Gonzalez & Woods, 2002), esta operación morfológica consiste en obtener la reflexión del conjunto B —en la mayoría de casos está representado como un elemento estructurado de 4 u 8 puntos de conectividad— véase Figura 58(b) sobre su origen, desplazándose x posiciones, de tal manera que esta reflexión y el conjunto A solapen en al menos un elemento distinto de cero. De acuerdo con lo anterior, la dilatación se representa en la Ecuación 22:

$$A \oplus B = \left\{ x \mid \left[(\hat{B})_x \cap A \right] \subseteq A \right\} \quad (22)$$

$$\hat{B} = B$$

A continuación, en la Figura 58 se muestra un ejemplo del proceso de dilatación:

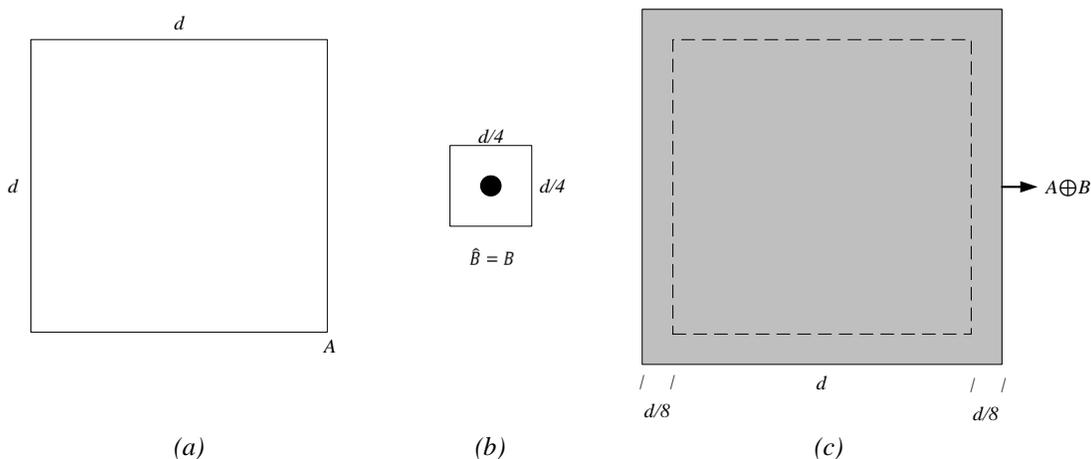


Figura 58. (a) Conjunto bidimensional A. (b) Elemento estructurado B, cuatro veces más pequeño que el conjunto A. (c) Resultado del proceso de la dilatación entre el conjunto A y B.

Fuente (Gonzalez & Woods, 2002)

3.1.2. Erosión

Esta función corresponde a reducir x puntos en los bordes del conjunto A, de acuerdo con los puntos x pertenecientes al elemento estructurado B. Este operador se representa en la Ecuación 23:

$$A \ominus B = \{x \mid (B)_x \subseteq A\} \quad (23)$$

Con el fin de entender mejor este concepto, se presenta la Figura 59 donde se ilustra el funcionamiento de este operador morfológico:

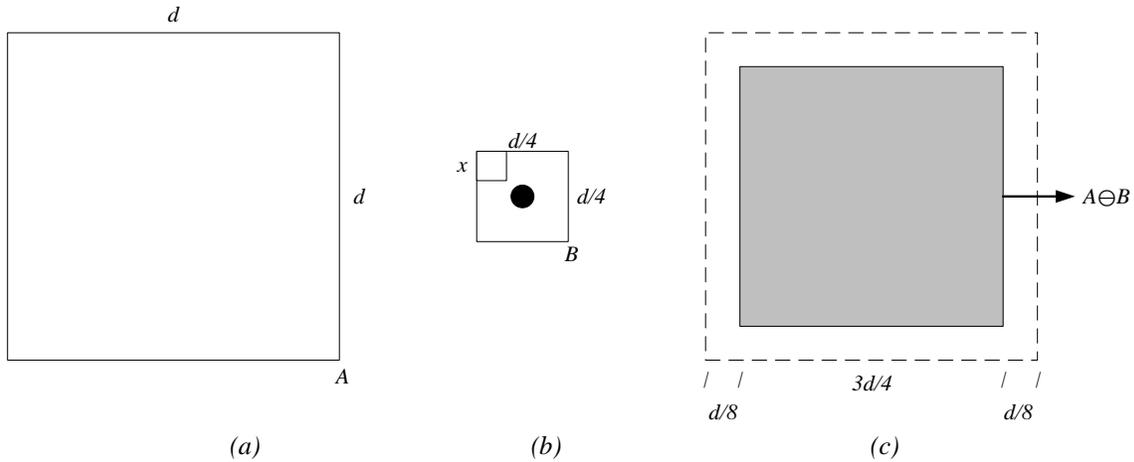


Figura 59. (a) Conjunto original A. (b) Elemento de estructura B. (c) Resultado de la erosión entre el conjunto A y el elemento estructurado B.

Fuente: (Gonzalez & Woods, *Digital Image Processing*, 2002)

3.1.3. Apertura y cierre

La operación morfológica de apertura consiste en suavizar el contorno de la imagen, y así eliminar franjas que interfieran con dicho contorno y protuberancias delgadas. El operador de cierre también permite suavizar partes del contorno, pero, en el caso contrario de la apertura, el algoritmo integra separaciones estrechas y entrantes delgados profundos, para eliminar huecos pequeños y rellenar vacíos en el contorno.

La función que representa el operador de apertura se expresa como $A \circ B$, donde el resultado de la erosión entre un conjunto A y un elemento estructurado B se dilata con el elemento estructurado B, en la Ecuación 24:

$$A \circ B = (A \ominus B) \oplus B \quad (24)$$

El operador de cierre consiste en dilatar el conjunto A y el elemento estructurado B. Luego con este resultado, se erosiona nuevamente con el elemento estructurado B. El cierre se ilustra en la Ecuación 25:

$$A \cdot B = (A \oplus B) \ominus B \quad (25)$$

De acuerdo con lo anterior, se usan los comandos brindados por las librerías de OpenCV (“*dilate()*” y “*erode()*”), con el fin de realizar el operador morfológico de cierre, el cual permite mejorar la imagen segmentada, y así disminuir el ruido (puntos negros) presente en dicha imagen. En la Figura 60, se aprecia los resultados de los procesos de erosión, dilatación y cierre, realizados a la imagen segmentada para las prendas camisa, camiseta y buso.

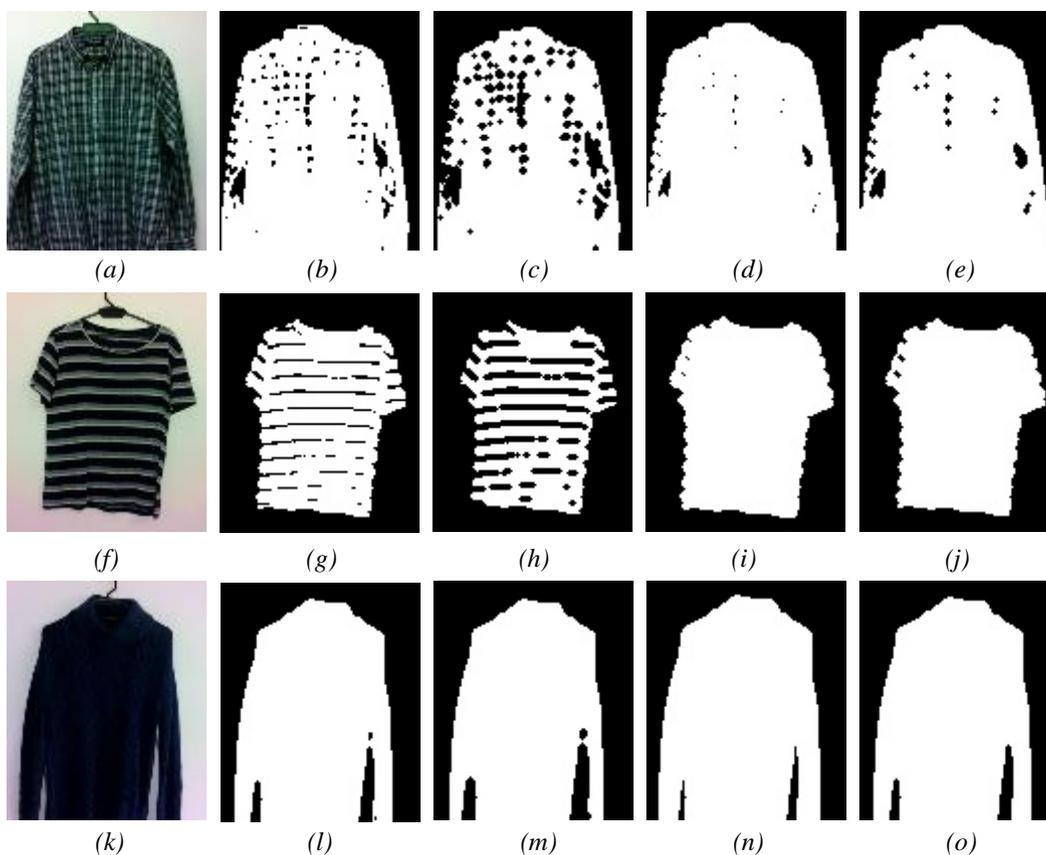


Figura 60. (a, f, k) Imagen original de prenda. (b, g, l) Resultados de la segmentación. (c, h, m) Resultado del operador erosión de la segmentación. (d, i, n) Resultados del operador dilatación de la segmentación. (e, j, o) Resultados del operador cierre de la segmentación.

Fuente autores.

Se puede ver que en los literales (e), (j) y (o) de la Figura 60, ha disminuido la cantidad de puntos negros con respecto a la segmentación original, mejorando la calidad de la imagen. Sin embargo, existen resultados en la segmentación donde aún posee puntos negros, como es el caso de la sección(e). Esta imagen no es suficiente para realizar una representación de firmas, ya que estas son sensibles a los ruidos y probablemente tenga un resultado no deseado. Para solucionar este problema, se aplica el método de relleno de regiones mediante el comando de OpenCV “*findContours*” y “*drawContours*”. La primera función se encarga de hallar los diferentes contornos que hay dentro de la imagen segmentada: tómease como referencia el caso de la Figura 61 (a), donde se tiene una imagen segmentada. Al aplicar esta función, los resultados que se obtiene se presentan en la Figura 61 (b). Allí aparecen los diferentes contornos que posee la imagen. La segunda función permite, con la información obtenida de la primera función acerca de los contornos, seleccionar y rellenar el contorno más grande dentro de la imagen para así obtener la figura de la prenda, tal cual como se ilustra en la Figura 61 (c).

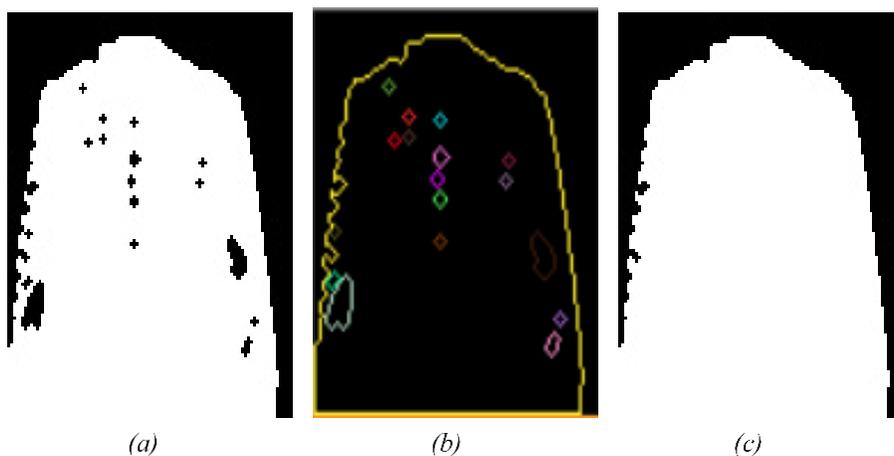


Figura 61. (a) Imagen de operador cierre. (b) Resultados de los contornos que existen en la segmentación de operador cierre. (c) Resultado del relleno de regiones.

Fuente Autores.

3.2. Extracción de características

Consiste en definir los datos o valores que representan el conjunto de rasgos internos y externos del objeto segmentado. Ejemplo de características que puede cuantificar la imagen segmentada son la forma, la textura, el contorno, el color, entre otras.

Para realizar esta extracción, se divide el proceso en dos etapas: la primera consiste en el proceso de representación de los límites o contornos del objeto segmentado. En la segunda etapa se aplican los descriptores, el cual están representados por dos clases: Descriptores de límite *-Boundary Descriptors-* y descriptores regionales *-Regional Descriptors-*.

En los esquemas de representación, existe diversos métodos que identifican los datos que más tarde serán utilizados para el proceso de los descriptores. Para el proyecto, se implementan el siguiente esquema de representación:

- Firmas (*Signatures*): Es una representación unidimensional que calcula la distancia entre el centro de masa y el contorno del objeto. Normalmente se conoce como el radio de la figura. Además, la firma se comporta como una función del ángulo, ya que cada uno de estos presenta su valor de radio de acuerdo con la forma del objeto. Por ejemplo, el caso de un círculo y un cuadrado con su centro de masa, radio y firma respectivos, como se ve en la Figura 62 (Gonzalez & Woods, 2002).

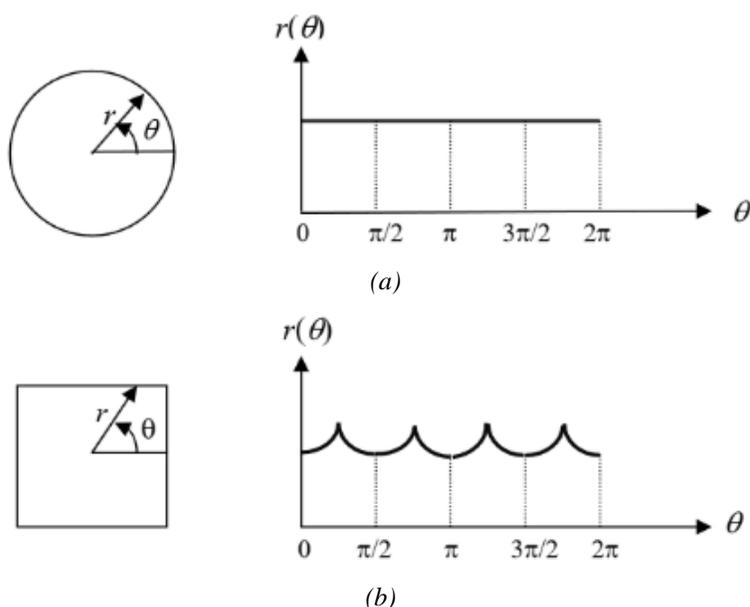


Figura 62. (a) Firma del círculo, (b) Firma del cuadrado.
Fuente (Gonzalez & Woods, 2002).

La Figura 62 indica que cada grafica de bidimensional representa su comportamiento de contorno. Por lo tanto, el eje X representa el ángulo de 0 a 360°, mientras que el eje Y presenta el radio de la figura. Sin embargo, este método es robusto ante la traslación y no ante la rotación y ni ante el escalado.

- Cerco Convexo CH (*Convex Hull*): Es una herramienta que analiza los lados del contorno para reducir la complejidad de este, especialmente, con los objetos segmentados que presenta concavidades en su forma. En otras palabras, este método cubre todos los puntos del contorno de la figura o produce un conjunto convexo, donde se asume la unión de la distancia entre un punto a otro punto en trayectoria recta, formando un polígono aproximado de ella. En este punto, se encuentran dos conjuntos: el primero representa el polígono aproximado S y H, el cual representa al cerco convexo. El segundo se conoce como deficiencia convexa, el cual se define como el conjunto de la diferencia H-S. Como se ve en la Figura 63, es posible hallar el área del conjunto S y el número de deficiencia convexa de la figura que reconoce la descripción de ella (Gonzalez & Woods, 2002).

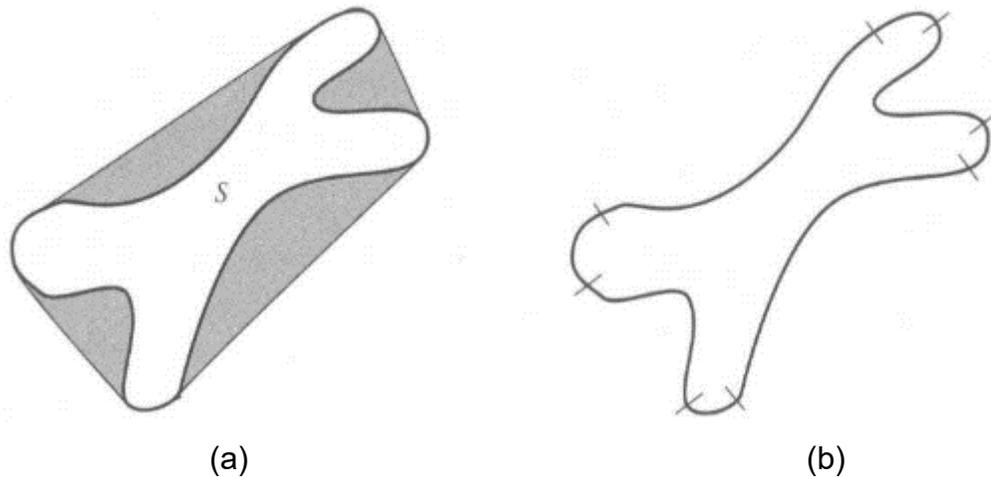


Figura 63. (a) Conjunto S y Conjunto $H-S$ (sombreada); (b) Contorno dividido.
Fuente (Gonzalez & Woods, *Digital Image Processing*, 2002).

Después de realizar la representación, se procede con el proceso de descripción:

- Momentos (*Moments*):

Este método pertenece a los descriptores de regiones con base matemática que proporcionan información sobre las diferentes estructuras geométrica. Esto quiere decir que, los datos obtenidos de esta clase permiten elaborar un reconocimiento de formas, e incluso la clasificación de objetos, la reconstrucción, el acoplamiento de escenas, reconocimiento de expresiones faciales y el análisis de texturas (Sánchez Lopez, 2011).

Matemáticamente, los momentos de orden $(p + q)$ de una función continua bidimensional $I(x, y)$ están definidos por la Ecuación 26 (Gonzalez & Woods, 2002):

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q I(x, y) dx dy \quad (26)$$

$p, q = 0, 1, 2, 3 \dots$

Para el caso de una imagen digital, los momentos se definen mediante en la Ecuación 27:

$$m_{pq} = \sum_x \sum_y x^p y^q I(x, y) \quad (27)$$

Por otra parte, para hallar los momentos centrales, se usa la Ecuación 28:

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^p (y - \bar{y})^q I(x, y) dx dy \quad (28)$$

Donde los valores de \bar{x} y \bar{y} se definen en la Ecuación 29:

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (29)$$

La Ecuación 28 puede ser reemplazada por la Ecuación 30, la cual va dirigida específicamente a imágenes digitales:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x, y) \quad (30)$$

Por lo tanto, las siguientes ecuaciones expresan los momentos centrales desde el orden cero hasta el orden tres (0,1,2,3):

$$\begin{aligned} \mu_{00} &= m_{00} \\ \mu_{10} &= \mu_{01} = 0 \\ \mu_{20} &= m_{20} - \bar{x}m_{10} \\ \mu_{02} &= m_{02} - \bar{y}m_{01} \\ \mu_{11} &= m_{11} - \bar{x}m_{01} = m_{11} - \bar{y}m_{10} \\ \mu_{21} &= m_{21} - 2\bar{x}m_{11} - \bar{y}m_{20} + 2\bar{x}^2m_{01} \\ \mu_{12} &= m_{12} - 2\bar{y}m_{11} - \bar{x}m_{02} + 2\bar{y}^2m_{10} \\ \mu_{30} &= m_{30} - 3\bar{x}m_{20} + 2\bar{x}^2m_{10} \\ \mu_{03} &= m_{03} - 3\bar{y}m_{02} + 2\bar{y}^2m_{01} \end{aligned} \quad (31)$$

Para aplicar el método de Firmas en el algoritmo, se hace uso de uno de los comandos ya preestablecidos por la librería de OpenCV “*findContours*”. Este comando, como se explica en el subtema anterior, detecta los contornos de todos los elementos que se encuentran en la imagen segmentada. El algoritmo de este comando consiste en encontrar las coordenadas de cada uno de los contornos y con estos datos, la representación de firmas puede aplicar el método de triángulos

cuadrados para determinar el radio y el ángulo de los puntos del contorno, con respecto al centro de masa. Su expresión se define en la Ecuación 32.

$$\begin{aligned} \text{radio} &= \sqrt{(y_p - \bar{y})^2 + (x_p - \bar{x})^2} \\ \theta &= \tan^{-1} \left(\frac{y_p - \bar{y}}{x_p - \bar{x}} \right) \end{aligned} \quad (32)$$

Donde

$(\bar{x}, \bar{y}) = \text{Punto de centroide}$

$(x_p, y_p) = \text{Punto de coordenada de contorno}$

De la ecuación 32 se obtiene una gráfica que ayudará a comprender y diferenciar los objetos que se examinen. En la Figura 64 se observa un ejemplo de esto, donde se tiene una imagen segmentada de una camiseta (véase Figura 64 (a)) y una pantaloneta (véase Figura 64 (c)). Se observa que el área en blanco dentro del contorno de la camiseta es mayor que la de la pantaloneta. Esto genera que el rango de amplitud del radio para la camiseta oscile entre 35 y 65, en cambio para la pantaloneta esta se encuentra ubicada entre los valores de 20 y 50 aproximadamente. En la Figura 64 (b) y (d) se pueden comprobar estos resultados. Con estos valores obtenidos se puede lograr una diferencia entre las prendas de vestir, exceptuando aquellas que poseen una forma muy similar como lo son las camisas y las chaquetas.

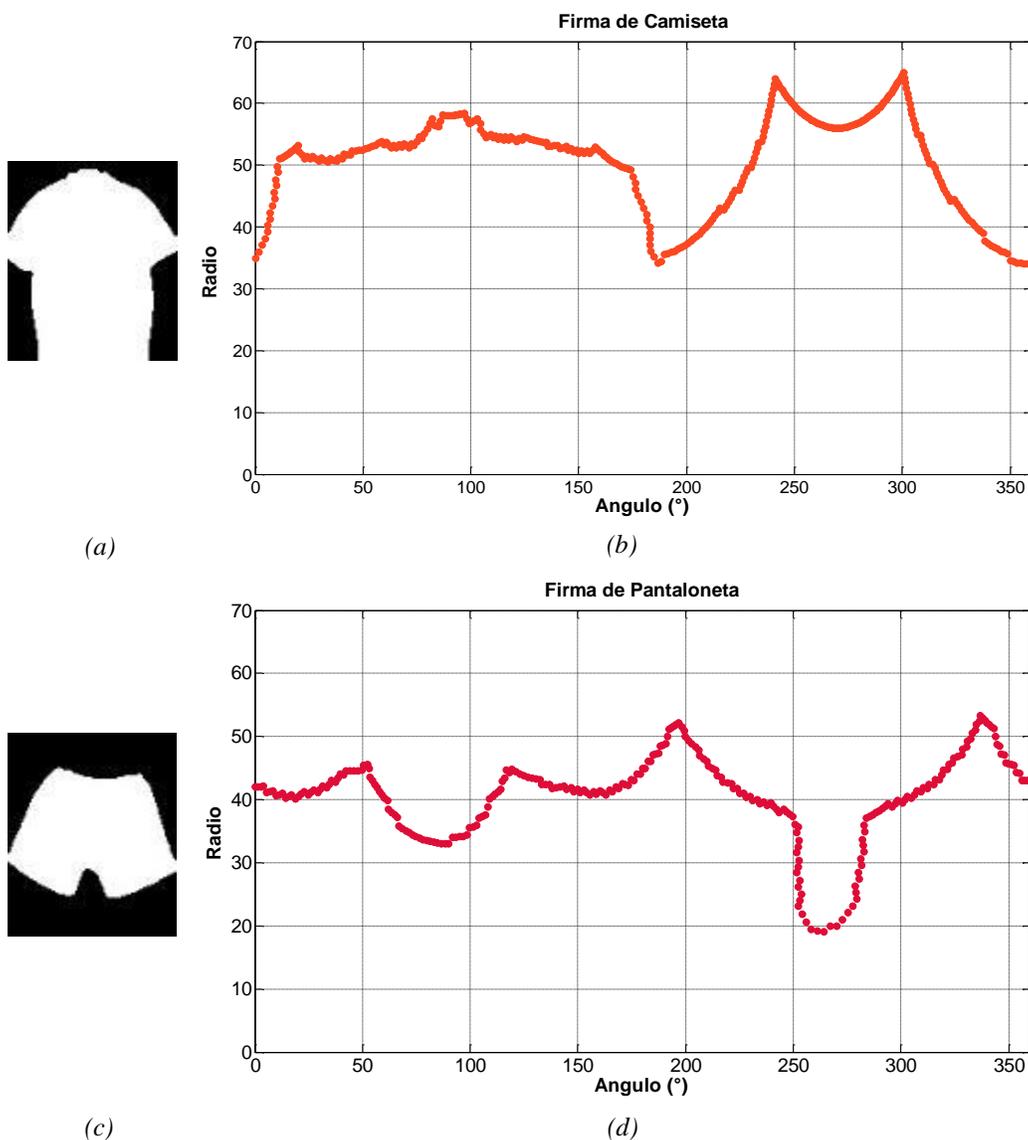


Figura 64. (a) Imagen segmentada de la camiseta, (b) Firma de la camiseta segmentada, (c) Imagen segmentada de la pantaloneta, (d) Firma de la pantaloneta segmentada.

Fuente Autores

Para el caso de los Momentos se usa el comando de OpenCV, “*moments()*” con el fin obtener los valores de orden cero, uno, dos y tres de los momentos. Así mismo, se obtienen los datos de los momentos centrales y la normalización de los mismos. Con esta información se representa como la forma de la prenda. La Tabla 5, Tabla 6 y Tabla 7 muestran los resultados de la camiseta y pantaloneta de la Figura 64:

Tabla 5. Resultados de variables “Momentos” para las prendas camiseta y pantaloneta.

	m_{00}	m_{01}	m_{02}	m_{03}	m_{10}	m_{11}	m_{12}	m_{20}	m_{21}	m_{30}
Camiseta	8103,5	527857	4,13E+07	3,57E+09	430533	2,83E+07	2,24E+09	2,70E+07	1,81E+09	1,87E+09
Pantaloneta	5258,5	313055	2,04E+07	1,42E+09	263038	1,57E+07	1,03E+09	1,62E+07	9,82E+08	1,11E+09

Fuente autores.

Tabla 6. Resultados de variables “Momentos centrales” para las prendas camiseta y pantaloneta.

	mu_{02}	mu_{03}	mu_{11}	mu_{12}	mu_{20}	mu_{21}	mu_{30}
Camiseta	6,92E+06	-2,06E+07	299652	3,98E+06	4,13E+06	1,66E+07	-3,96E+06
Pantaloneta	1,77E+06	-3,47E+06	82809,7	1,19E+06	3,04E+06	9,39E+06	-1,06E+06

Fuente autores.

Tabla 7. Resultado de variables “Momentos normalizados” para las prendas camiseta y pantaloneta.

	nu_{02}	nu_{03}	nu_{11}	nu_{12}	nu_{20}	nu_{21}	nu_{30}
Camiseta	0,105447	-0,00347916	0,00456322	0,00067251	0,0628603	0,00280115	-0,00067050
Pantaloneta	0,0639721	-0,0017317	0,00299473	0,000594311	0,110084	0,0046836	-0,000526751

Fuente autores.

Con los datos de la Firma y de Momentos anteriormente mostrados, se puede analizar y determinar una forma de distinguir los tipos de prendas y así, clasificarlas.

3.3. Análisis de datos obtenidos por los descriptores y los esquemas de representación

Para desarrollar el proceso de clasificación, primero se analiza los valores obtenidos por los descriptores, con el fin de seleccionar aquellos donde se logra obtener una diferencia entre las clases. Luego de seleccionarlos, se grafican estos valores para conocer su comportamiento y de allí definir qué tipo de modelo de clasificación se puede implementar para separar las clases, que luego será llevados al lenguaje de programación.

Para la categoría del hombre:

Los pantalones y pantalonetas poseen diferentes formas de doblarse a la hora de colgarse, como se ve la Figura 65. Al tener en cuenta esto, se programa el sistema de tal manera que, usando el método de representación “Firmas”, solo se enfoque en analizar dichas prendas que se encuentren en forma “Abierta”, tal como se ilustra en la Figura 65 (a). Esto permite que sea sencillo determinar el comportamiento de la prenda al usar este método. Véase el caso de la gráfica de la Figura 64 (d), donde se muestra que el valor mínimo del radio de la pantaloneta es menor que 30. Además su ángulo oscila entre 250 y 300 grados, mientras que la camiseta presente en la Figura 64 (a), posee tres valores mínimos, las cuales están sobre 30 y sus ángulos se ubican en tres posiciones diferentes: la primera se encuentra entre 0 y 5 grados, la segunda está en 150 y 200, y la tercera en 350 y 360 grados (Figura 64 (b)).



(a)

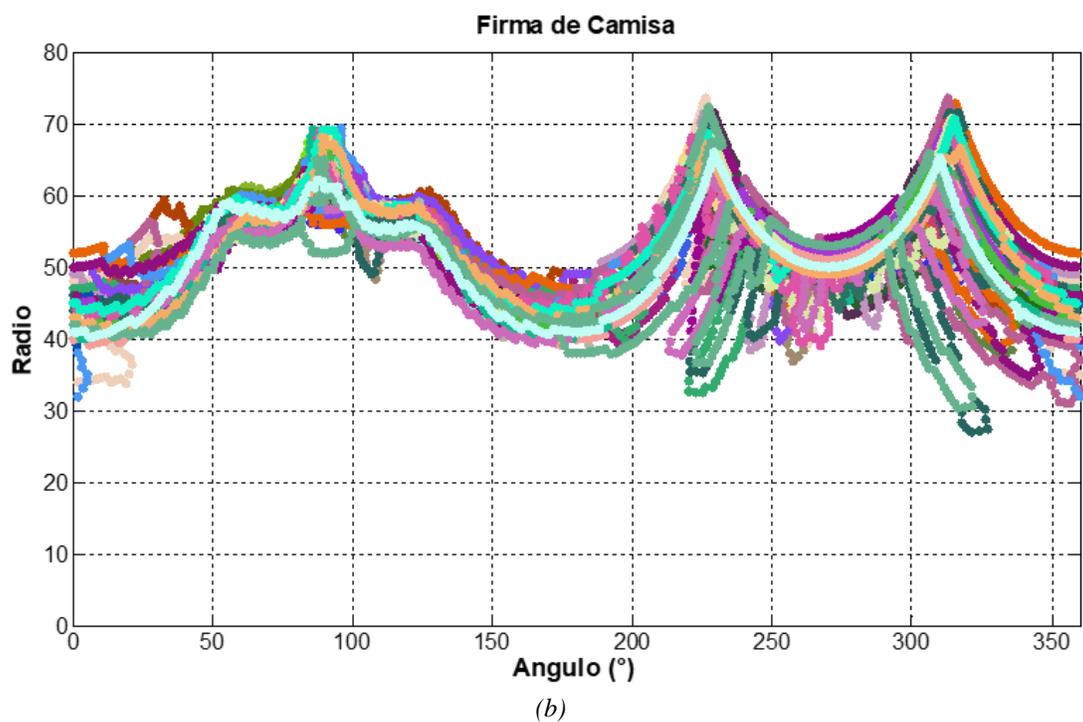
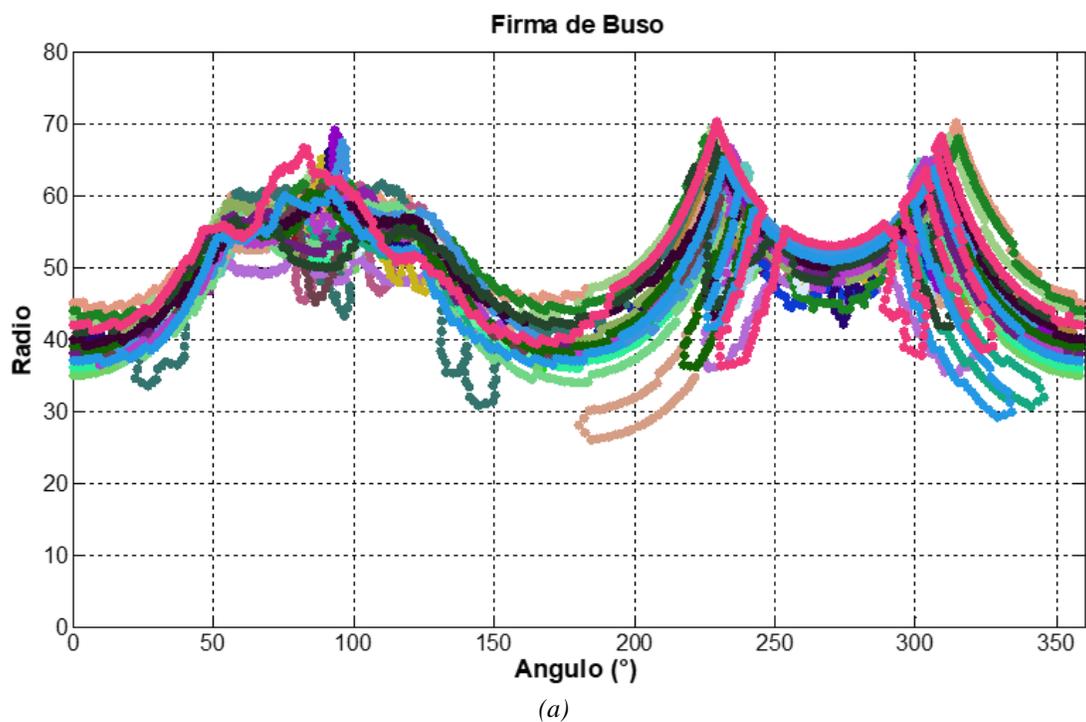
(b)

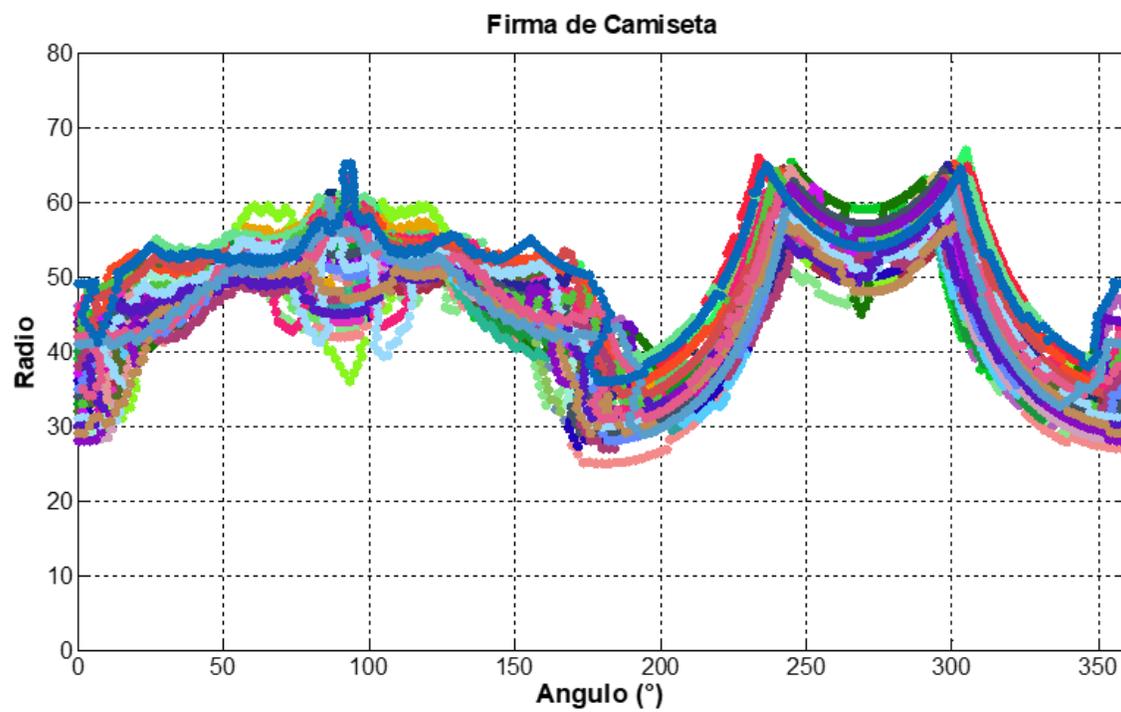
Figura 65. (a) Pantalón “Abierta”, (b) Pantalón “Cerrada”.

Fuente Autores

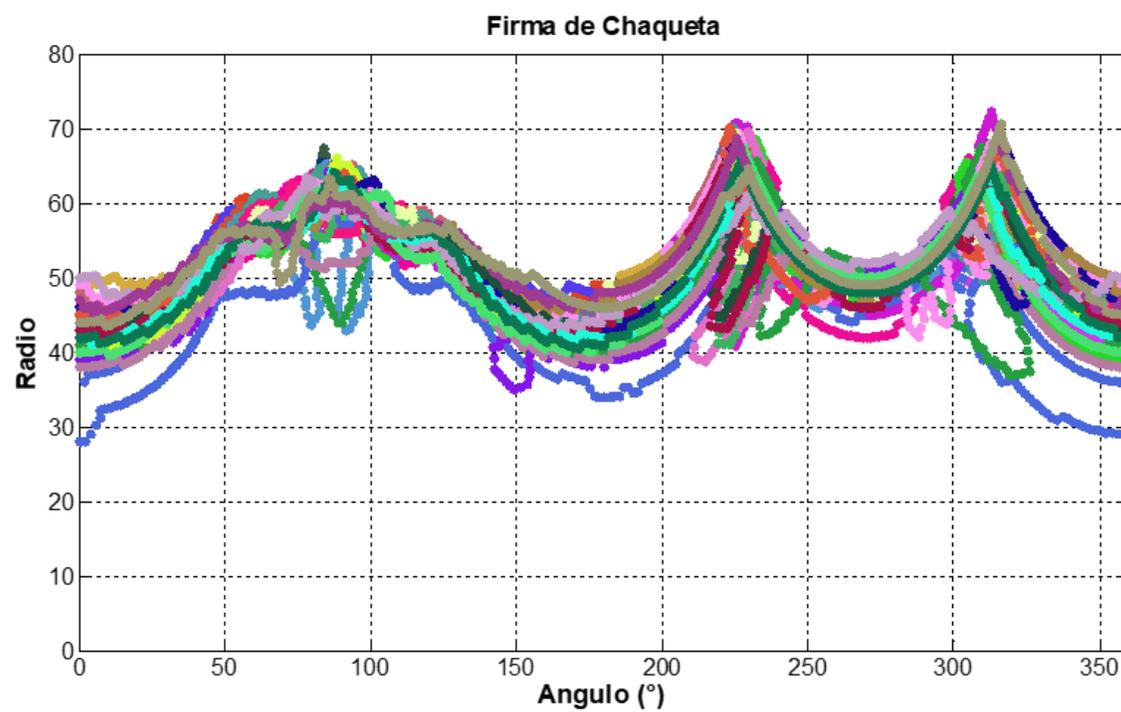
En conclusión, los valores mínimos en la mayoría de los pantalones y pantalonetas que poseen la forma “Abierta” son menores que 30 en su radio, con ángulos de 250° a 300° como lo muestra la Figura 66(f) y (g). Estas prendas pueden clasificarse dentro del grupo de “No torso”. Es decir que, a esta clase pertenecen las prendas que se encuentran debajo de la cintura. De otro lado, las camisas,

camisetas, busos y chaquetas son las prendas que están encima de la cadera y, por lo tanto, pertenecen al grupo del “Torso”.

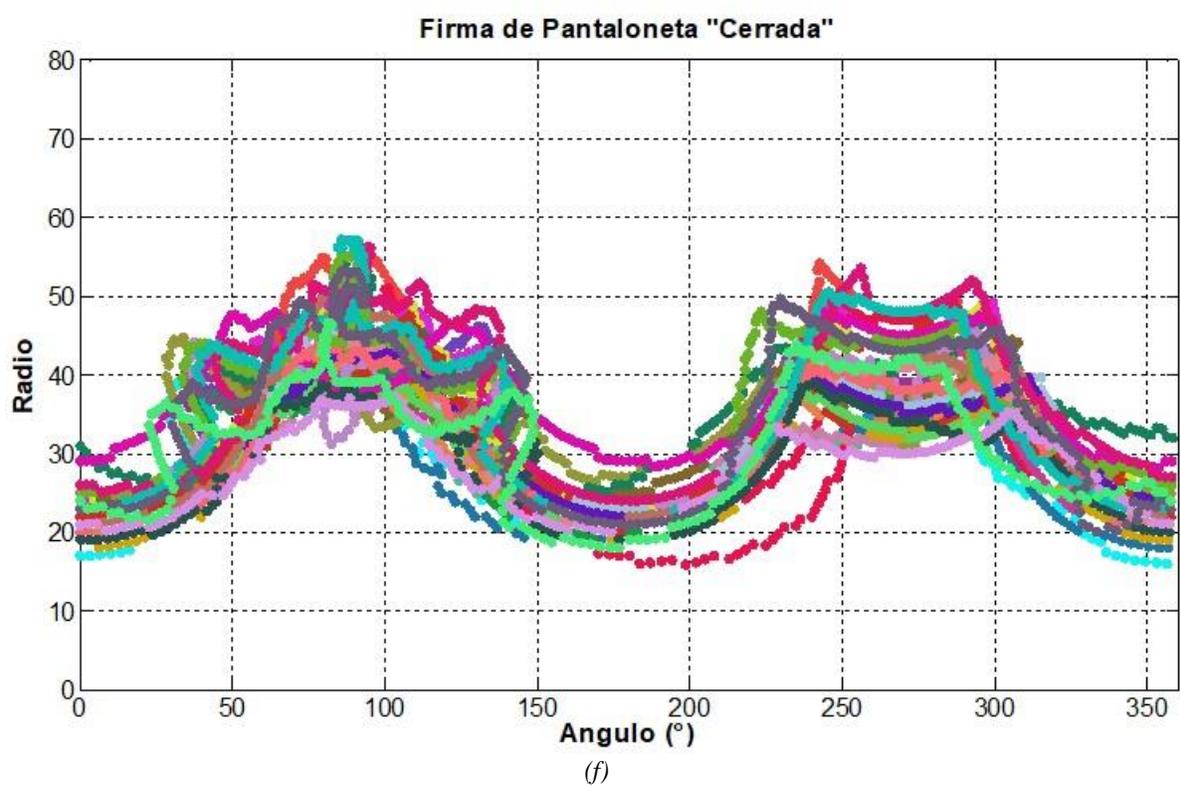
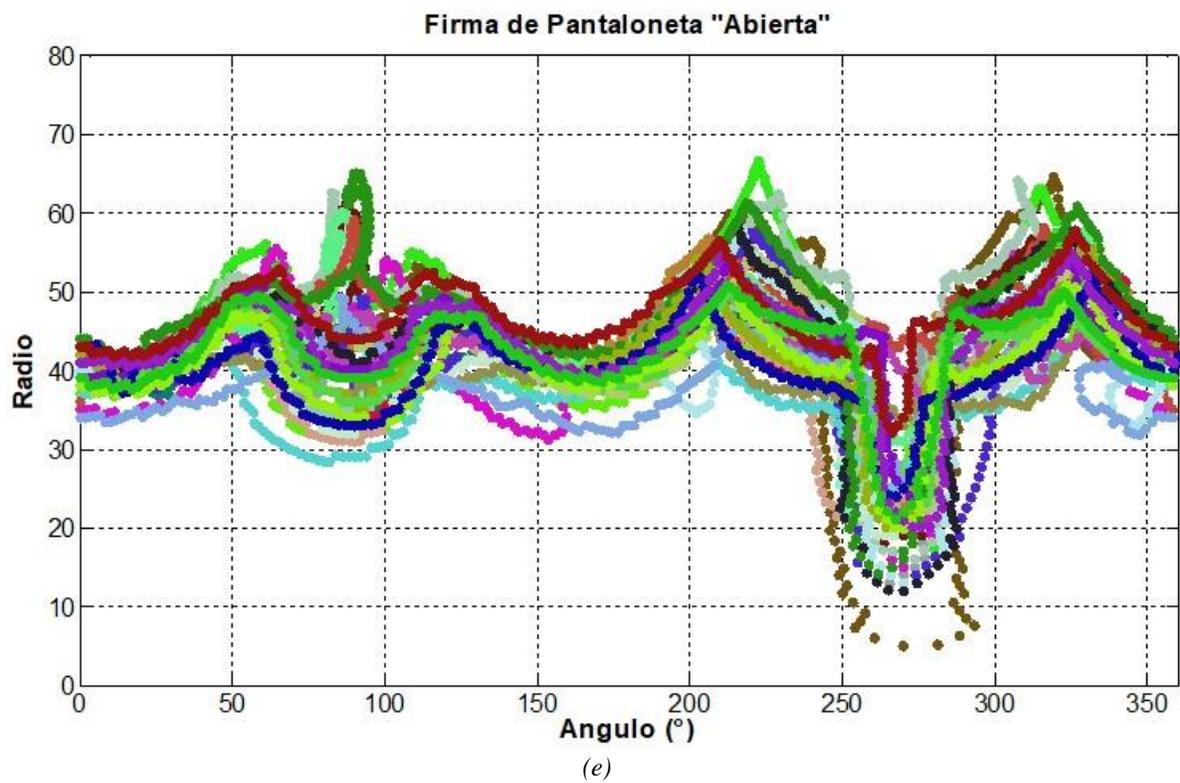




(c)



(d)



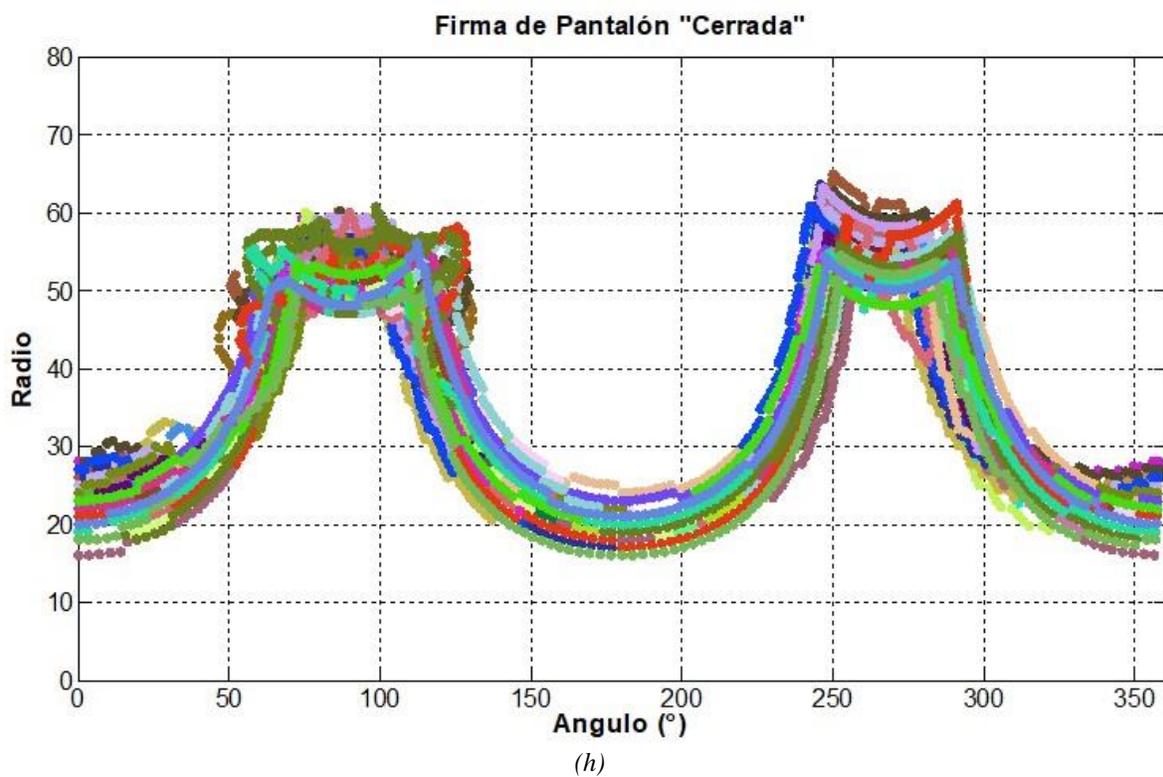
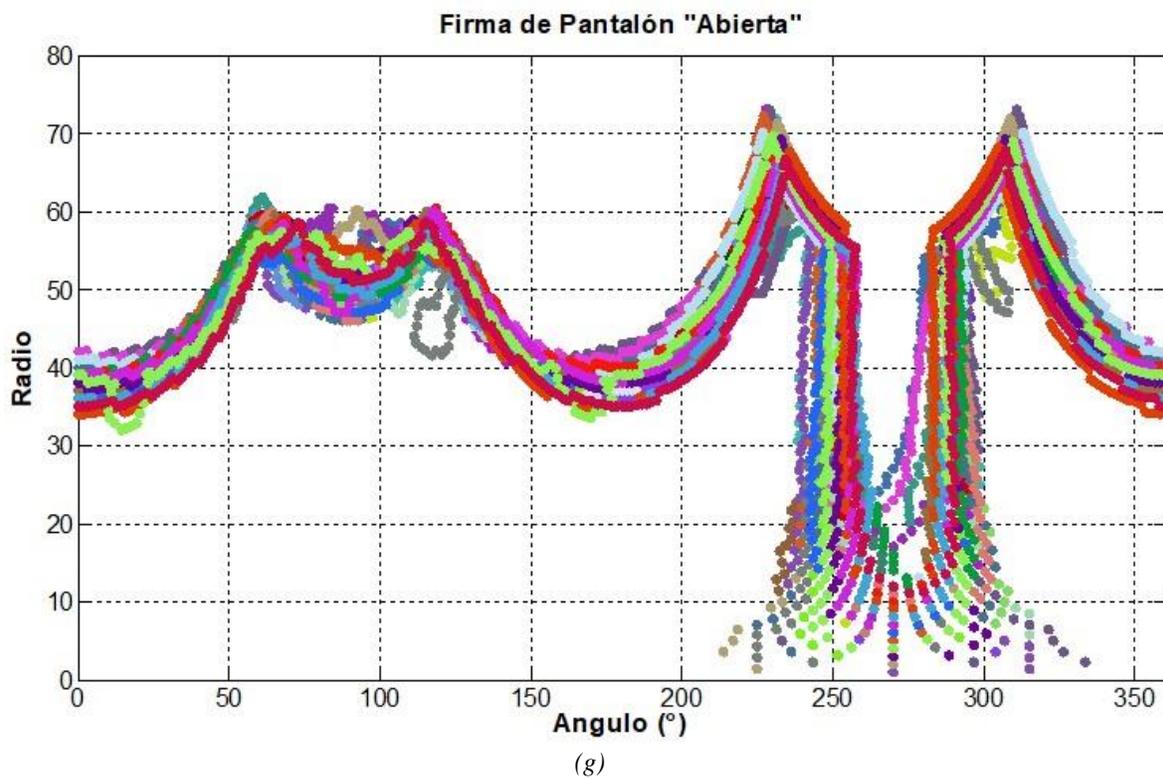


Figura 66. Resultados de grafica de Firmas en Matlab: (a) Buso, (b) Camisa, (c) Camiseta, (d) Chaqueta, (e) Pantalóneta "Abierta", (f) Pantalóneta "Cerrada", (g) Pantalón "Abierta", (h) Pantalón "Cerrada".

Fuente Autores

Sin embargo, en las secciones (f) y (h) de la Figura 66 se presenta una forma distinta a la “Abierta” de pantalón y pantaloneta. Por ende, este caso no se cumple con la condición mencionada anteriormente de “No torso” y, en estas figuras se aplica el método de “Momentos” del descriptor regional. En otras palabras, la forma de “Abierta” y “Cerrada” de los pantalones y pantalonetas deben clasificarse dentro de “No torso”. Para resolver este inconveniente de la forma de “Cerrada”, se aplican los momentos, obteniendo los valores que poseen menores de áreas de la imagen segmentada y son angostas con respecto las demás de prendas.

El **ANEXO A** se muestra las tablas de los resultados del comando OpenCV, “*moments()*” de las prendas sobre las que se ha realizado una sesión de fotos en un ambiente controlado.

Para esta categoría, se enfoca en las variables m_{01} y mu_{02} del operador Momentos y el valor del radio obtenido por el método de representación Firmas. Para este último, solo se seleccionan los valores mínimos del radio que se encuentren entre los ángulos 245° y 295° . Con estos datos, se logra formar una gráfica de tres dimensiones, la cual permite observar el comportamiento de los patrones de las prendas, y así buscar una condición donde se puedan distinguir las prendas entre “Torso” y “No torso”. Sin embargo, hay que tener en cuenta que el rango del área tiene que cumplir con unos estándares aproximados (1900 a 9900) pixeles. Para realizar esta condición se utiliza la variable m_{00} de los momentos, el cual mostrará si el área de la imagen segmentada cumple o no cumple con este parámetro.

La Figura 67 muestra el resultado de la distribución de los seis distintos colores, los cuales representan los seis tipos de prendas: buso, camisa, camiseta, chaqueta, pantaloneta y pantalón. También se observa que la mayoría de los colores cian y magenta se encuentran en una zona

alejada de los demás de colores, mientras que, el rojo, verde, azul y amarillo se encuentran unidos entre ellos.

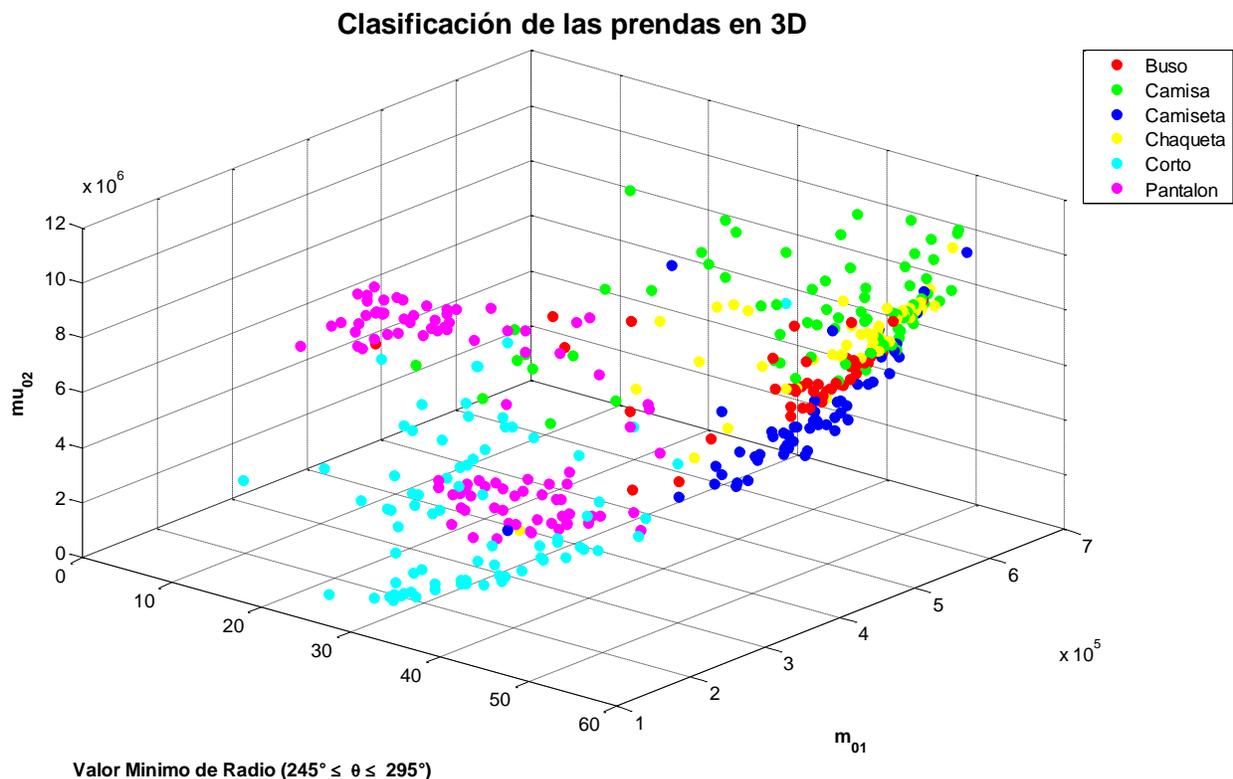
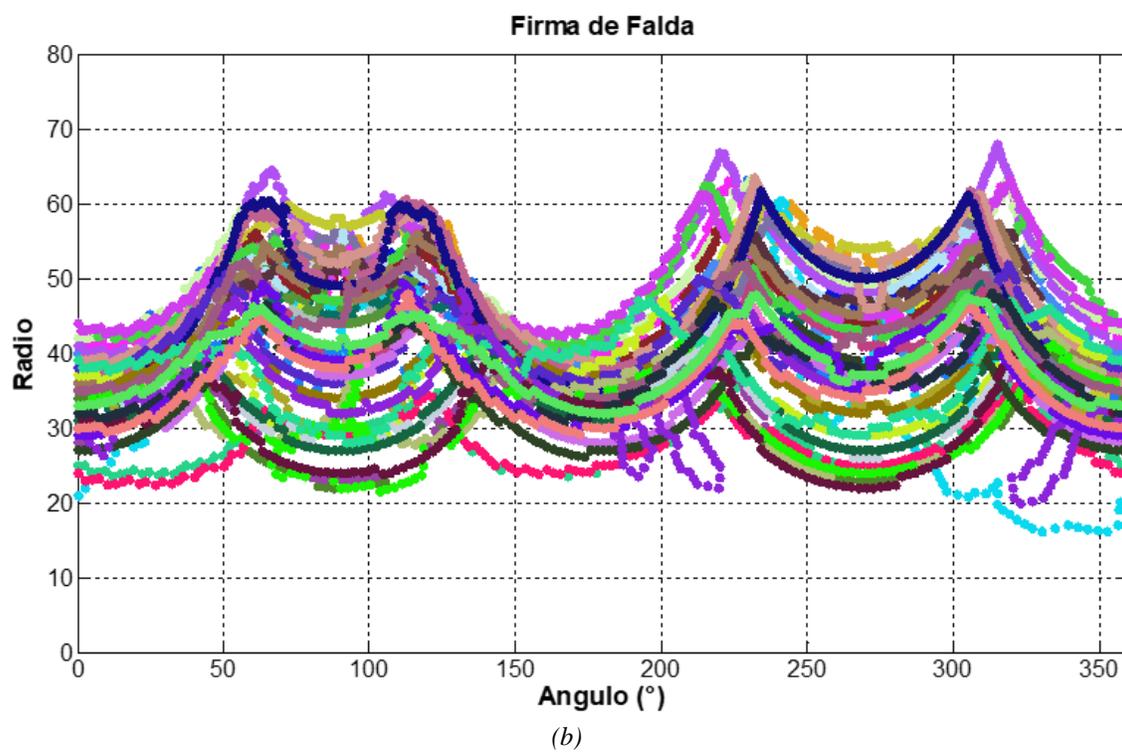
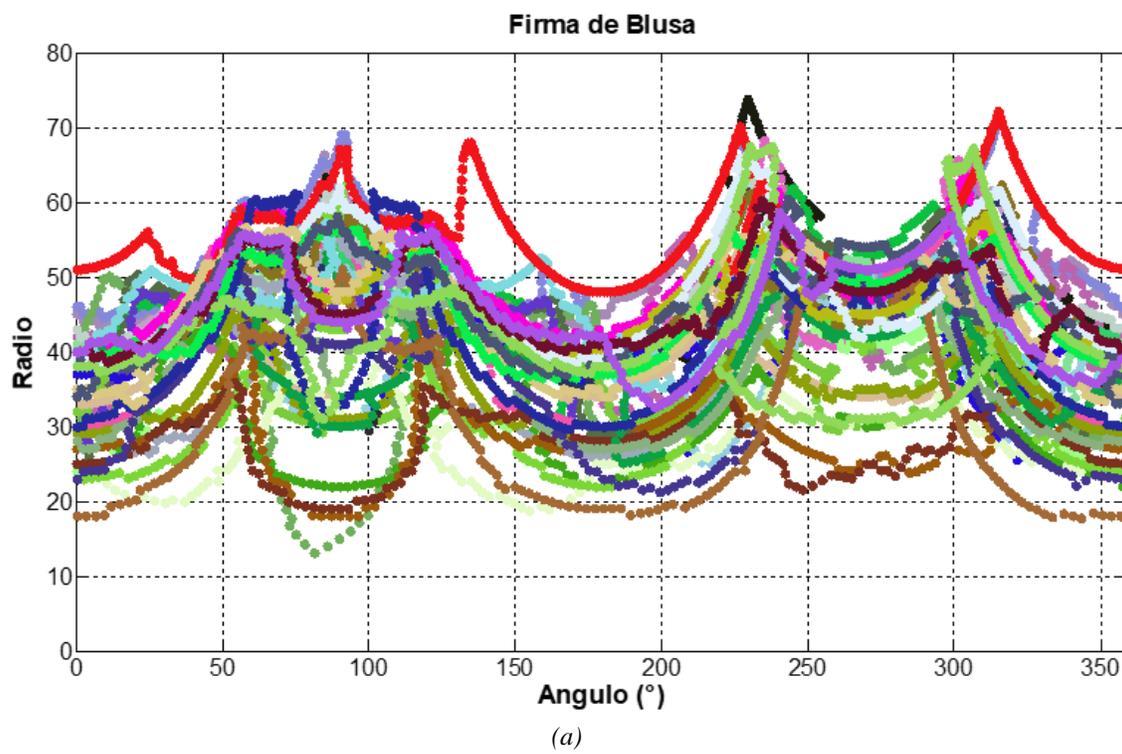
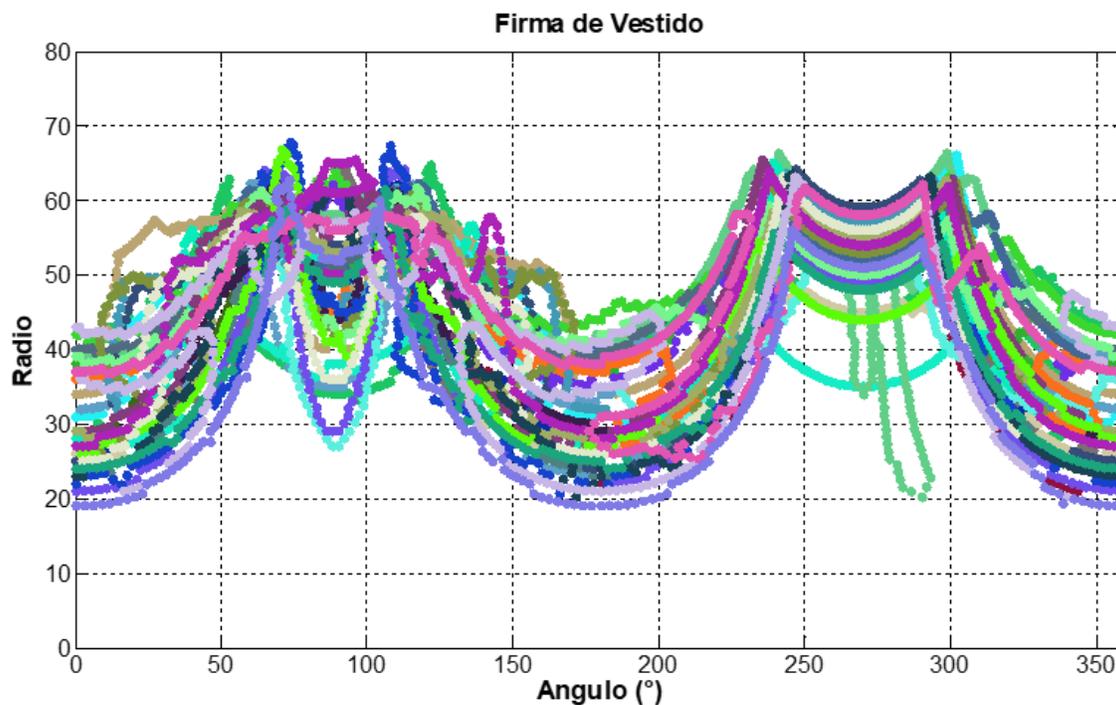


Figura 67. Resultados de distribución de los valores de las prendas para hombre en tres dimensiones.
Fuente Autores

Para la categoría de mujer:

Las prendas de mujer presentan un poco de complejidad en el reconocimiento de prendas debido a que estas poseen diferentes estilos de diseño en las blusas, las faldas y los vestidos. Esto hace que las firmas varíen su comportamiento (Véase a la Figura 68). En cambio, las prendas del hombre manejan una sola línea de estilo —formas, diseños y composición de dichas prendas—, por ende, sus firmas son similares.



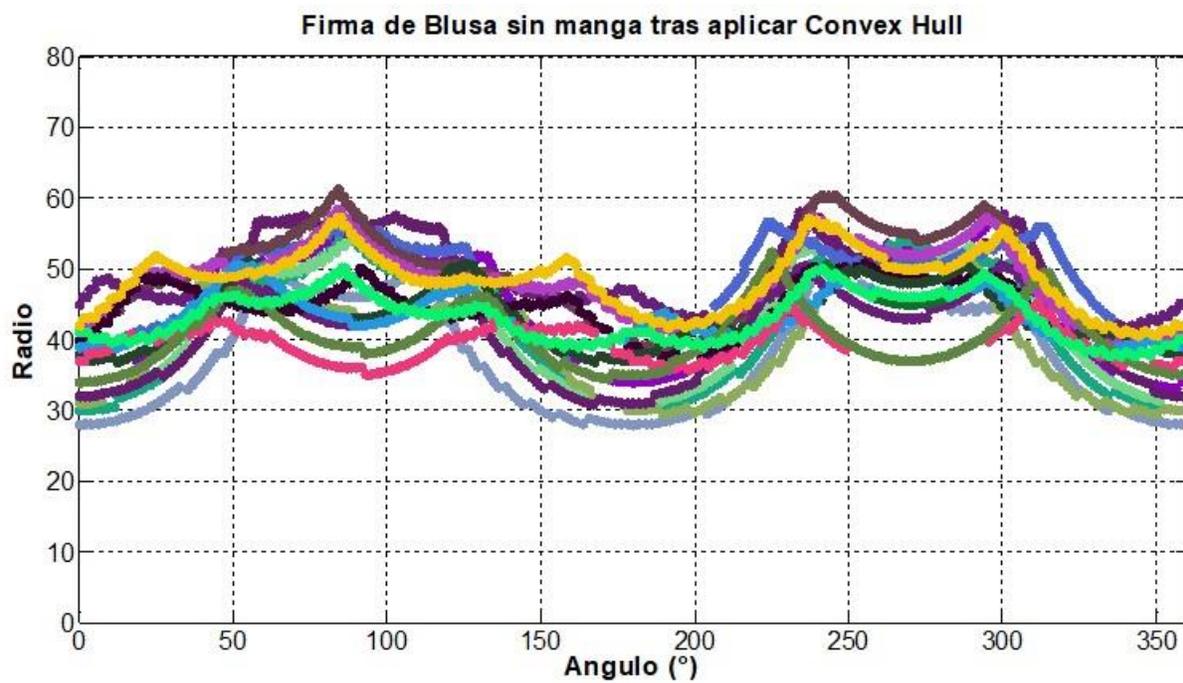


(c)

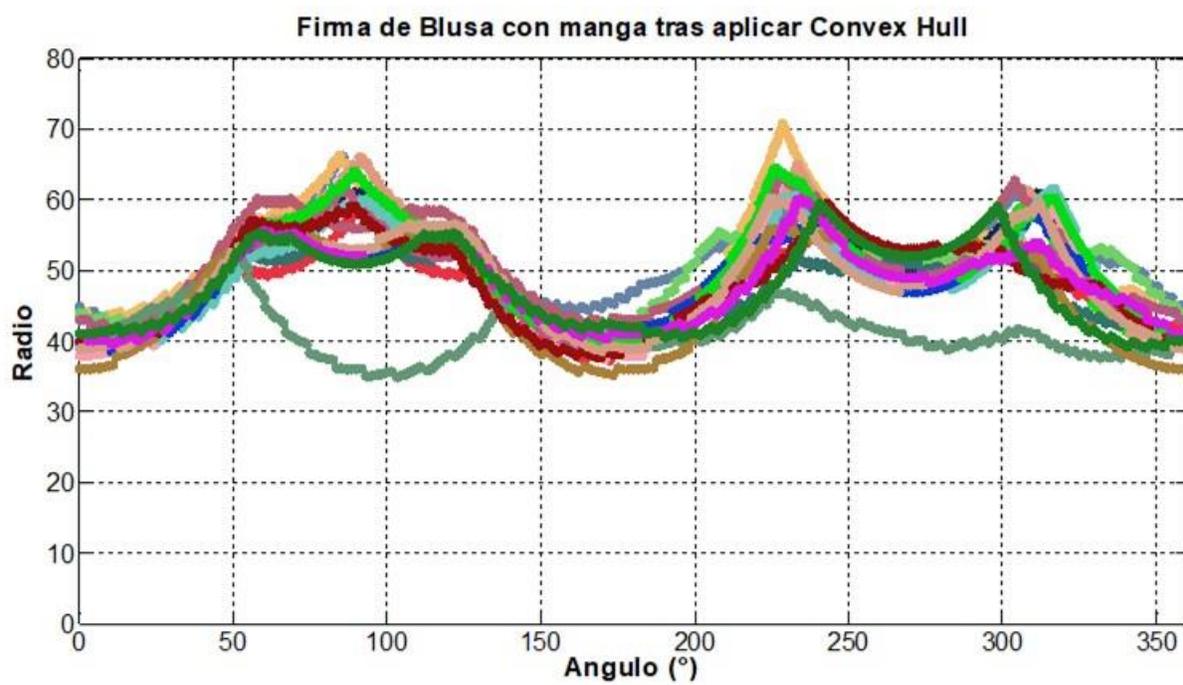
Figura 68. Resultados de grafica de Firmas en Matlab: (a) Blusa, (b) Falda, (c) Vestido.

Fuente Autores

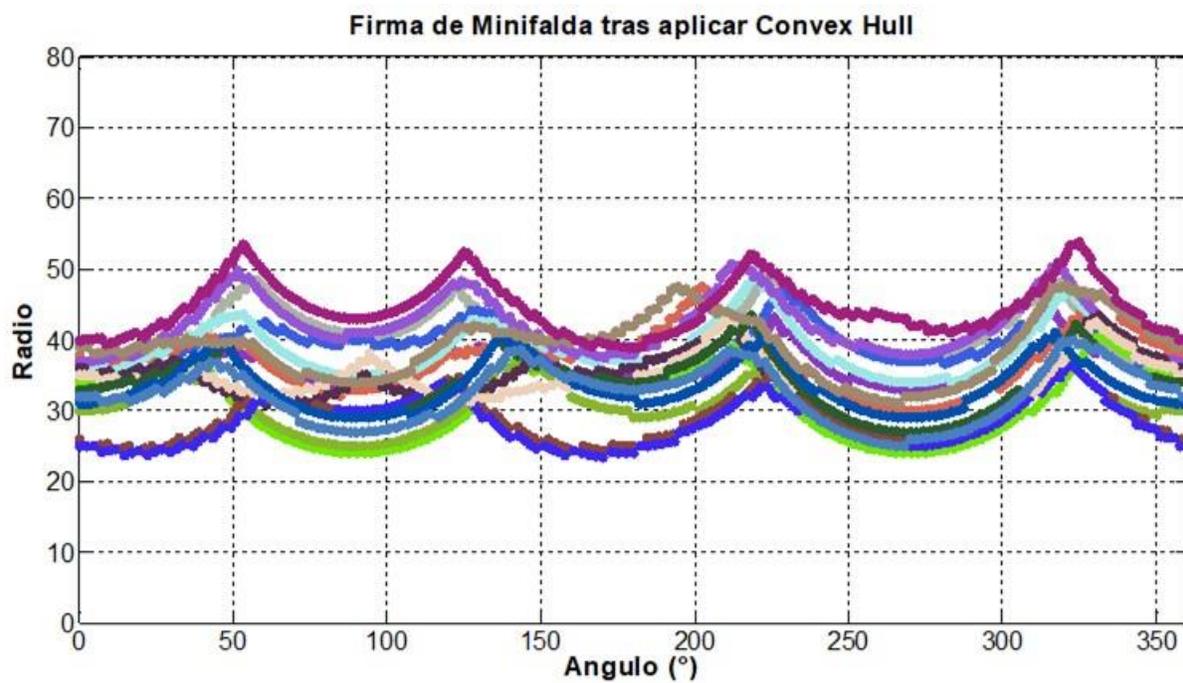
Para solucionar este problema, se simplifican los tipos de prendas para las mujeres. Por ejemplo, para las faldas existen dos estilos como la minifalda y la falda larga. Esta acción permite facilitar el reconocimiento de prendas, ya que una contiene menos área que la otra. Luego se aplica la representación de “*Convex Hull*”, utilizando la librería del OpenCV, con el fin de elaborar una aproximación del contorno de la imagen segmentada. Es decir que la imagen binaria con su forma de prenda se convierte en una figura más simple para calcular nuevamente la firma. Esto permite analizar el comportamiento de blusas con manga y sin manga, minifaldas, faldas, pantalones y vestidos (véase la Figura 69).



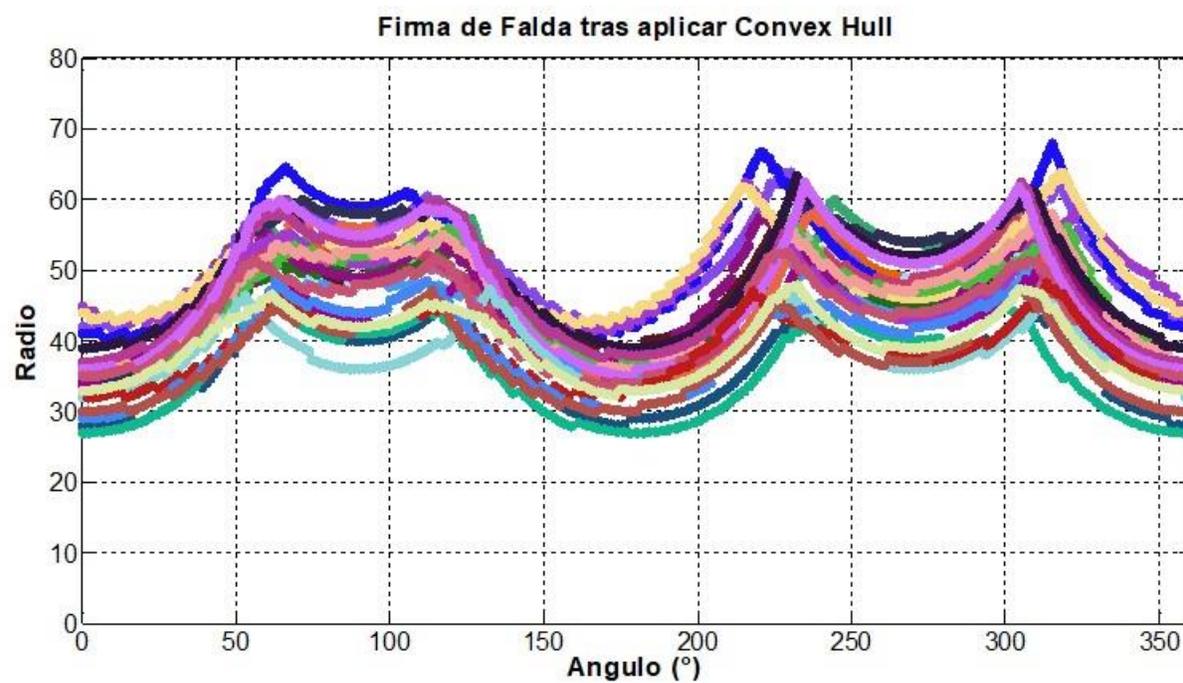
(a)



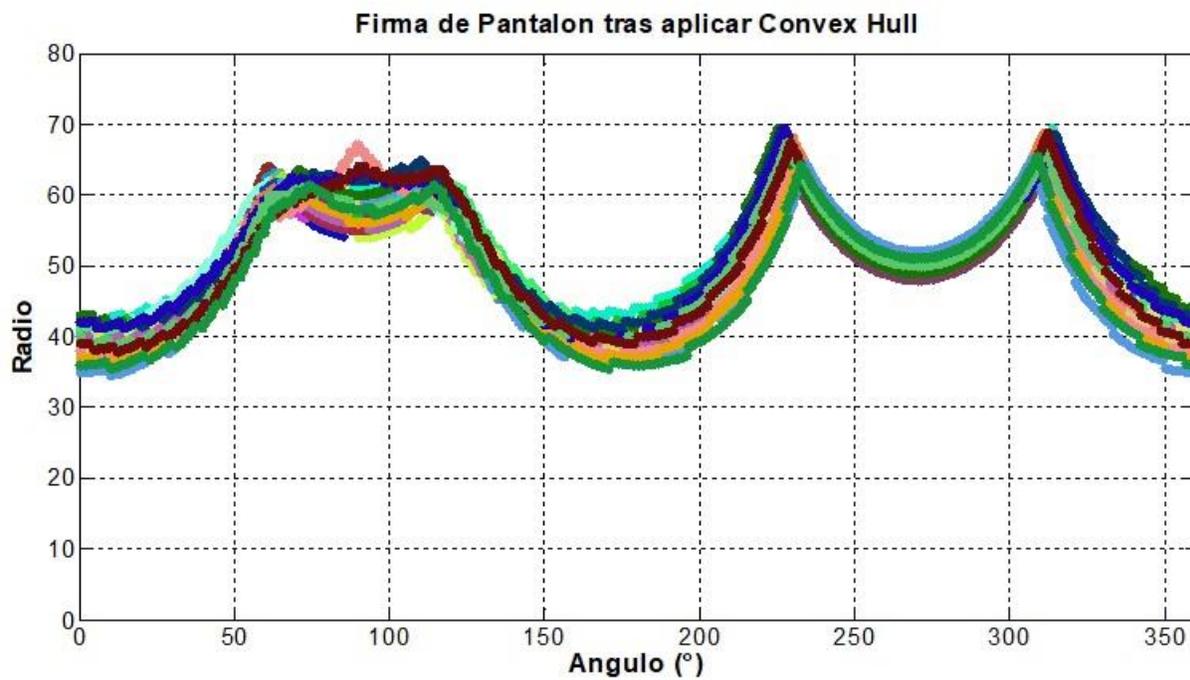
(b)



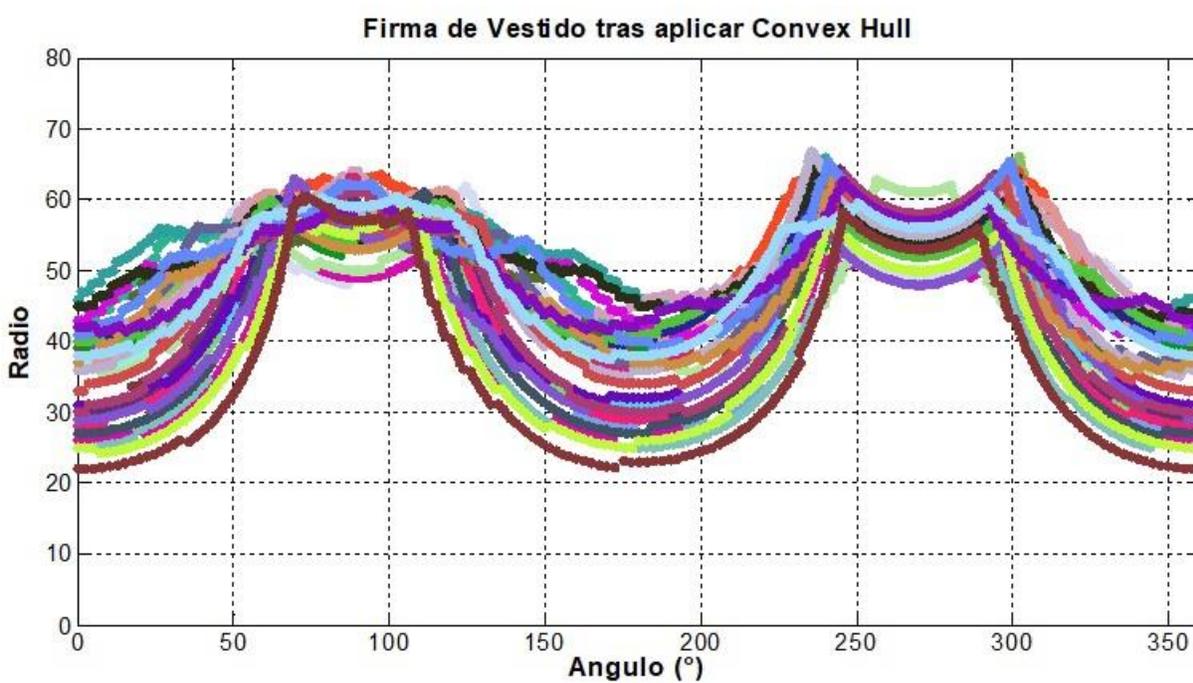
(c)



(d)



(e)



(f)

Figura 69. Resultados de grafica de Firmas de Convex Hull en Matlab: (a) Blusa sin manga, (b) Blusa con manga, (c) Minifalda, (d) Falda, (e) Pantalón, (f) Vestido.

Fuente Autores

Para el caso de mujer, se utilizan dos variables m_{03} y el valor mínimo del radio que se encuentre entre los ángulos 245° y 295° , ya que estos datos son suficientes para separar las clases deseadas (véase la Figura 70).

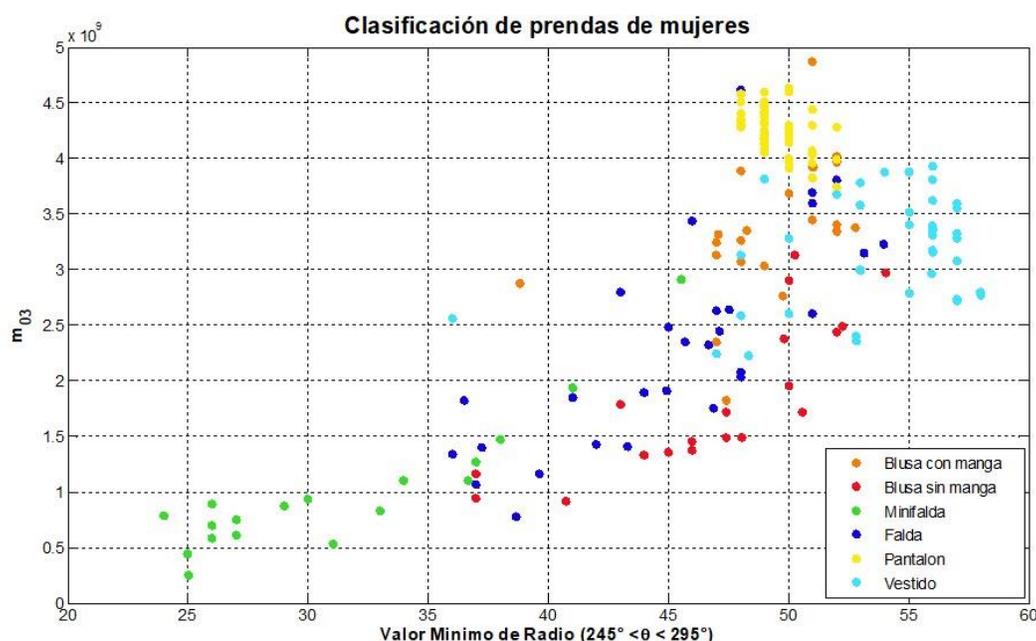


Figura 70. Resultados de distribución de los valores de las prendas para mujer en dos dimensiones.
Fuente Autores

Clasificación de representación y descriptor regional

3.3.1. Método de clasificación

Como se puede observar en la Figura 67 y Figura 70, para el proceso de clasificación se presenta un grado de complejidad debido a que los valores de las muestras se encuentran muy cerca entre ellas, por ello se decide emplear dos diferentes métodos de clasificación: el primero se basa en varios clasificadores lineales, los cuales ayudarán a definir las condiciones que separarán las clases; el segundo consiste en un clasificador no-paramétrico que, de acuerdo a las condiciones anteriormente definidas, identificará la clase a la cual pertenece la prenda que se esté analizando.

Clasificación lineal:

Dentro del campo del aprendizaje automático, el clasificador lineal se conoce como un método de aprendizaje supervisado que usa las características de un objeto para determinar a qué clase pertenece. Este se basa en los valores de la combinación lineal de dichas características, esta combinación es comúnmente representada por un vector llamado *vector de características*. El propósito de este clasificador es usar una función lineal para separar dos tipos de clases, tal como se ve en la siguiente Figura 71.

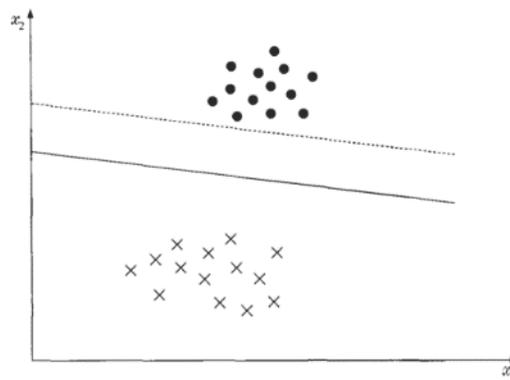


Figura 71. Ejemplo de clasificación lineal de dos clases.
Fuente (Theodoridis & Koutroumbas, 2003)

A continuación, se describe la ecuación 33 que representa la función de decisión:

$$g = f(\overline{w}^T * \overline{x} + b) = f\left(\sum_j \overline{w}_j^T * \overline{x}_j\right) \quad (33)$$

Donde \overline{x} es el vector de características, \overline{w}^T es el vector de pesos, es decir, los valores obtenidos del conjunto de muestras de entrenamiento, b el valor real de donde se intercepta la línea y f es una función que transforma el producto punto a punto de los dos vectores en la salida deseada. La función f puede representar, la mayoría de las veces, como una función que mapea todos los valores por encima de un cierto umbral, los cuales son definimos como la primera clase, y el restante como la segunda clase.

Para visualizar un problema de dos clases, se utiliza el clasificador lineal el cual permitirá realizar una partición al espacio con el fin de crear un hiperplano, aquí los puntos a un lado del hiperplano son clasificados como "sí", mientras que los demás son clasificados como "no".

Los clasificadores lineales se suelen usar en situaciones donde la velocidad de la clasificación es importante, ya que suelen realizar sus cálculos más rápido, especialmente cuando \bar{x} es disperso y cuando el número de dimensiones de este vector es grande (Yuan, Ho, & Lin, 2012).

Clasificador no paramétrico -árbol de decisiones-:

Este clasificador consiste en un sistema de decisiones multietapas, donde las clases secuencialmente rechazadas hasta llegar una etapa final donde esta sea aceptada, para ello se divide el espacio en regiones únicas, correspondiendo a cada clase de manera secuencial.

Cuando se ingresa un vector al sistema, este inicia la búsqueda de la clase a la que será establecida por medio de una secuencia de decisiones con múltiples nodos, este esquema ofrece ventajas cuando el número de clases es elevado. Esta secuencia de decisiones es aplicada a cada valor o componente, respondiendo a la condición de forma $si X_i \leq \alpha$ siendo α el valor del umbral. Esta clase de árboles son conocidas como árbol de clasificación binario ordinario, la idea básica del OBCT (por sus siglas en inglés) es que, mediante la división sucesiva secuencial de espacio se creen regiones que correspondan al número de clases indicadas, como se muestra en la Figura 72.

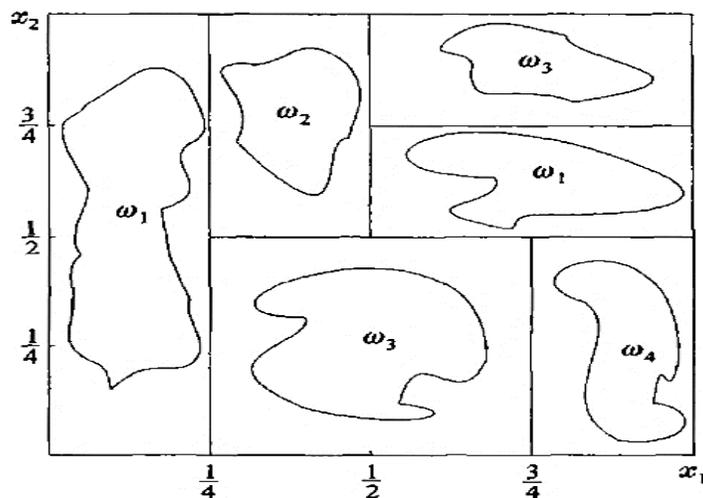


Figura 72. Partición del árbol de decisiones.
Fuente (Theodoridis & Koutroumbas, 2003)

En la Figura 73 se muestra su respectivo árbol binario, donde se encuentran las diferentes secuencias que puede tomar el clasificador, nótese que es posible alcanzar una secuencia sin necesidad de corroborar todas las otras decisiones.

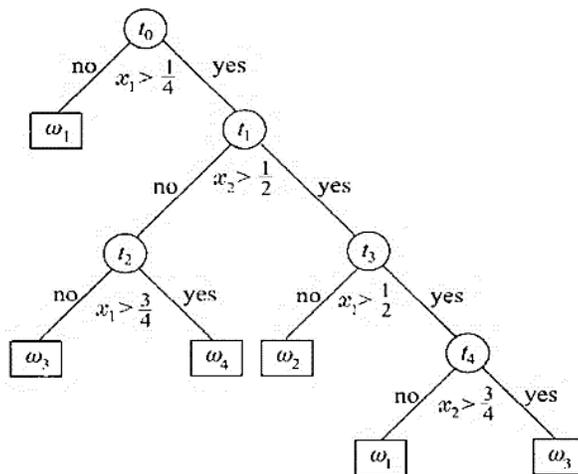


Figura 73. Clasificación árbol de decisiones.
Fuente (Theodoridis & Koutroumbas, 2003)

Los umbrales usados por la división binaria para cada nodo del árbol fueron hallados mediante el análisis de la geometría del problema. Sin embargo, esto no es posible realizarlo en espacios mayores a dos dimensiones. Para desarrollar adecuadamente el árbol binario de decisiones se debe tomar en cuenta ciertos elementos de diseño:

Por cada nodo se debe definir el conjunto de preguntas o condiciones, la cual se dividirá en dos nodos descendientes.

- Para cada nodo deben definirse las decisiones para las dos ramas, sin importar si esta nunca se cumple. Cada decisión o pregunta corresponde a una división binaria específica. El primer nodo raíz se encuentra asociado con el conjunto de entrenamiento “ X ”. Cada nodo “ t ” se encuentra asociado con un subconjunto específico “ X_t ”, presente en el conjunto de entrenamiento “ X ”, la división del nodo equivale a la división del subconjunto “ X_t ” en dos subconjuntos debajo de este: el primero, “ X_{tY} ” consiste en los vectores “ X_t ” que pertenecen a la respuesta “Si” de la pregunta, y el segundo “ X_{tN} ” corresponde a la respuesta “No”.

Por lo tanto, lo descrito en la ecuación 34 es verdadero:

$$\begin{aligned} X_{tY} \cap X_{tN} &= \emptyset \\ X_{tY} \cup X_{tN} &= X_t \end{aligned} \tag{34}$$

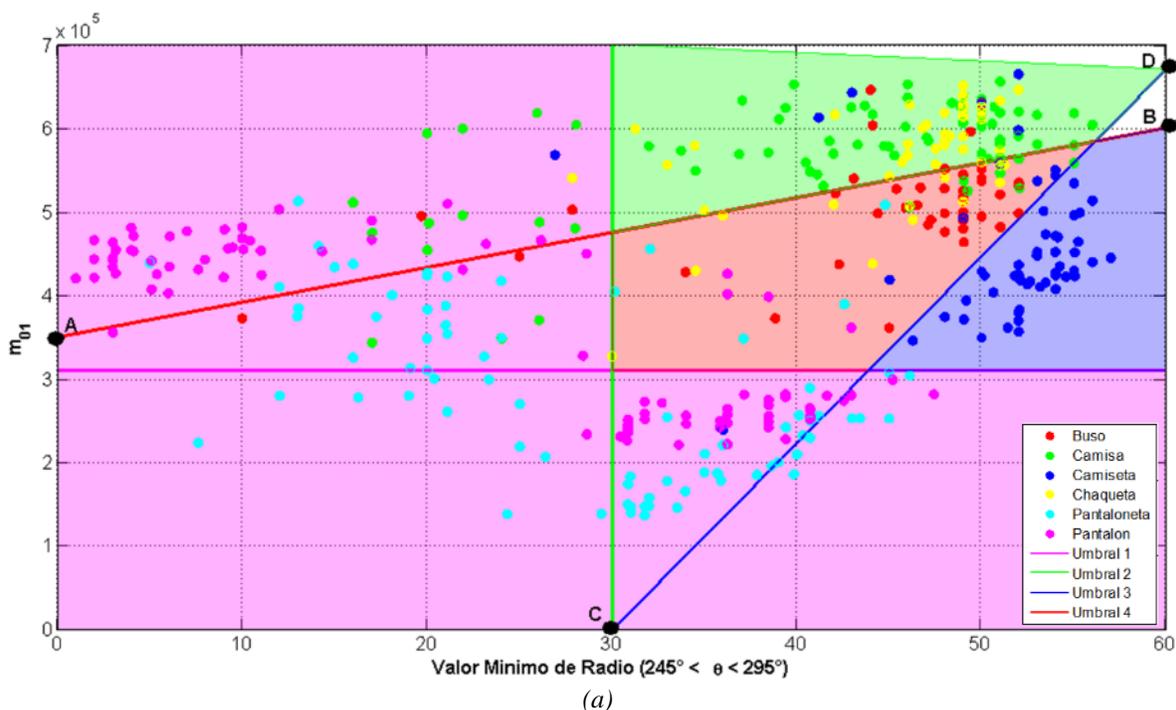
- Se debe adoptar un criterio de división donde el cual se elija la mejor división del conjunto de candidatos.
- Se requiere una regla que detiene la división para controlar el crecimiento del árbol y que además declare el nodo como el final de la rama (hoja).
- Se requiere una regla que asigna cada hoja a una clase específica (Theodoridis & Koutroumbas, 2003).

3.3.2. Sistema de clasificación

Para el diseño del proceso de clasificación se opta por establecer las condiciones de los umbrales para el clasificador lineal de forma empírica, debido a que se poseen más de dos clases para realizar la clasificación, además mediante este proceso, es posible conseguir resultados más rápido en el desarrollo del proceso de clasificación.

Para la categoría del hombre:

Con el fin de identificar los umbrales para el proceso de clasificación, se debe tener en cuenta el comportamiento entre dos descriptores. Para ello se realiza un ejercicio sencillo y práctico que es, de la gráfica en tres dimensiones, expresada en la Figura 67, descomponerla en tres diferentes gráficas, correspondientes a la proyección sobre los planos donde se comparan dos tipos de descriptores. Esto facilita definir intuitivamente las condiciones que separan a los diferentes tipos de prenda. En la Figura 74 se muestra esta descomposición.



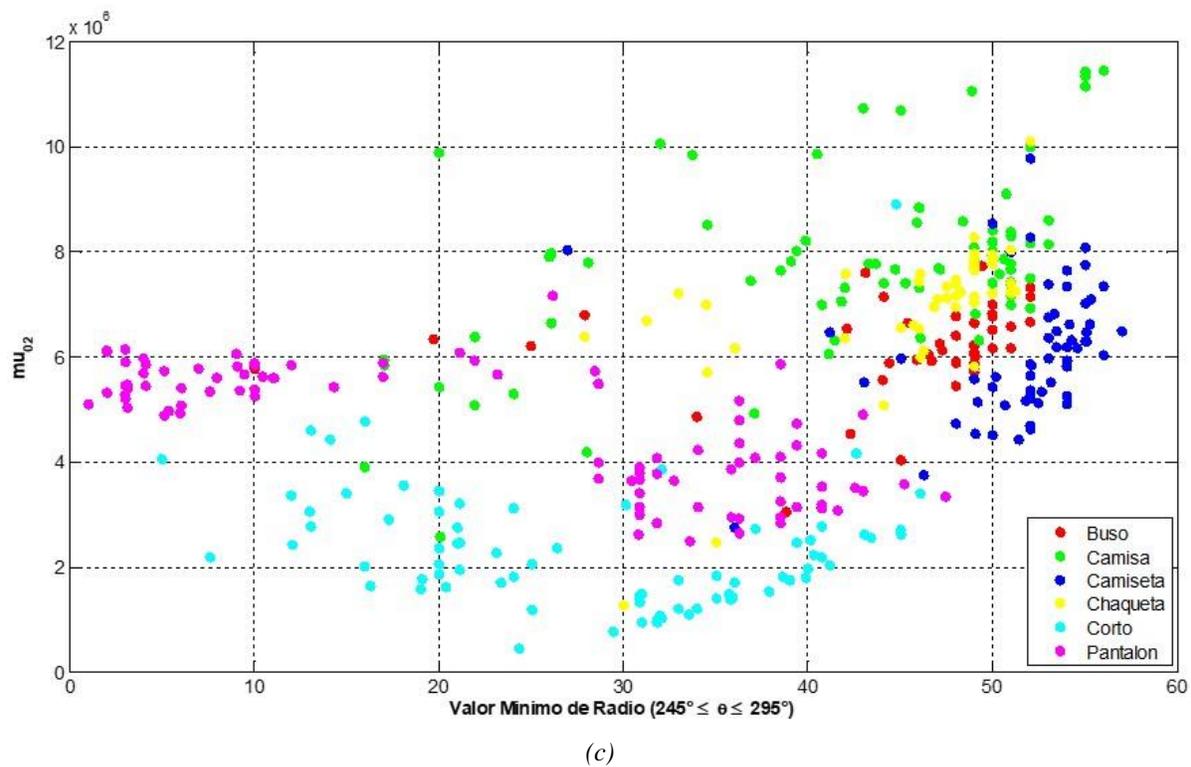
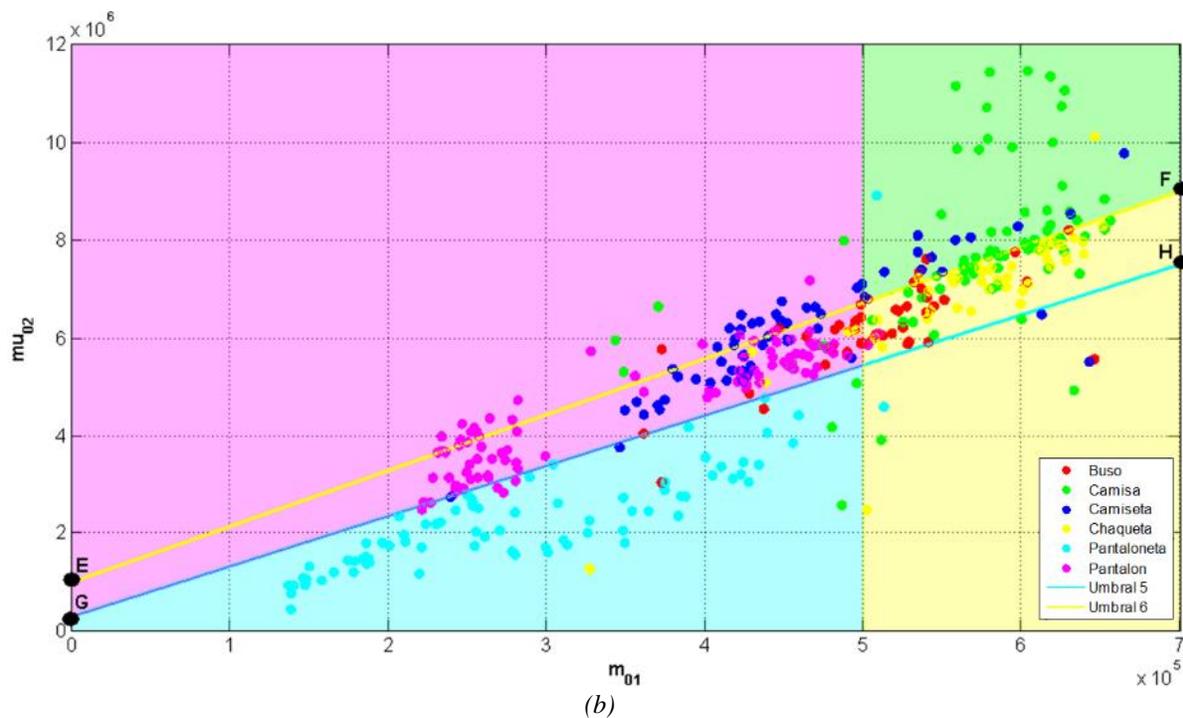


Figura 74. Resultados de distribución de los valores de las prendas en dos dimensiones (a) Plano cartesiano de valor mínimo de radio y μ_{02} , (b) Plano cartesiano de m_{01} y μ_{02} , (c) Plano cartesiano de valor mínimo de radio y μ_{02} .

Fuente Autores

En la Figura 74, se observa los resultados de la proyección sobre plano de la Figura 67. Se puede ver que las secciones (a) y (b) presentan cuatro y dos umbrales respectivamente, los cuales fueron seleccionados para separar los valores posibles que permitirán identificar los tipos de prenda: buso, camisa, camiseta, chaqueta, pantaloneta y pantalón.

Para identificar estos umbrales, se utiliza la ecuación de la línea recta para calcular los valores de las pendientes. En los umbrales 1 y 2, no es necesario, pues basta con una variable. Para el primer umbral se hace el corte en el eje Y, mientras que el umbral 2 lo hace con el eje X.

Para los umbrales restantes (3, 4, 5 y 6), se utiliza la ecuación 35 para hallar sobre respectiva pendiente.

$$m = \frac{Y_2 - Y_1}{X_2 - X_1} \quad (35)$$

A continuación, se conoce las siguientes expresiones de las pendientes de los umbrales utilizando la ecuación 35 con la información de la Figura 74 :

$$\text{Pendiente de Umbral 3: } m_3 = \frac{B_y - A_y}{B_x - A_x} \quad (36)$$

$$\text{Pendiente de Umbral 4: } m_4 = \frac{D_y - C_y}{D_x - C_x} \quad (37)$$

$$\text{Pendiente de Umbral 5: } m_5 = \frac{F_y - E_y}{F_x - E_x} \quad (38)$$

$$\text{Pendiente de Umbral 6: } m_6 = \frac{H_y - G_y}{H_x - G_x} \quad (39)$$

Con estos primeros umbrales -1 y 2- y las demás pendientes, permiten formar zonas para así conocer la ubicación de los valores según la prenda a la que corresponde.

De acuerdo con lo descrito en este capítulo y la experimentación, para la clasificación se toman los siguientes rangos de valores para cada uno de los descriptores usados:

Para la primera clasificación, la cual consiste en diferenciar las prendas tipo “Torso” y “No torso”, se usa la información obtenida por el método de representación “Firma”. Se toma en cuenta que el valor mínimo del radio se encuentre entre los ángulos 245° y 295° , como se describe a continuación:

$$r_{min} = \text{Valor M\u00ednimo de Radio } (245^\circ < \theta < 295^\circ)$$

$$\text{Primera clasificaci\u00f3n} = \begin{cases} \text{Si } (r_{min} > 30) \text{ \u00f3 } (m_{01} > 3.1 * 10^5) = \text{Torso} \\ \text{Si } (r_{min} \leq 30) \text{ \u00f3 } (m_{01} \leq 3.1 * 10^5) = \text{No torso} \end{cases} \quad (40)$$

De acuerdo con el resultado obtenido en la primera clasificaci\u00f3n, se usan nuevamente los descriptores mencionados, pero con unas condiciones distintas, ya que esta decisi\u00f3n consiste en identificar qu\u00e9 tipo de prendas es. Para llegar a esto, se conoce los valores de los descriptores de la prenda capturada las cuales representan como punto de coordenada donde luego se halla la pendiente de este. Donde los puntos iniciales para hallar dicha pendiente son los mismos puntos usados por los umbrales — 3, 4, 5 y 6 — y el punto de coordenada final ser\u00e1n los valores de los descriptores de la prenda capturada como muestras las siguientes ecuaciones:

$Q = \text{Punto de coordenada de la prenda capturada}$

$$\text{Pendiente 1: } p_1 = \frac{Q_y - A_y}{Q_x - A_x} \quad (41)$$

$$\text{Pendiente 2: } p_2 = \frac{Q_y - C_y}{Q_x - C_x} \quad (42)$$

$$\text{Pendiente 3: } p_3 = \frac{Q_y - E_y}{Q_x - E_x} \quad (43)$$

$$\text{Pendiente 4: } p_4 = \frac{Q_y - G_y}{Q_x - G_x} \quad (44)$$

Luego se procede la segunda clasificación de las prendas, esto mediante los valores de las pendientes calculadas y con las ecuaciones 39, 40, 41 y 42, con el fin de distinguir seis tipos de prendas. Para el caso de “Torso” se expresa en la siguiente manera:

$$\text{Torso} = \begin{cases} \text{Si } p_2 \leq m_4 = \text{Camiseta} \\ \text{Si } p_2 > m_4 \text{ y } p_1 \leq m_1 = \text{Buso} \\ \text{Si } p_1 > m_1 \text{ y } p_3 \leq m_5 = \text{Chaqueta} \\ \text{Si } p_1 > m_1 \text{ y } p_3 > m_5 = \text{Camisa} \end{cases} \quad (45)$$

Para las prendas de “No torso”:

$$\text{No Torso} = \begin{cases} \text{Si } p_4 \leq m_6 = \text{Pantaloneta} \\ \text{Si } p_4 > m_6 = \text{Pantalón} \end{cases} \quad (46)$$

En la Figura 75 se muestra el árbol de decisiones establecido con las condiciones anteriormente descritas:

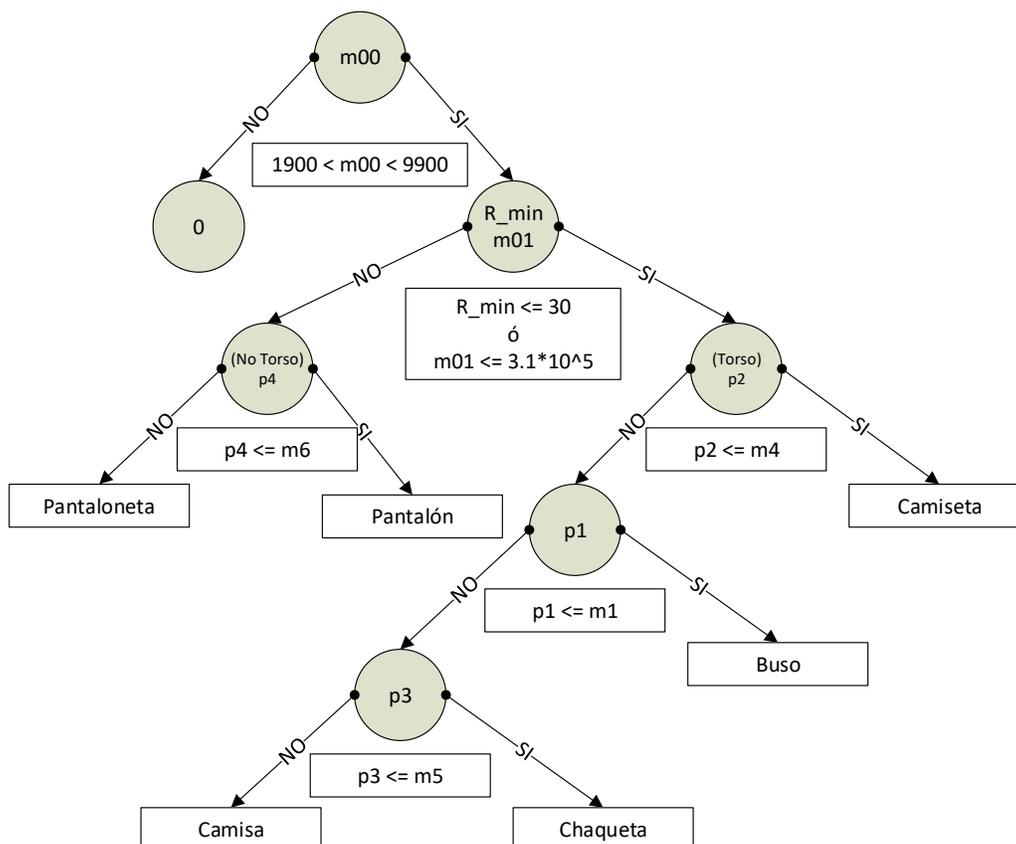


Figura 75. Árbol de decisiones para el sistema de clasificación para prendas de hombre.
Fuente Autores.

Para la categoría de la mujer:

Para realizar la clasificación de estas prendas, se usa la misma metodología utilizada en el proceso de clasificación de prendas para hombre. Este caso, para las prendas de mujeres, se tienen en cuenta la información extraída de las variables m_{03} y el valor mínimo del radio entre los ángulos 245° y 295° . Estos datos son suficientes para la clasificación, como se ve en la Figura 76. En esta figura se presentan cinco umbrales, los cuales fueron seleccionados para crear fronteras de los seis tipos de prendas: blusa con manga, blusa sin manga, minifalda, falda, pantalón y vestido.

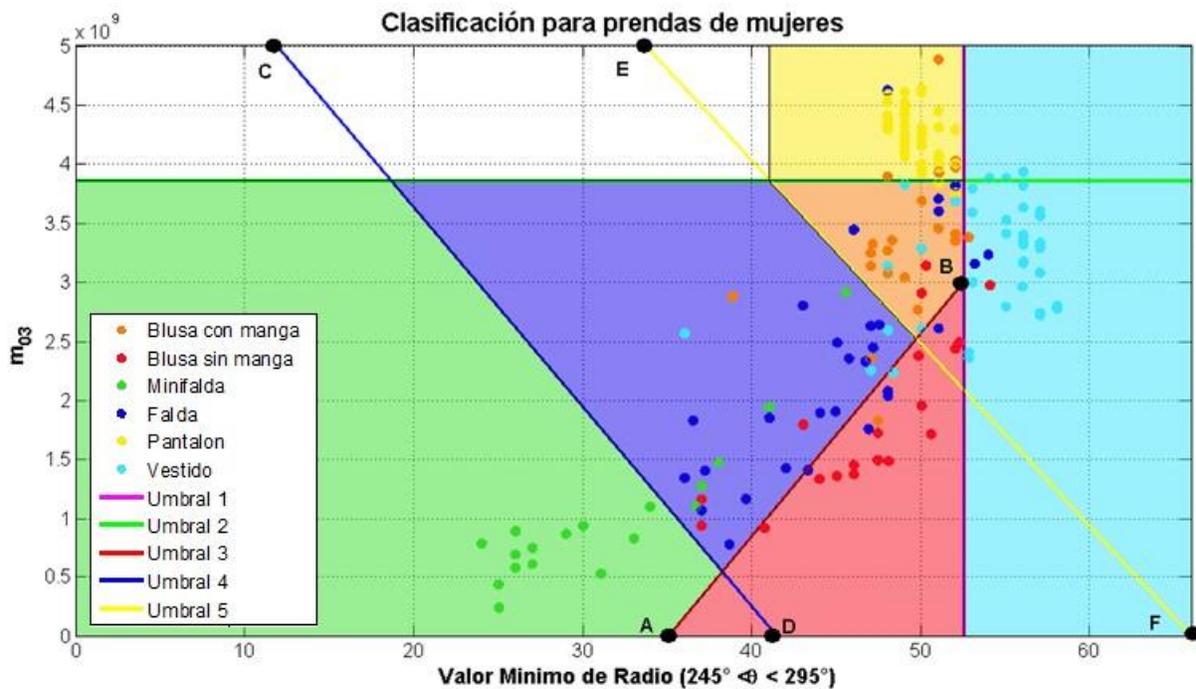


Figura 76. Resultados de clasificación para prendas de mujeres con sus hiperplanos.
Fuente Autores

Para identificar los umbrales, se utiliza la ecuación 33 con el fin de hallar la pendiente de ella con los valores de los puntos A, B, C, D, E y F de la Figura 76.

Q = Punto de coordenada de la prenda capturada

$$\text{Pendiente 3: } p_3 = \frac{Q_y - A_y}{Q_x - A_x} \quad (47)$$

$$\text{Pendiente 4: } p_4 = \frac{Q_y - C_y}{Q_x - C_x} \tag{48}$$

$$\text{Pendiente 5: } p_5 = \frac{Q_y - E_y}{Q_x - E_x} \tag{49}$$

Después de obtener la pendiente de la prenda capturada, se procede el método de clasificación, árbol de decisiones, con el fin de crear una jerarquía de prendas con la información dada de la gráfica y las pendientes de los umbrales de las ecuaciones 45, 46 y 47.

$$\text{Clasificación} = \begin{cases} r_{min} = \text{Valor Mínimo de Radio } (245^\circ < \theta < 295^\circ) \\ \text{vestido, } r_{min} > 52.5 \\ \text{blusa sin manga, } r_{min} < 52.5 \text{ y } P_3 < P_{u3} \\ \text{pantalón, } r_{min} < 32.5 \text{ y } m_{03} > U_2 \\ \text{Blusa con manga, } r_{min} < 52.5 \text{ y } m_{03} < U_3 \text{ y } P_5 > P_{u5} \\ \text{faldas, } r_{min} < 52.5 \text{ y } P_5 > P_{u5} \text{ y } P_3 > P_{u3} \text{ y } P_4 > P_{u4} \\ \text{minifalda, } P_4 < P_{u4} \end{cases} \tag{50}$$

En la Figura 77 se muestra el árbol de decisiones desarrollado de acuerdo con las condiciones previamente descritas:

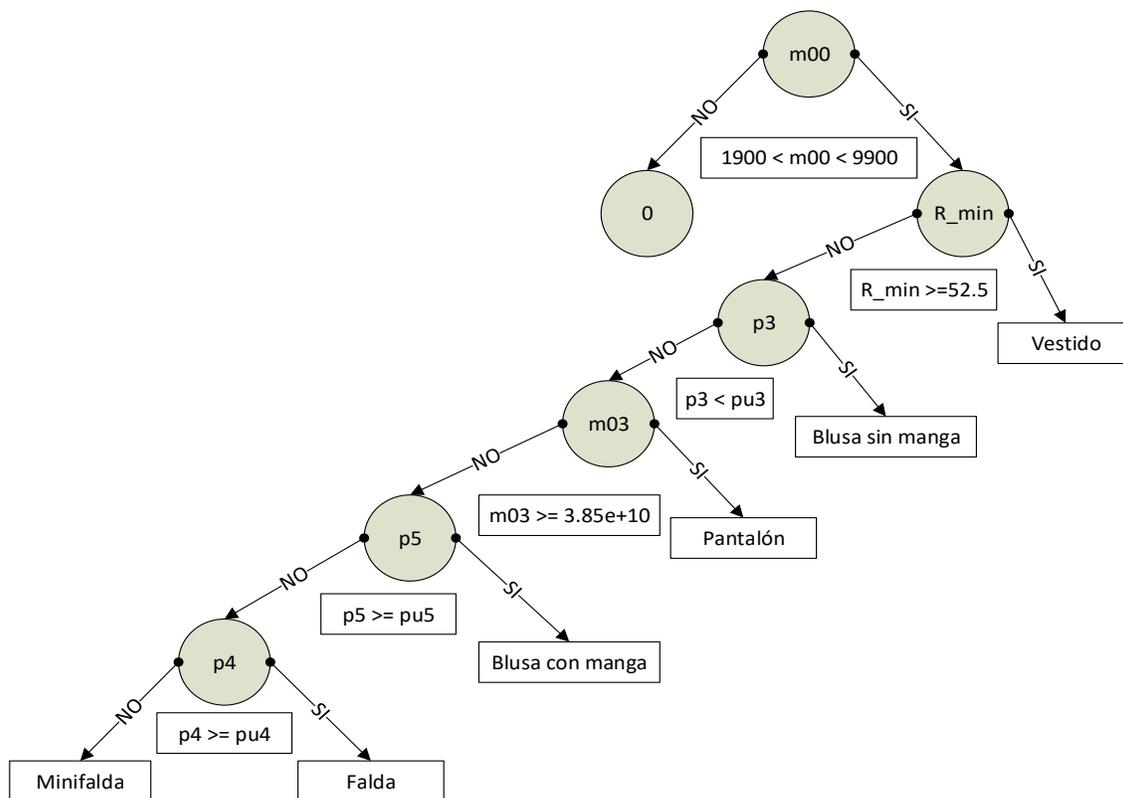


Figura 77. Árbol de decisiones para sistema de clasificación de prendas para mujer.

Fuente Autores.

Para conocer en más detalle el funcionamiento del sistema desarrollado, se adjuntan los códigos de programación para ser ejecutados, tanto en el sistema operativo Windows como para Ubuntu Mate para ODROID XU4, estos se encuentran en la carpeta “**ANEXO B**” dentro del medio digital adjunto a este documento. Para poder hacer uso del sistema en plataforma Windows es necesario instalar el software de Visual Studio y OpenCV, el manual de estas instalaciones y configuraciones se encuentran en la carpeta “**ANEXO C**” del medio digital.

Capítulo 4

ANALISIS Y RESULTADOS

4.1. Revisión del espacio de trabajo y configuración del sistema

Para comenzar con el proceso de pruebas, primero se examinan ciertas condiciones ambientales del lugar donde se encuentran las prendas de vestir. Esto con el fin de determinar si es adecuado para realizar el reconocimiento, ya que algunos de estos pueden perturbar, ya sea de manera negativa o positiva, al funcionamiento del sistema. Dentro de estos factores se revisaron aspectos como iluminación del espacio, forma de organización de las prendas, espacio suficiente para moverse dentro del área donde están las prendas y color de los estantes donde se cuelgan las prendas:

- Iluminación: Se verifica la luminiscencia del lugar, con el fin de comprobar si es la suficiente para que el sistema no genere al error en el procesamiento.
- Organización de las prendas: Se rectifica si las prendas se encuentran organizadas de manera que puedan ser analizadas por el sistema, sin presentar ningún problema, véase un ejemplo en la Figura 78.
- Espacio suficiente para moverse: Se corrobora que el espacio entre los estantes, donde transita la gente, sea mayor o igual a un metro de distancia.



*Figura 78. Ejemplo de organización de las prendas para que el sistema lo analice correctamente.
Fuente Autores.*

Después de verificar estos factores, se procede con la inicialización del sistema. En esta se hace una revisión a los parámetros de funcionamiento para confirmar que este realice su tarea de manera adecuada, tales como el estado de conexión de la cámara, ajuste del módulo de audio, volumen del sonido, recepción y captura de imagen, y validación de la medición hecha por el sensor de distancia.

Una vez iniciado el sistema, se valida la información transmitida por el sensor de distancia, de forma visual y auditiva:

- Visual: Para comprobar el sistema de forma visual se usa un software VNC (*Virtual Network Computing*) que utilice el protocolo RFB (*Remote Frame Buffer*) para controlar remotamente el sistema desde un computador externo y observar el comportamiento del sistema operativo.
- Auditivo: Como el sistema fue diseñado de tal manera que, cuando este detecte que el individuo se encuentra en la distancia correcta para realizar el procesamiento, reproduzca un archivo de audio advirtiendo al usuario de esto.

Después de esto, el sistema comienza la transmisión de vídeo proveniente del módulo de cámara, cuando este detecte que la distancia entre el dispositivo y el estante donde se encuentra la prenda de vestir sea de un (1) metro, tomará el fotograma del vídeo en ese instante de tiempo. Luego esta imagen capturada pasa como valor de entrada en las funciones de procesamiento de imágenes y clasificación. Una vez el fotograma de la prenda procesada es clasificado, el sistema reproduce un archivo de audio confirmando el tipo de prenda identificada.

Para las pruebas de reconocimiento de prendas en el ambiente controlado, solo se tomaron 764 de muestras debido a que solo se contaba con este número de prendas para realizar esta verificación.

Para el caso de las tiendas de ropa, solamente se obtuvieron 3260 de muestras, ya que este era el número de prendas disponibles en estos lugares.

Las muestras utilizadas en este proyecto se encuentran disponibles en la carpeta “ANEXO D” en el medio digital, allí se podrán examinar tanto estas muestras como los resultados obtenidos del sistema de segmentación desarrollado para este proyecto. Además, se adjunta un vídeo donde se muestra el funcionamiento del sistema en las tiendas de ropa, el cual se encuentra dentro de la carpeta “ANEXO E”.

4.2. Pruebas del sistema en ambiente controlado

Antes de realizar pruebas del sistema en el ambiente para el cual fue diseñado, se realizaron unas pruebas de funcionamiento en un área controlada, es decir que factores como la iluminación, posición de la prenda y el espacio para la interacción del sistema, eran ideales para que el sistema funcionara de tal manera que se lograra un reconocimiento y una identificación mayor a un ochenta por ciento (80 %). Para lograr esto, se adecuó un espacio que cumplió con los siguientes requerimientos:

- Área de trabajo: Se usó un espacio mayor a tres metros cuadrados ($3 m^2$), con paredes de tono claro.
- Iluminación: En esta área se instala un sistema de iluminación que apunte directamente a la prenda, de tal manera que se minimice la formación de sombras en esta y la iluminación natural como se ve en la Figura 79 .
- Organización de las prendas: Las prendas fueron colgadas en forma frontal, ya que es necesario para el procesamiento.



Figura 79. Representación gráfica de los factores ambientales controlados para obtener una prueba exitosa.
Fuentes Autores.

4.2.1. Resultados de clasificación para prendas de hombre

En la Tabla 8 se exponen los resultados de la clasificación de 457 muestras recolectadas en esta prueba (véase en la Figura 80). Se puede observar que solo 403 prendas lograron ser procesadas y clasificadas por el sistema. Las muestras restantes —54 imágenes— no pasaron por la clasificación debido a que no cumplieron con las condiciones establecidas en el proceso de segmentación, donde se toma en cuenta el área de la figura en el primer plano de la imagen, debe estar entre un valor de 1900 a 9900 píxeles, para poder pasar al proceso de clasificación, tal como se indica en el capítulo 3. Esto sucede debido a que la prenda a procesar posee algún tipo de etiqueta o diseño, que se destaca más en el primer plano de la imagen que la misma prenda, perturbando la segmentación y posteriormente la clasificación. Otro caso es cuando el objeto en el primer plano -la prenda de vestir- posee un tono de color muy similar al fondo de la imagen, logrando que el sistema lo reconozca como un solo plano.



Figura 80. Ejemplo de muestras recolectadas y usadas en el sistema.
Fuente Autores.

Tabla 8. Estadísticas de las prendas para hombre que se obtuvieron en la prueba de ambiente controlado.

PRENDA	TOTAL	NO CLASIFICA	SUBTOTAL	TORSO	NO TORSO
BUSO	52	7	45	41	4
CAMISA	84	7	77	63	14
CAMISETA	74	14	60	58	2
CHAQUETA	52	2	50	48	2
CORTO	100	23	77	4	73
PANTALON	95	1	94	4	90
TOTAL	457	54	403	218	185

Fuente Autores.

Para expresar de manera concreta la información adquirida por el sistema, se emplea la Matriz de confusión. Este analiza un problema con n clases por medio de una matriz $n \times n$, donde en las filas se designa las clases que son reales y las columnas las previstas por el modelo. Aquí se muestra de manera explícita cuando una clase es confundida por la otra, permitiendo que sea posible trabajar los diferentes errores de forma separada (Visa, Ramsay, Ralescu, & Van Der Knaap, 2011).

Con el fin de comprender mejor el comportamiento del sistema de clasificación, se elabora una matriz de confusión para la primera y segunda clasificación. En la Tabla 9 se expresan los resultados obtenidos en la primera clasificación, la cual radica en diferenciar las prendas entre “Torso” (es decir, vestimenta que se pone/lleva puesto en la parte superior del cuerpo), y “No torso” (prendas que se ponen/llevan puestas en la parte inferior del cuerpo). Para esta primera se obtuvo una clasificación satisfactoria para estos dos grupos, donde para prendas tipo “Torso” se obtuvo una sensibilidad de 90.52 % y para tipo “No torso” de 95.32%.

Tabla 9. Matriz de confusión de la primera clasificación para el ambiente controlado.

RECALL		VALOR PREDICHO (%)	
		TORSO	NO TORSO
VALOR REAL	TORSO	90,52 %	9,48 %
	NO TORSO	4,68 %	95,32 %

Fuente Autores.

En el segundo tipo de clasificación para la identificación de las prendas segmentadas, se obtuvo que prendas tipo “No torso” se clasificaron correctamente, obteniendo una sensibilidad mayor de 90%, por otro lado, para la clasificación tipo “Torso” se generó una sensibilidad menor al 60 % (véase la Tabla 10), esto se debe a que las prendas que poseen mangas largas como el buso y la chaqueta poseen una forma similar en el contorno de una camisa, generando que los descriptores posean un valor igual al de la camisa, y se obtenga un error en la clasificación.

Tabla 10. Matriz de confusión de la segunda clasificación para el ambiente controlado.

RECALL		VALOR PREDICHO (%)					
		BUSO	CAMISA	CAMISETA	CHAQUETA	CORTO	PANTALON
VALOR REAL	BUSO	77,78 %	4,44 %	0,00 %	8,89 %	0,00 %	8,89 %
	CAMISA	10,39 %	36,36 %	0,00 %	35,06 %	6,49 %	11,69 %
	CAMISETA	10,00 %	5,00 %	78,33 %	3,33 %	1,67 %	1,67 %
	CHAQUETA	22,00 %	2,00 %	0,00 %	72,00 %	2,00 %	2,00 %
	CORTO	5,19 %	0,00 %	0,00 %	0,00 %	94,81 %	0,00 %
	PANTALON	4,26 %	0,00 %	0,00 %	0,00 %	5,32 %	90,43 %

Fuente Autores.

4.2.2. Resultados de clasificación para prendas de mujeres

Para esta prueba fueron analizadas un total de 207 muestras (véase la Figura 81), donde se ven incluidas las prendas para dama como vestidos, faldas, blusas y pantalones. Los resultados de esta prueba se registran en la Tabla 11, donde se observa que solo 177 prendas fueron procesadas por el sistema, del valor restante de muestras pertenecen al grupo de prendas que obtuvieron una segmentación fallida, provocando que no pueda ser reconocido por el proceso de clasificación. El motivo de este comportamiento es debido a que, como sucedió anteriormente con la clasificación de prendas para hombre, existían factores que perturbaban el proceso de segmentación, para este caso esta perturbación fue provocada por los colores presentes en el primer plano de las prendas, donde en algún área dentro de la prenda se exhibe un tono muy similar al fondo de la imagen.

En la Tabla 12 se muestra los resultados en este ambiente, con esto se puede concluir que los pantalones fue la prenda que se obtuvo una clasificación de 95,92% con respecto los demás. Tomando en cuenta que las formas de las prendas son muy similares, se logran clasificar cada una de ellas con una sensibilidad media alta.



Figura 81. Ejemplo de muestras recolectadas de prendas para mujer, que fueron usadas en el sistema.
Fuente Autores.

Tabla 11. Estadísticas de las prendas para mujer que se obtuvieron en la prueba de ambiente controlado.

PRENDA	SUBTOTAL	NO CLASIFICA	TOTAL
BUSO CON MANGA	34	13	21
BUSO SIN MANGA	26	7	19
MINIFALDA	19	1	18
FALDA	34	3	31
PANTALON	51	2	49
VESTIDO	43	4	39
TOTAL	207	30	177

Fuente Autores.

Tabla 12. Matriz de confusión de las prendas de mujeres en el ambiente controlado

RECALL		VALOR PREDICHO (%)					
		BUSO CON MANGA	BUSO SIN MANGA	MINIFALDA	FALDA	PANTALON	VESTIDO
VALOR REAL	BUSO CON MANGA	57,14%	4,76%	0,00%	9,52%	23,81%	4,76%
	BUSO SIN MANGA	10,53%	68,42%	0,00%	15,79%	0,00%	5,26%
	MINIFALDA	0,00%	0,00%	72,22%	27,78%	0,00%	0,00%
	FALDA	12,90%	12,90%	0,00%	58,06%	9,68%	6,45%
	PANTALON	4,08%	0,00%	0,00%	0,00%	95,92%	0,00%
	VESTIDO	12,82%	0,00%	0,00%	10,26%	0,00%	76,92%

Fuente Autores.

4.3. Resultado global de las pruebas realizadas

Después de concluidas las pruebas en el ambiente controlado, se procedió a verificar el sistema en el espacio para el cual fue diseñado. Para ello se eligieron cuatro establecimientos comerciales donde vendían ropa, y allí se hizo el mismo análisis a los factores ambientales que pueden perturbar el funcionamiento del sistema, dando como resultado lo siguiente:

- **Iluminación:** En la mayoría de establecimientos contaban con iluminación artificial, se confirmó por medio de muestras fotográficas, examinando más exactamente el contraste de la imagen, para determinar la calidad de la iluminación. Se concluyó que era deficiente, ya que, en varios casos dependían más de la luz natural que de la artificial. Otro factor que desfavoreció su calidad fue la dirección de enfoque de la luz sobre las prendas.
- **Espacio de interacción:** Este era el requerido, esto quiere decir que se podía movilizar dentro del lugar sin ningún problema y sin afectar el funcionamiento del sistema.
- **Organización de las prendas:** La mayoría de las prendas se encontraban colgadas de forma frontal.

4.3.1. Resultados de clasificación para prendas de hombre

En esta prueba se muestra los resultados de la clasificación de 2663 muestras recolectadas en las tiendas de ropa (véase en la Figura 82), donde se puede ver que solo 2556 prendas procesadas por la clasificación del sistema. Las 107 imágenes restantes no lograron clasificarse debido a que existen factores que perjudican al proceso de segmentación, como la temperatura del color de la luz dentro de la tienda (luz cálida, neutra o fría), la organización de las prendas, el color de las paredes y del fondo de los muebles para tender la prendas. En la Tabla 13 se muestran los resultados en este ambiente, con esto se puede concluir que las pantalonetas y los pantalones no fueron agrupados correctamente en la clase correspondiente, debido a que la mayoría de estos

fueron reconocidas como tipo “Torso”, por esta razón, la clasificación de tipo “No torso” obtuvo una sensibilidad de 45,61%. En cambio, el tipo “Torso” se obtuvo una sensibilidad de 80% como lo indica en la Tabla 14.



Figura 82. Ejemplo de muestras recolectadas de prendas para hombre, que fueron usadas en el sistema.
Fuente Autores.

Tabla 13. Estadísticas de las prendas para hombre que se obtuvieron en la prueba de tienda de ropa.
Fuente autores.

PRENDA	TOTAL	NO CLASIFICO	SUBTOTAL	TORSO	NO TORSO
BUSO	351	28	323	224	99
CAMISA	801	32	769	681	88
CAMISETA	873	34	839	698	141
CHAQUETA	335	6	329	206	123
CORTO	115	4	111	61	50
PANTALON	188	3	185	100	85
TOTAL	2663	107	2556	1970	586

Tabla 14. Matriz de confusión de la primera clasificación para la tienda ropa.
Fuente Autores.

RECALL		VALOR PREDICHO (%)	
		TORSO	NO TORSO
VALOR REAL	TORSO	80,04	19,96
	NO TORSO	54,39	45,61

Tabla 15. Matriz de confusión de segunda clasificación para la tienda ropa.

RECALL		VALOR PREDICHO (%)					
		BUSO	CAMISA	CAMISETA	CHAQUETA	CORTO	PANTALON
VALOR REAL	BUSO	46,75%	7,43%	4,64%	10,53%	14,55%	16,10%
	CAMISA	23,32%	29,27%	11,92%	24,09%	1,30%	10,10%
	CAMISETA	32,06%	9,21%	38,76%	3,11%	3,23%	13,64%
	CHAQUETA	37,39%	7,90%	4,26%	13,07%	11,55%	25,84%
	CORTO	23,42%	0,00%	2,70%	28,83%	30,63%	14,41%
	PANTALON	30,81%	4,86%	17,30%	1,08%	5,41%	40,54%

Fuente Autores.

De acuerdo con los resultados de la Tabla 15, se observa que la segunda clasificación se obtuvo un reconocimiento menor al 50 % en todas las prendas, esto se debió a que factores ambientales como la iluminación, afectaron el proceso de segmentación y consecuentemente a la clasificación. Un ejemplo de esto se demuestra en la sensibilidad obtenida en el clasificador para pantalones, donde el 30.81% fue identificado como buso, esto debido a lo anteriormente dicho, se muestra el resultado de esta segmentación en la Figura 83.

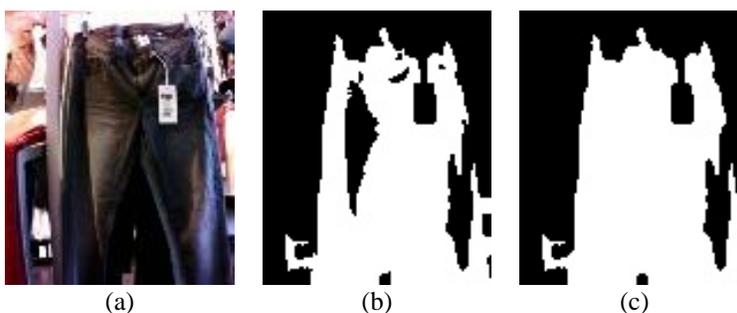


Figura 83. (a) Imagen original del pantalón, (b) Imagen segmentada de pantalón, (c) Resultado no deseado de relleno de la imagen segmentada.

Fuentes Autores.

En la Figura 83 se observa que de la imagen original no se obtuvo una segmentación exitosa de la prenda, esto se debió, tanto al color de la prenda, el cual era similar al fondo de la imagen, como a la iluminación, ya que no era constante en el primer plano de esta. Todo lo anterior que el proceso de clasificación lo identificará como una prenda tipo “Torso”, y consecuentemente como un buso.

Tabla 16. Comparación de los valores de los descriptores de Buso y Pantalón.

PRENDA REAL	CLASE 1	CLASE 2	VALOR RADIO	VALOR M01	VALOR MU02
Pantalón	T	Buso	41,05	431742	5,09E+06
Buso	T	Buso	38,48	420539	5,74E+06
Pantalón1	NT	Pantalón	30,87	169035	2,76E+06
Pantalón2	NT	Pantalón	22,85	151559	2,03E+06

Fuente Autores.

En la Tabla 16 se muestra una comparación de los valores de los descriptores resultantes para el proceso de clasificación, donde se observa que el pantalón es detectado como un buso ya que posee valores similares al de un buso real. Además, se muestran los valores cercanos que debe tener un pantalón para que el sistema lo clasifique correctamente.

Para corregir este problema, se tuvo que enfocar la cámara en diferentes posiciones para lograr que entrara más luz al obturador de la cámara, para así aclarar un poco más y resaltar el primer plano de la imagen, como se muestra a continuación:

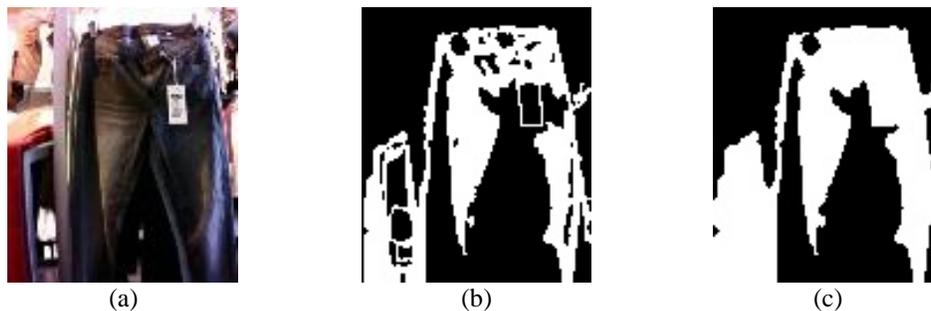


Figura 84. (a) Imagen original del pantalón, (b) Imagen segmentada de pantalón, (c) Resultado exitoso de relleno de la imagen segmentada.

Fuentes Autores.

A continuación, se muestra el resumen de los resultados de las imágenes segmentadas las cuales presenta la mala segmentación y la buena segmentación respectivamente (Véase la Figura 85).



*Figura 85. Resumen de la segmentación de las prendas para hombre.
Fuentes Autores*

4.3.2. Resultados de clasificación para prendas de mujer

Para esta prueba se obtuvo un total de muestras de 1597, donde el 1179 de las muestras fueron procesadas correctamente por el sistema. De la cantidad de muestras restantes se examinó que estas no fueron clasificadas correctamente debido a las mismas condiciones que se presentaron en

la anterior clasificación: la iluminación y el color del fondo de la imagen. Sin embargo, se obtuvo una buena clasificación para prendas como los vestidos. Por otro lado, con las demás clases no se obtuvieron los mismos resultados, debido a que la mayoría de ellas fueron reconocidos como vestidos. Esto se debió a que las otras clases poseían valores en sus descriptores bastante similares al vestido, como se ve en la Tabla 18. Otra razón es que las prendas como las blusas y las faldas comparten rasgos similares, como la forma y el área de esta, con el vestido, ya que estos se encuentran compuestos por una combinación de una blusa en la parte superior y una falda en la parte inferior.

Tabla 17. Estadísticas de las prendas para mujer que se obtuvo en la prueba de tienda de ropa.

PRENDA	TOTAL	NO CLASIFICO	SUBTOTAL
BUSO CON MANGA	246	70	176
BUSO SIN MANGA	242	46	196
MINIFALDA	198	39	159
FALDA	87	14	73
PANTALON	165	45	120
VESTIDO	659	204	455
TOTAL	1597	418	1179

Fuente Autores.

Tabla 18. Matriz de confusión de clasificación para prendas de mujer en la tienda ropa.

RECALL		VALOR PREDICHO (%)					
		BUSO CON MANGA	BUSO SIN MANGA	MINIFALDA	FALDA	PANTALON	VESTIDO
VALOR REAL	BUSO CON MANGA	26,70%	21,59%	0,00%	6,82%	13,64%	31,25%
	BUSO SIN MANGA	3,57%	38,78%	3,06%	15,31%	1,02%	38,27%
	MINIFALDA	9,43%	15,09%	11,95%	33,33%	10,06%	20,13%
	FALDA	6,85%	12,33%	1,37%	12,33%	6,85%	60,27%
	PANTALON	18,33%	5,83%	5,00%	8,33%	26,67%	35,83%
	VESTIDO	9,01%	5,27%	2,42%	15,82%	7,25%	60,22%

Fuente Autores.

A continuación, la Figura 86 se muestra el resumen de los resultados de las imágenes segmentadas las cuales presenta la mala segmentación y la buena segmentación respectivamente.



*Figura 86. Resumen de la segmentación de las prendas para mujer.
Fuentes Autores*

CONCLUSIONES

El sistema embebido ODROID XU4 ofrece unas especificaciones técnicas similares o mejores en comparación a sistemas como UDOO, BANANA PI, RADXA, ORANGE PI, RASPBERRY PI, DRAGONBOARD y BEAGLEBONE, aun así, puede conseguirse a un precio económico. Dentro de sus especificaciones, este posee dos puertos USB 3.0, lo cual favorece la transmisión de video proveniente del módulo de la cámara —característica que otros sistemas no ofrecen, a excepción del UDOO.

Para la cámara se utilizó el módulo de cámara *oCam* desarrollado por la misma empresa del sistema embebido, la cual ofrece imágenes en alta definición en diferentes resoluciones, lo cual facilitó el proceso. Además, se elige esta por motivos de compatibilidad, debido a que hay pocas cámaras web genéricas compatibles, la desventaja de estas es que no ofrecen una resolución y definición suficiente para obtener la información requerida de las prendas.

Además de la cámara *oCam*, la implementación del sensor ultrasónico favoreció el procesamiento de las imágenes, pues permitió que se ajustara la distancia entre la cámara y las prendas, para así mantener la proporción de esta, y reducir la complejidad en su reconocimiento.

Se decidió emplear un sistema operativo basado en Linux (Ubuntu MATE), ya que este requiere menos espacio en la memoria RAM, con el fin de que el restante sea reservado para los programas desarrollados en el sistema embebido. Estos programas fueron desarrollados en lenguaje de C++, ya que su compilación y depuración es mejor con respecto a otros lenguajes de computación, además requiere de menos comandos de acción para la interacción con los ajustes, tanto del sistema operativo como del sistema embebido.

En la evaluación e implementación de las técnicas de procesamiento de imágenes se analizaron tres técnicas: umbralización, segmentación de imágenes por color y textura (ISTC), y crecimiento de regiones.

Para el método de umbralización, se observó que, dentro del ambiente controlado, la técnica logró separar el primer plano de la imagen (la prenda de vestir) con el fondo de esta. En cambio, en el ambiente real (tiendas de ropa), no se obtuvieron los mismos resultados cuando se realizaron las pruebas, ya que en su diseño de entorno existían varios tonos de color y objetos que perturbaron la segmentación de la prenda.

En el algoritmo de ISTC se decidió usar solo la función de segmentación por color y no por textura, debido a que la técnica no logró distinguir la textura de la prenda con los demás elementos presentes dentro de la imagen. Con la segmentación por color se obtuvieron buenos resultados en el ambiente controlado y en las tiendas de ropa. A pesar de esto, en las prendas que contenían más de dos colores o poseían algún estampado, el algoritmo no determinó el color dominante de la prenda capturada. Además, al elegir píxeles al azar para la segmentación por color, el algoritmo aumentaba la duración de este proceso, y por lo tanto perjudica la transmisión del video.

Por último, se seleccionó el método de crecimiento de regiones, debido a que redujo componentes no deseados, que permitieron resaltar el primer plano donde se encuentra la prenda, además, se implementaron técnicas de realce de imágenes, como el aumento lineal de contraste y la selección de la capa RGB la cual detecta el tono más dominante de la prenda, con el objetivo de mejorar los resultados obtenidos en la segmentación. En los resultados se obtuvo una buena calidad de segmentación de la imagen, tanto en el ambiente controlado como en el real (tienda de ropa).

Además, presentó mejor desempeño en ejecuciones en tiempo real comparado con las otras técnicas.

En la extracción de características de prendas se implementaron varios métodos basados en esquemas de representación y descriptores, como los son: códigos de cadena, aproximación polinomial, firmas, esqueleto, descriptores Topológicos, Fourier y Momentos. Sin embargo, solo se enfocó en métodos como Firmas y Cerco Convexo, y el descriptor Momentos basado en regiones, ya que por medio de estos se pueden representar datos como el contorno y el área de la prenda en valores numéricos, permitiendo así diferenciarlas y, posteriormente, clasificarlas.

El sistema de clasificación se dividió en dos grupos: prendas para mujer, y para hombre, debido a que, a pesar de ser de distinta clase, existen prendas femeninas que se encuentran en la misma región de las masculinas. Por ejemplo, los valores de las blusas coinciden con los de las camisas. De este modo, esta división facilitó el reconocimiento de prendas para cada de género.

Para las prendas de hombre se presentó una dificultad en distinguirlas en el plano bidimensional, ya que algunas de estas que coincidieron en regiones donde no pertenecen. Por lo tanto, se adicionó una tercera variable de descriptores para formar una gráfica tridimensional. Esta se distribuyó en dos etapas: la primera consiste en usar dos clasificadores lineales para diferenciar prendas tipo "Torso" y "No torso"; la segunda consiste en un árbol de decisiones basado en cuatro clasificadores lineales, que además depende de valores de los descriptores de región para clasificar la prenda. Para el caso de las prendas de mujer, solo fueron necesarios cinco clasificadores lineales para distinguir las prendas en un plano bidimensional.

La validación del funcionamiento del sistema se dividió en dos etapas. En la primera, se adecuó un espacio con condiciones ideales, y con un número específico de prendas, para obtener una

precisión alta en el sistema de clasificación. En la segunda, se realizaron las pruebas del dispositivo en el ambiente real (tiendas de ropa). De esta validación se concluyó lo siguiente:

- Para las prendas de hombre, se tomaron 457 muestras en el ambiente controlado de las cuales se obtuvo una sensibilidad mayor que 90% en la primera fase de clasificación de hombre (Torso y No torso). Sin embargo, en la segunda fase, se encontró un 36.3% de sensibilidad en las camisas, debido a que sus contornos y formas son similares a las de busos y chaquetas. Y las demás prendas se encontró una sensibilidad mayor al 70%.

Sin embargo, dentro del ambiente real los resultados se presentaron de manera opuesta; ya que se encontró una sensibilidad menor que 50% en la segunda fase de clasificación, pues los colores en el primer plano eran muy similares a los del fondo de la imagen, haciendo que el área fuese mayor y que el sistema lo detectase como otra prenda. Esto mismo sucedió para “No torso” en la primera fase de la clasificación de prendas, por ello se obtuvo porcentajes de sensibilidad bajos.

- Para las prendas de mujeres, se realizaron 207 muestras en el ambiente controlado, en las cuales se encontró un promedio de 70% de sensibilidad, puesto que el contorno y la forma de vestido es similar a las demás —excepto al pantalón el cual tuvo una sensibilidad mayor que 95%.

Por otro lado, en el ambiente real (tiendas de ropa) se encontró ciertas dificultades en el reconocimiento de prendas femeninas, ya que la sensibilidad presentó un promedio de 30% de clasificación en las 1597 muestras. Esto hace que las blusas y las faldas se clasifiquen como los vestidos dado su contorno y forma.

De acuerdo con las pruebas de reconocimiento de prendas en las tiendas de ropa, se encontraron dificultades en la captura de las imágenes y en la movilidad dentro del entorno donde estaban

situadas las prendas. Esto se debió a que los establecimientos comerciales poseen diferentes diseños en sus áreas interiores, que afectan la posición, el color de la luz artificial y el color del fondo de las paredes y de los estantes, además de la forma de organización de las prendas y del espacio de movimiento dentro de la tienda.

BIBLIOGRAFÍA

- Balarini, J. P., & Nesmachnow, S. (2016). A C++ Implementation of Otsu's Image Segmentation Method. *Image Processing On Line (IPOL)*, (July), 1–10.
<https://doi.org/10.5201/ipol.2016.158>
- Cedano Olvera, M. A., Rubio González, J. A., & Vega Gutiérrez, A. C. (2014). *Fundamentos de computación para ingenieros*. (G. E. Patria, Ed.) (Primera ed). Mexico: Larousse - Grupo Editorial Patria. Retrieved from
<https://books.google.com.co/books?id=Kt3hBAAAQBAJ&printsec=frontcover#v=onepage&q&f=false>
- DANE Departamento Administrativo Nacional de Estadísticas. (2010). POBLACION CON REGISTRO PARA LA LOCALIZACION Y CARACTERIZACION DE LAS PERSONAS CON DISCAPACIDAD A NIVEL NACIONAL. Colombia: DANE Departamento Administrativo Nacional de Estadísticas. Retrieved from
http://www.dane.gov.co/files/investigaciones/discapacidad/Total_nacional.xls
- Dhall, A. (2015). Image Segmentation using Texture and Color features in C++. Retrieved February 21, 2018, from <https://github.com/ankitdhall/imageSegmentation>
- Enriquez Vásquez, R. A. (2013). Binarización. In *Sistema Móvil de Recuperación de Información Visual Utilizando Formas y Colores para el Reconocimiento de Obras Arquitectónicas*. (1st ed.). Puebla, Mexico: Universidad de las Américas Puebla. Retrieved from http://catarina.udlap.mx/u_dl_a/tales/documentos/1st/enriquez_v_ra/capitulo2.pdf
- Esqueda Elizondo, J. J., & Maestre Palafox, L. E. (2005). *Fundamentos de Procesamiento de Imágenes*. (L. E. Medina Gomez, Ed.) (1st ed.). Mexicali, Mexico: Universidad Autónoma de Baja California. Retrieved from
https://books.google.com.co/books?id=h4Gj8GuwPVkC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
- Gallagher, A. C., & Chen, T. (2008). Clothing cosegmentation for recognizing people. *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 1–8.
<https://doi.org/10.1109/CVPR.2008.4587481>
- Gonzalez, R. C. ., & Woods, R. E. (2002). *Digital Image Processing* (Second Edi). New Jersey: Prentice Hall. <https://doi.org/10.1049/ep.1978.0474>
- Graffigna, J. P. (2016a). Segmentación. In *Apuntes de cátedra* (1st ed., pp. 204–208). San Juan, Argentina: Universidad Nacional de San Juan. Retrieved from
<http://dea.unsj.edu.ar/imagenes/recursos/capitulo5.pdf>
- Graffigna, J. P. (2016b). Técnicas de Realce de Imágenes. In *Apuntes de cátedra* (1st ed., pp. 128–129). San Juan, Argentina: Universidad Nacional de San Juan. Retrieved from
<http://dea.unsj.edu.ar/imagenes/recursos/capitulo3.pdf>
- Hardkernel co. Ltd. (2015a). ODROID-XU4. Retrieved July 13, 2017, from

- http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143452239825
- Hardkernel co. Ltd. (2015b). USB Audio Adapter. Retrieved December 14, 2017, from http://www.hardkernel.com/main/products/prdt_info.php?g_code=G141629380416
- Hardkernel co. Ltd. (2016). oCam: 5MP USB 3.0 Camera. Retrieved July 15, 2017, from http://www.hardkernel.com/main/products/prdt_info.php?g_code=G145231889365
- ITEAD Intelligent Systems Co.Ltd. (2010). Ultrasonic ranging module : HC-SR04. Shenzhen, China. Retrieved from <http://www.electroschematics.com/wp-content/uploads/2013/07/HC-SR04-datasheet-version-2.pdf>
- Kearney, S. (2016). What are the best single-board computers? Retrieved December 14, 2017, from <https://www.slant.co/topics/1629/~single-board-computers>
- Marshall, D., & Martin, R. (1992). *Computer Vision, Models And Inspection*. World Scientific.
- Mary Jenitha, Jm., & Bashashaik, K. (2015). Possibilistic-Fuzzy C-Means Clustering Approach for the Segmentation of Satellite Images in HSL Color Space. *Procedia - Procedia Computer Science*, 57(57), 49–56. <https://doi.org/10.1016/j.procs.2015.07.364>
- OpenCV. (2014). Miscellaneous Image Transformations. Retrieved November 21, 2017, from https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html
- OpenCV. (2017). Image Thresholding. Retrieved November 21, 2017, from https://docs.opencv.org/3.3.1/d7/d4d/tutorial_py_thresholding.html
- Roy, R., Bommanakanti, V., & Doiche, B. (2015). User manual ODROID-XU4. *Hard Kernel, Ltd.* South Korea. Retrieved from <http://magazine.odroid.com/wp-content/uploads/odroid-xu4-user-manual.pdf>
- Sánchez Lopez, K. V. (2011). *Descriptores de imagenes digitales con momentos de Zernike*. UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO. Retrieved from <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/4682/tesis.pdf?sequence=1>.
- Theodoridis, S., & Koutroumbas, K. (2003). *Pattern Recognition* (2nd ed.). San Diego: Elsevier Academic Press.
- Valvano, J. (2014). *Embedded Systems - Introduction to ARM Cortex Microcontrollers* (5th ed.). Austin, Texas, U.S.A.: Jonathan Valvano. Retrieved from <http://users.ece.utexas.edu/~valvano/>
- Visa, S., Ramsay, B., Ralescu, A., & Van Der Knaap, E. (2011). Confusion matrix-based feature selection. *CEUR Workshop Proceedings*, 710(January), 120–127. <https://doi.org/10.1.1.666.8961>
- Yang, M., & Yu, K. (2011). REAL-TIME CLOTHING RECOGNITION IN SURVEILLANCE VIDEOS. *Ieee International Conference On Image Processing*, 2998–3001. <https://doi.org/10.1109/ICIP.2011.6116276>

Yuan, G., Ho, C., & Lin, C. (2012). Recent Advances of Large-scale Linear Classification. *Proceedings of the IEEE*, 100(9), 2584–2603.
<https://doi.org/10.1109/JPROC.2012.2188013>

ANEXO A

Resultados de los momentos obtenidas de las pruebas realizadas en el ambiente controlado

Tabla A1. Los momentos para el orden 0, 1, 2 y 3 de las prendas.

Prenda	m00	m01	m02	m03	m10	m11	m12	m20	m21	m30
Blusa0	81015	535109	4,17E+12	3,57E+14	383176	2,52E+12	1,95E+13	2,26E+12	1,49E+14	1,49E+14
Blusa1	11773	700964	5,56E+12	4,96E+14	582879	3,47E+12	2,75E+14	3,85E+12	2,29E+14	2,86E+14
Blusa2	7377	466630	3,59E+12	3,08E+14	352979	2,18E+12	1,64E+14	2,02E+12	1,25E+14	1,29E+14
Blusa3	11157	694134	5,55E+12	4,96E+14	529574	3,41E+12	2,74E+14	3,38E+12	2,23E+14	2,44E+13
Blusa4	7762	511608	4,02E+12	3,48E+14	391783	2,57E+12	2,00E+14	2,36E+12	1,56E+11	1,58E+14
Buso0	74405	507611	4,07E+11	3,59E+13	360875	2,45E+12	1,96E+14	2,10E+12	1,43E+14	1,35E+14
Buso1	7817	544131	4,45E+12	4,00E+14	373941	2,59E+12	2,11E+14	2,17E+12	1,52E+14	1,40E+13
Buso2	62535	436954	3,51E+12	3,04E+13	297664	2,06E+11	1,64E+14	1,72E+12	1,20E+14	1,10E+14
Buso3	703	355542	1,88E+11	1,02E+13	336432	1,70E+11	8,97E+12	1,65E+11	8,32E+12	8,21E+12
Buso4	13015	738623	4,33E+11	2,61E+13	596543	3,38E+11	1,98E+12	2,89E+11	1,63E+13	1,46E+12
Camisa0	8037	524685	4,06E+12	3,46E+14	373666	2,43E+12	1,88E+14	2,18E+11	1,43E+14	1,42E+14
Camisa1	8239	560796	4,54E+12	4,05E+14	393557	2,64E+12	2,11E+14	2,31E+12	1,55E+14	1,51E+14
Camisa2	11769	700490	5,56E+12	4,96E+14	582801	3,47E+12	2,75E+13	3,85E+11	2,29E+14	2,86E+14
Camisa3	115915	684466	5,37E+11	4,74E+13	577781	3,43E+11	2,71E+14	3,82E+12	2,27E+14	2,84E+14
Camisa4	82685	563777	4,56E+12	4,06E+14	398992	2,69E+12	2,15E+14	2,36E+12	1,60E+13	1,56E+14
Camiseta0	7050	492204	3,99E+12	3,57E+14	368061	2,58E+11	2,10E+14	2,25E+12	1,57E+14	1,51E+13
Camiseta1	11212	652332	5,02E+11	4,34E+13	554456	3,23E+12	2,48E+14	3,69E+12	2,16E+14	2,77E+13
Camiseta2	5679	346070	2,48E+12	1,99E+14	289225	1,74E+12	1,24E+14	1,67E+12	9,90E+13	1,05E+14
Camiseta3	6504	416933	3,21E+12	2,75E+14	336183	2,16E+12	1,67E+14	1,97E+12	1,26E+14	1,25E+14
Camiseta4	70065	422750	3,20E+12	2,74E+14	359232	2,12E+12	1,58E+14	2,10E+12	1,21E+14	1,35E+14
Chaqueta0	93555	650592	5,35E+12	4,84E+14	461382	3,19E+11	2,61E+14	2,91E+12	2,04E+14	2,05E+14
Chaqueta1	86005	580511	4,66E+12	4,12E+14	414792	2,76E+11	2,20E+14	2,52E+12	1,70E+14	1,71E+14
Chaqueta2	7871	508438	3,92E+12	3,34E+14	370586	2,36E+12	1,80E+14	2,19E+12	1,41E+14	1,44E+14
Chaqueta3	75235	490118	3,81E+12	3,26E+13	366094	2,37E+11	1,83E+13	2,15E+12	1,40E+14	1,40E+14
Chaqueta4	81605	556641	4,51E+12	4,02E+14	386024	2,60E+12	2,09E+14	2,25E+12	1,53E+14	1,46E+14
Corto0	51335	310430	2,06E+12	1,47E+14	261042	1,58E+12	1,05E+14	1,59E+12	9,72E+13	1,08E+13
Corto1	5179	325834	2,25E+12	1,67E+14	265572	1,67E+12	1,16E+14	1,62E+12	1,03E+14	1,10E+14
Corto2	55855	353738	2,49E+12	1,88E+13	280005	1,78E+12	1,26E+14	1,69E+12	1,09E+13	1,13E+14
Corto3	52585	313055	2,04E+12	1,42E+14	263038	1,57E+12	1,03E+14	1,62E+12	9,82E+13	1,11E+14
Corto4	59265	374352	2,65E+10	2,03E+14	296338	1,89E+12	1,35E+14	1,81E+12	1,18E+14	1,23E+14
Corto50	3136	186598	1,25E+12	9,04E+13	163246	9,79E+11	6,58E+13	8,97E+11	5,42E+13	5,16E+13
Corto51	28445	150320	9,27E+11	6,29E+13	146899	7,56E+11	4,57E+13	8,00E+11	4,01E+13	4,56E+13
Corto52	2512	137139	8,43E+11	5,61E+13	123033	6,60E+11	4,01E+13	6,33E+11	3,34E+13	3,39E+13
Corto53	2627	140552	8,46E+11	5,49E+13	133849	7,12E+11	4,27E+13	7,17E+11	3,81E+13	4,01E+13
Corto54	3590	209921	1,42E+12	1,06E+14	186632	1,09E+12	7,44E+13	1,03E+12	6,07E+13	6,01E+13
Falda0	66885	417313	3,04E+12	2,44E+14	331438	2,06E+11	1,50E+14	1,95E+12	1,22E+14	1,27E+14
Falda1	67765	442450	3,46E+12	3,00E+14	345586	2,27E+12	1,78E+13	2,01E+11	1,33E+14	1,28E+14
Falda2	60095	357907	2,44E+12	1,81E+14	283949	1,69E+12	1,15E+14	1,63E+12	9,73E+13	1,05E+14
Falda3	73125	456709	3,36E+12	2,71E+13	376449	2,35E+12	1,72E+14	2,34E+11	1,48E+13	1,61E+14
Falda4	48955	274432	1,77E+11	1,24E+14	241327	1,35E+12	8,65E+12	1,36E+12	7,62E+12	8,43E+13
Pantalon0	69905	461352	3,61E+12	3,12E+14	364629	2,41E+12	1,89E+14	2,23E+11	1,49E+14	1,50E+14
Pantalon1	69325	454874	3,52E+12	3,03E+13	361393	2,36E+12	1,82E+14	2,30E+12	1,54E+14	1,63E+14
Pantalon2	68645	452369	3,53E+11	3,05E+14	364624	2,39E+12	1,85E+14	2,36E+12	1,58E+14	1,69E+14

Pantalon3	67765	441300	3,45E+12	3,00E+13	347627	2,25E+12	1,75E+14	2,12E+12	1,40E+14	1,43E+14
Pantalon4	6986	457311	3,56E+11	3,08E+14	366048	2,36E+11	1,82E+14	2,34E+12	1,55E+14	1,67E+14
Pantalon51	41545	257141	1,99E+12	1,73E+14	217205	1,34E+11	1,02E+14	1,19E+12	7,25E+13	6,75E+13
Pantalon52	43855	264057	2,03E+12	1,75E+14	226350	1,34E+12	1,01E+14	1,23E+12	7,18E+13	6,99E+12
Pantalon53	40385	258883	2,04E+12	1,79E+13	204263	1,26E+12	9,62E+13	1,09E+11	6,50E+13	6,05E+13
Pantalon54	4370	281270	2,22E+12	1,95E+13	224849	1,41E+11	1,09E+14	1,22E+11	7,49E+13	6,94E+13
Pantalon55	38255	231452	1,77E+12	1,52E+14	186712	1,10E+12	8,27E+13	9,52E+11	5,50E+13	5,05E+13

Tabla A2. *Los momentos centrales de las prendas.*

Prenda	mu02	mu03	mu11	mu12	mu20	mu21	mu30
Blusa0	6,33E+11	-1,75E+12	-1,44E+05	1,37E+11	4,47E+11	1,55E+12	-6,91E+05
Blusa1	1,39E+12	-3,91E+05	-6,81E+05	2,64E+05	9,62E+10	-9,96E+05	-5,03E+05
Blusa2	6,34E+11	5,85E+10	-4,94E+05	-1,74E+12	3,34E+09	1,32E+12	-1,26E+11
Blusa3	1,23E+12	-2,30E+12	1,13E+11	-3,32E+12	8,67E+11	2,29E+10	1,63E+12
Blusa4	6,45E+11	-2,14E+12	-1,43E+05	-5,90E+11	3,81E+11	2,00E+12	2,46E+11
Buso0	6,06E+11	-1,51E+12	-1,52E+05	3,02E+11	3,46E+11	1,42E+12	-3,45E+11
Buso1	6,64E+11	-1,88E+12	-1,51E+05	9,41E+05	3,83E+11	1,80E+12	-2,34E+11
Buso2	4,54E+11	-4,34E+12	-2,01E+05	2,54E+11	3,00E+11	1,70E+12	-5,88E+11
Buso3	7,81E+04	-3,42E+05	-1,25E+05	2,89E+05	3,56E+05	1,21E+05	-3,32E+05
Buso4	1,36E+05	6,55E+05	-2,73E+05	4,80E+05	1,52E+05	-5,45E+05	-1,40E+05
Camisa0	6,32E+11	-1,55E+12	-8,31E+05	4,37E+11	4,41E+11	1,71E+12	7,38E+05
Camisa1	7,21E+11	-2,48E+12	-3,80E+05	-7,23E+09	4,26E+11	2,08E+12	-2,95E+05
Camisa2	1,39E+12	-2,13E+04	5,19E+04	5,38E+05	9,61E+11	-2,91E+05	-7,88E+05
Camisa3	1,33E+11	2,42E+05	2,08E+05	8,41E+11	9,44E+11	-5,47E+11	-2,93E+11
Camisa4	7,17E+11	-2,84E+12	-3,11E+05	-4,91E+11	4,34E+11	2,17E+12	-1,31E+09
Camiseta0	5,58E+10	3,98E+10	7,74E+05	-7,65E+05	3,26E+11	-8,64E+11	-1,44E+05
Camiseta1	1,22E+12	-6,62E+05	3,43E+05	-7,59E+05	9,51E+10	7,28E+11	1,39E+11
Camiseta2	3,76E+11	1,42E+12	-1,99E+05	-2,52E+11	1,96E+11	-7,51E+11	2,43E+10
Camiseta3	5,33E+11	1,66E+12	7,14E+04	-5,39E+05	2,31E+11	-7,58E+11	-1,31E+11
Camiseta4	6,46E+11	3,59E+12	-4,55E+05	-3,47E+11	2,60E+11	-1,20E+12	3,65E+11
Chaqueta0	8,27E+11	-3,63E+12	-2,28E+05	5,00E+10	6,30E+11	4,13E+12	-4,45E+11
Chaqueta1	7,43E+11	-3,04E+12	-4,21E+05	3,89E+11	5,21E+11	3,89E+11	-7,43E+11
Chaqueta2	6,36E+11	-1,70E+11	-3,80E+05	8,58E+05	4,42E+11	3,43E+12	-7,16E+11
Chaqueta3	6,12E+11	-1,98E+12	-1,96E+05	3,57E+11	3,66E+11	2,40E+12	-5,60E+11
Chaqueta4	7,13E+11	-2,43E+12	-3,49E+05	6,61E+11	4,27E+10	2,49E+12	-5,56E+11
Corto0	1,87E+11	-4,20E+11	-2,81E+05	1,59E+11	2,66E+11	1,08E+12	-1,21E+11
Corto1	2,01E+11	-2,49E+11	1,26E+05	-5,14E+05	2,62E+11	8,51E+11	1,61E+05
Corto2	2,46E+11	-5,57E+11	6,56E+05	4,47E+05	2,86E+11	1,06E+12	-7,70E+05
Corto3	1,77E+11	-3,47E+11	8,28E+05	1,19E+10	3,04E+11	9,39E+11	-1,06E+11
Corto4	2,89E+11	-9,02E+11	1,44E+05	7,78E+11	3,31E+11	1,82E+12	-7,18E+09
Corto50	1,38E+11	-2,23E+10	7,81E+05	-1,24E+11	4,67E+05	5,12E+05	3,79E+05
Corto51	1,33E+10	-1,27E+10	-2,07E+05	4,31E+05	4,10E+05	-4,22E+05	4,92E+05
Corto52	9,45E+05	-2,56E+11	-1,21E+05	1,25E+11	3,03E+05	8,15E+05	-7,00E+05

Corto53	9,36E+05	-3,45E+11	-3,85E+05	6,64E+05	3,53E+05	1,59E+11	-4,87E+05
Corto54	1,97E+11	-2,92E+11	3,25E+05	-4,59E+05	6,19E+05	5,72E+05	-2,97E+05
Falda0	4,38E+11	-7,31E+11	-4,12E+05	-3,87E+05	3,09E+11	6,89E+09	1,93E+05
Falda1	5,68E+11	6,89E+05	1,62E+05	-1,90E+11	2,46E+11	8,33E+05	8,10E+05
Falda2	3,05E+11	-4,33E+11	-2,54E+05	-1,10E+11	2,91E+11	3,33E+11	1,40E+11
Falda3	5,04E+11	-2,03E+12	-3,68E+04	-2,06E+11	3,98E+11	2,56E+12	8,65E+05
Falda4	2,29E+10	-3,81E+11	-5,95E+05	2,42E+05	1,74E+11	3,86E+11	-3,54E+05
Pantalon0	5,66E+11	-6,08E+11	9,86E+05	1,74E+11	3,29E+11	2,01E+12	-1,65E+11
Pantalon1	5,37E+11	1,05E+12	-1,40E+05	-6,08E+05	4,18E+10	4,38E+12	-1,99E+11
Pantalon2	5,45E+11	9,95E+11	-1,47E+05	-6,01E+05	4,19E+11	4,60E+12	-6,22E+11
Pantalon3	5,72E+10	7,98E+11	-9,72E+05	-2,20E+05	3,34E+11	2,86E+12	-1,72E+05
Pantalon4	5,67E+11	7,88E+11	-3,35E+05	-2,43E+11	4,25E+11	4,97E+12	-4,77E+11
Pantalon51	3,99E+11	4,43E+11	-8,88E+05	-7,95E+11	5,11E+05	-6,95E+05	9,21E+05
Pantalon52	4,35E+11	1,13E+12	-2,09E+05	-1,03E+12	6,15E+05	-9,89E+05	5,37E+05
Pantalon53	3,77E+11	-2,43E+11	-4,87E+05	-5,82E+11	5,40E+05	2,01E+11	1,93E+05
Pantalon54	4,08E+11	-5,91E+11	-3,84E+05	-6,67E+11	6,38E+05	2,53E+11	3,36E+05
Pantalon55	3,65E+11	1,29E+12	-2,61E+05	-3,10E+11	4,09E+05	-9,76E+05	4,93E+05

Tabla A3. *Los momentos normalizados de las prendas.*

Prenda	nu02	nu03	nu11	nu12	nu20	nu21	nu30
Blusa0	964551	-295528	-218815	232019	680841	263167	-116913
Blusa1	100105	-0,260149	-4,91146	0,17578	693871	-0,662603	-0,334277
Blusa2	116421	125172	-908659	-373104	614294	283169	-269333
Blusa3	988633	-175137	910478	-252735	69671	174319	124196
Blusa4	107008	-402771	-238118	-11117	632343	376029	463853
Buso0	109506	-315626	-274084	632415	624256	296982	-721969
Buso1	108624	-34754	-246693	174179	62653	333724	-433745
Buso2	116164	-140294	-513489	821632	766836	550732	-190163
Buso3	158121	-26124	-253289	220847	720556	926068	-253162
Buso4	805005	107201	-161349	784911	895561	-89229	-229102
Camisa0	978147	-267577	-128593	755023	682901	295101	1,27495
Camisa1	106195	-401952	-559959	-117374	628248	336929	-4,78282
Camisa2	100073	-0,141487	0,37469	3,58241	693866	-1,93913	-0,524359
Camisa3	989783	1,67162	154596	581224	702689	-377814	-202293
Camisa4	104813	-456373	-455609	-789838	634505	349641	-210559
Camiseta0	112286	952954	155703	-1,83422	656898	-207075	-3,46056
Camiseta1	97219	-4,97654	2,72774	-5,69939	756255	547094	104229
Camiseta2	116505	583827	-615596	-103804	608966	-308946	997981
Camiseta3	125957	487013	168779	-157944	545211	-222118	-384965
Camiseta4	131629	87442	-926805	-844515	530286	-292524	889322
Chaqueta0	945278	-4292	-260486	590659	720108	488336	-525421
Chaqueta1	100431	-44249	-569622	567788	703766	567206	-108382

Chaqueta2	102697	-310169	-613902	156152	713016	624237	-130245
Chaqueta3	108175	-404282	-346639	726445	646539	488721	-114023
Chaqueta4	107037	-403353	-523561	109885	641816	413554	-923798
Corto0	708953	-222682	-106582	842861	100969	572753	-642139
Corto1	750784	-12886	471071	-266439	978452	440787	8,32072
Corto2	787807	-238887	21019	191787	916555	452775	-330405
Corto3	639721	-17317	299473	594311	110084	46836	-526751
Corto4	823881	-333573	410373	287653	942063	672664	-265687
Corto50	140046	-40439	794515	-224657	475353	929143	688257
Corto51	164463	-294715	-256109	999237	506239	-978504	114002
Corto52	149806	-80974	-192359	394743	479599	25772	-221369
Corto53	135702	-974478	-557331	187689	511748	450077	-137595
Corto54	153074	-378157	251944	-594072	480146	740749	-383987
Falda0	979182	-199857	-921363	-105889	690927	18843	0,052657
Falda1	123603	18234	351882	-503714	536543	220328	214186
Falda2	843357	-154731	-704688	-391715	805454	118947	501668
Falda3	942614	-444289	-688596	-450971	744554	558991	189225
Falda4	955899	-227113	-248234	144521	724169	230231	-210835
Pantalon0	115859	-148704	201689	425164	672586	492403	-403478
Pantalon1	111752	263541	-290787	-151944	870149	109566	-497983
Pantalon2	115723	254936	-311282	-154054	889332	117815	-159203
Pantalon3	124542	211042	-211759	-5,80945	727493	757274	-4,55872
Pantalon4	116136	193118	-686167	-595042	870009	121933	-116939
Pantalon51	23142	397835	-514736	-714726	296187	-6,24758	827729
Pantalon52	226315	885681	-108687	-80656	32002	-776461	421748
Pantalon53	231085	-234362	-298872	-561753	330858	19425	186598
Pantalon54	213505	-467845	-201166	-52851	334064	200273	266359
Pantalon55	249481	142547	-178458	-342118	279812	-107807	544595

ANEXO B

Códigos de programación del sistema desarrollado

Los códigos de programación desarrollados para este proyecto se encuentran disponibles en la carpeta “ANEXO B” dentro del medio digital adjunto a este documento, allí se podrá examinar dos proyectos: el primer proyecto consiste en la programación realizada en el software de Visual Studio para Windows, y el segundo contiene todos los archivos necesarios para su ejecución en SO Linux, este solo puede ser ejecutado en un sistema embebido ODROID XU4.

ANEXO C

Instalación y configuración del software de Visual Studio y OpenCV

Dentro de la carpeta “ANEXO C” del medio digital se encuentra: el manual de instalación y configuración del software de Visual Studio y OpenCV para sistemas operativos Windows y los archivos ejecutables necesarios para la instalación.

ANEXO D

Librería de muestras adquiridas.

En la carpeta “ANEXO D” dentro del medio digital se encuentra todas las muestras adquiridas y utilizadas por el sistema para su procesamiento, además se pueden examinar las imágenes de los resultados obtenidos de los métodos de segmentación para su posterior clasificación. Además se incluye un archivo de texto donde se explica una introducción referente a las etiquetas usadas en las imágenes de muestra y de los resultados obtenidos.

ANEXO E

Pruebas visuales del funcionamiento del sistema

En la carpeta “ANEXO E” se encuentra archivos de video donde se muestra el comportamiento que presente el sistema dentro del sistema operativo, como realiza su procesamiento y los resultados que se obtienen.