



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES

**“DISEÑO Y CONSTRUCCIÓN DE UN DISPOSITIVO PORTATIL
PARA LA IDENTIFICACIÓN DE BILLETES ORIENTADO A
PERSONAS CON DISCAPACIDAD VISUAL MEDIANTE EL USO
DE VISIÓN ARTIFICIAL”**

TRABAJO DE TITULACIÓN: DISPOSITIVO TECNOLÓGICO
Para optar al Grado Académico de:
INGENIERA EN ELECTRÓNICA, CONTROL Y REDES
INDUSTRIALES

AUTORA: SOLANO AGUILAR GABRIELA DEL CISNE

TUTOR: ING. LOZADA YANEZ PABLO EDUARDO

Riobamba-Ecuador

2018

©2018, Solano Aguilar Gabriela del Cisne

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES

El Tribunal de Tesis certifica que: El dispositivo tecnológico: “DISEÑO Y CONSTRUCCIÓN DE UN DISPOSITIVO PORTATIL PARA LA IDENTIFICACIÓN DE BILLETES ORIENTADO A PERSONAS CON DISCAPACIDAD VISUAL MEDIANTE EL USO DE VISIÓN ARTIFICIAL”, de responsabilidad de la señorita Gabriela del Cisne Solano Aguilar, ha sido minuciosamente revisado por los Miembros del Tribunal de Tesis, quedando autorizada su presentación.

Dr. Julio Santillán C.

VICEDECANO DE LA FACULTAD DE
INFORMÁTICA Y ELECTRÓNICA

Ing. Freddy Chávez V.

DIRECTOR DE ESCUELA DE
INGENIERÍA ELECTRÓNICA EN
CONTROL Y REDES INDUSTRIALES

Ing. Pablo Lozada Y.

DIRECTOR DEL TRABAJO DE
TITULACIÓN

Ing. Fausto Cabrera A.

MIEMBRO DEL TRIBUNAL

Yo, Gabriela del Cisne Solano Aguilar soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual del Trabajo de Titulación pertenece a la Escuela Superior Politécnica de Chimborazo.

Gabriela del Cisne Solano Aguilar

Estudiante

DEDICATORIA

A mis padres que siempre me han apoyado y han sido un pilar fundamental para mi vida, a mi alma gemela, mi mejor amigo y compañero de vida que nunca ha dejado de creer en mí y me ha apoyado en todo momento con su amor, paciencia y entendimiento, a la luz más inmensa que brilla en mi vida, mi hijo, quién cada día alumbra mi alma y me hace sentir la persona más bendecida del mundo. A mi familia, amigos y maestros.

Gabriela del Cisne Solano Aguilar

AGRADECIMIENTO

Le agradezco principalmente a Dios que me ha dado la vida, la salud y me ha permitido llegar hasta este momento. A mi hijo, mis padres, suegros, cuñados y el amor de mi vida que me supieron dar palabras reconfortantes en momentos de frustración y que se han convertido en parte fundamental de apoyo para la culminación de este trabajo de titulación

A la Escuela Superior Politécnica de Chimborazo, por darme la oportunidad de seguir con mis estudios con una formación académica con calidad, prestigio y sabiduría. A cada uno de mis maestros especialmente al Ing. Pablo Lozada que me guio en la realización de este trabajo de titulación.

Gabriela del Cisne Solano Aguilar

ÍNDICE DE ABREVIATURAS

A	Amperios
CAD	Computer-Aided Desing (Diseño asistido por computador)
CCD	Charge Coupled Device (Dispositivo de carga acoplada)
cm	Centímetros
CMOS	Complementary Metal Oxide Semiconductor (Semiconductor Complementario de Óxido Metálico)
GND	Ground (Tierra)
GPIO	General Purpose Input Output (Puerto de Entrada/Salida de Propósito General)
h	Horas
LBP	Local Binary Pattern (Patrones Binarios Locales)
Ma	Mili Amperios
mAh	Mili Amperios Hora
mm	Milímetros
s	Segundos
OCR	Optical Character Recognition (Reconocimiento Óptico de Caracteres)
OMS	Organización Mundial de la Salud
OpenCV	Open Source Computer Vision (Visión por computadora de código abierto)
RGB	Red, Green, Blue (Rojo, Verde, Azul)
SD	Secure Digital (Seguro Digital)
ULBP	Uniform Local Binary Pattern (Patrones Binarios Locales Uniformes)
USB	Universal Serial Bus
V	Voltaje
VCC	Voltaje de Corriente Continua
2D	Dos dimensiones
3D	Tres dimensiones
W	Vatios

TABLA DE CONTENIDO

DEDICATORIA	v
AGRADECIMIENTO	vi
ÍNDICE DE ABREVIATURAS	vii
ÍNDICE DE FIGURAS	xi
ÍNDICE DE GRÁFICOS	xiii
ÍNDICE DE TABLAS	xiv
RESUMEN	xv
SUMMARY	xvi
INTRODUCCIÓN	1

CAPÍTULO I

1. MARCO TEÓRICO REFERENCIAL	5
1.1 Discapacidad Visual	5
1.1.1 Definición	5
1.1.2 Problemas y dificultades de las personas con discapacidad visual con el entorno	6
<i>1.1.2.1 Movilidad</i>	<i>6</i>
<i>1.1.2.2 Orientación</i>	<i>6</i>
<i>1.1.2.3 Comunicación</i>	<i>7</i>
<i>1.1.2.4 Reconocimiento de objetos</i>	<i>7</i>
1.1.3 Dispositivos de detección de dinero orientado personas con discapacidad visual	9
1.2 Inteligencia artificial	10
1.2.1 Definición	10
1.2.2 Visión Artificial	10
<i>1.2.2.1 Sistema de visión artificial</i>	<i>11</i>
<i>1.2.2.2 Detección de objetos</i>	<i>14</i>
<i>1.2.2.3 Local Binary Pattern</i>	<i>15</i>
<i>1.2.2.4 Template Matching</i>	<i>19</i>
1.2.3 Aprendizaje automático	20
<i>1.2.3.1 Aprendizaje supervisado y no supervisado</i>	<i>20</i>
<i>1.2.3.2 Regresión Logística</i>	<i>22</i>
1.3 Hardware	25
1.3.1 Raspberry pi 3	25

1.3.1.1	<i>Características</i>	26
1.3.2	<i>Cámara Raspberry Pi</i>	26
1.3.2.1	<i>Caractrísticas</i>	27
1.3.3	<i>Altavoz</i>	27
1.3.4	<i>Batería externa</i>	28
1.3.4.1	<i>Características de una batería externa de 3000mAh</i>	29
1.4	Software	29
1.4.1	<i>Sistema Operativo Raspbian</i>	29
1.4.2	<i>Python</i>	30
1.4.3	<i>OpenCV</i>	30
1.4.4	<i>SolidWorks</i>	31

CAPÍTULO II

2.	MARCO METODOLÓGICO	32
2.1	Hardware	32
2.1.1	<i>Componentes</i>	33
2.1.2	<i>Circuitos electrónicos</i>	34
2.1.2.1	<i>Alimentación del dispositivo</i>	34
2.1.2.2	<i>Circuito pulsador</i>	35
2.1.2.3	<i>Circuito altavoz</i>	35
2.1.2.4	<i>Esquema de conexiones al puerto GPIO de la raspberry pi 3</i>	36
2.1.2.5	<i>Esquema electrónico del dispositivo 3D</i>	36
2.1.3	<i>Diseño del dispositivo</i>	37
2.2	Software	40
2.2.1	<i>Adquisición de imágenes</i>	41
2.2.2	<i>Aprendizaje computacional para la clasificación Billete vs Fondo</i>	42
2.2.2.1	<i>Creación del conjunto de entrenamiento</i>	42
2.2.2.2	<i>Extracción de características del conjunto de entrenamiento</i>	44
2.2.2.3	<i>Entrenamiento y aplicación de Regresión Logística</i>	50
2.2.2.4	<i>Determinación del mejor umbral para la clasificación y segmentación del billete</i>	53
2.2.3	<i>Identificación de las denominaciones</i>	59
2.2.4	<i>Difusión de los resultados mediante comandos de voz</i>	66
2.2.5	<i>Diagrama de bloques del funcionamiento del dispositivo</i>	66

CAPÍTULO III

3.	MARCO DE RESULTADOS	68
-----------	----------------------------------	-----------

3.1	Desempeño de la identificación de denominación de billetes	68
3.1.1	<i>Características de las imágenes tomadas para las pruebas</i>	68
3.1.2	<i>Contraste sobre la igualdad de varias proporciones</i>	73
3.1.2.1	<i>Distancias de no funcionamiento del dispositivo</i>	75
3.1.2.2	<i>Distancias de funcionamiento del dispositivo.....</i>	76
3.2	Tiempo de respuesta.....	79
3.3	Consumo de energía del dispositivo.....	81
3.4	Análisis de costos.....	84
3.4.1	<i>Costos de materiales y herramientas</i>	84
3.4.2	<i>Costos de equipos</i>	85
3.4.3	<i>Costos de implementación</i>	85
3.4.4	<i>Costo total de construcción del dispositivo.....</i>	85
	CONCLUSIONES	87
	RECOMENDACIONES	88
	BIBLIOGRAFÍA	
	ANEXOS	

ÍNDICE DE FIGURAS

Figura 1-1:	Tablilla de billetes para personas con discapacidad.....	8
Figura 2-1:	Note Teller 2	9
Figura 3-1:	Ejemplos de iluminación en el campo industrial	12
Figura 4-1:	Sensor de imagen con y sin lente	12
Figura 5-1:	Proceso de digitalización de imágenes	13
Figura 6-1:	Imágenes aplicando escala de grises y disminución de ruido.....	13
Figura 7-1:	Ejemplo de segmentación	14
Figura 8-1:	Ejemplo de reconocimiento de objetos	14
Figura 9-1:	Reconocimiento de peatones	15
Figura 10-1:	Aplicación del descriptor LBP a una imagen de intensidad	16
Figura 11-1:	Proceso de obtención de los valores de Local Binary Pattern	17
Figura 12-1:	Patrones binarios locales Uniformes	18
Figura 13-1:	Ejemplos de patrones LBP uniforme y no uniforme	18
Figura 14-1:	Ejemplo de template matching con medidas de disimilitud	19
Figura 15-1:	Ejemplo de template matching con medidas de similitud	20
Figura 16-1:	Ejemplo de dos clases separables	21
Figura 17-1:	Ejemplo de clases no separables linealmente.....	21
Figura 18-1:	Clasificación de dos clases con regresión logística	223
Figura 19-1:	Regresión logística como neurona artificial.....	24
Figura 20-1:	Raspberry pi 3 Modelo B	25
Figura 21-1:	Cámara Raspberry pi v2	27
Figura 22-1:	Altavoz de 0.25W	28
Figura 23-1:	Ejemplo de batería externa y sus partes	28
Figura 24-1:	Escritorio del sistema operativo Raspbian	29
Figura 25-1:	Logo del lenguaje de programación Python.....	30
Figura 26-1:	Logo OpenCV	31
Figura 27-1:	Logo SolidWorks	31
Figura 1-2:	Diagrama esquemático de la alimentación del dispositivo	34
Figura 2-2:	Circuito pulsador	35
Figura 3-2:	Circuito amplificador de audio	35
Figura 4-2:	Esquema de conexiones al puerto GPIO	36
Figura 5-2:	Esquema electrónico del sistema en 3D	37
Figura 6-2:	Diseño del compartimento para la placa Raspberry pi	38
Figura 7-2:	Compartimento para la batería del dispositivo.....	38

Figura 8-2: Diseño del compartimiento para la cámara Raspberry	39
Figura 9-2: Diseño del dispositivo.....	39
Figura 10-2: Diseño del sujetador de banda	39
Figura 11-2: Montaje del dispositivo.....	40
Figura 12-2: Dispositivo armado	40
Figura 13-2: Billetes de dólar de 1, 5, 10 y 20 dólares, frontales y posteriores	42
Figura 14-2: Ejemplos de imágenes positivas	43
Figura 15-2: Ejemplos de imágenes negativas o fondo	43
Figura 16-2: Imagen en escala de grises y aplicado LBP invariante a la rotación	44
Figura 17-2: Posiciones de los píxeles y orden de comparación	46
Figura 18-2: Ejemplo de rotación del número binario	47
Figura 19-2: Ejemplo de LBP invariante a la rotación	48
Figura 20-2: Imagen e histograma en escala de grises y aplicando LBP	48
Figura 21-2: Ejemplo de pirámide de imágenes	55
Figura 22-2: Ventana deslizante sobre una imagen aplicada LBP	56
Figura 23-2: Ejemplo de una captura de imagen.....	58
Figura 24-2: Ejemplo de varias detecciones en una misma imagen	58
Figura 25-2: Ejemplo de segmentación del billete.....	59
Figura 26-2: Imagen en RGB y detección de bordes	59
Figura 27-2: Plantillas usadas para la diferenciación de denominaciones	62
Figura 28-2: Plantillas de 1, 5, 10 y 20 dólares frontales y posteriores a 0°, 5°, -5°, 10°, -10°, 15°, .. -15° grados	63
Figura 29-2: Ejemplo de detección de 1 dólar frontal.....	65
Figura 1-3: Algunas imágenes de prueba	69
Figura 2-3: Funcionamiento de las distancias entre el dispositivo y el billete a analizar	78

ÍNDICE DE GRÁFICOS

Gráfico 1-1: Flujograma de un sistema de reconocimiento de dinero	10
Gráfico 2-1: Histograma de una imagen aplicando LBP y ULBP	17
Gráfico 3-1: Función logística o sigmoide	22
Gráfico 1-2: Diagrama de bloques del dispositivo	33
Gráfico 2-2: Desarrollo del sistema detección de billetes y su denominación	41
Gráfico 3-2: Diagrama de flujo del descriptor LBP invariante a la rotación	45
Gráfico 4-2: Proceso de extracción de características	49
Gráfico 5-2: Diagrama de flujo de la regresión logística	50
Gráfico 6-2: Diagrama de flujo del entrenamiento	52
Gráfico 7-2: Muestras positivas y negativas en el primer entrenamiento	53
Gráfico 8-2: Imágenes positivas y negativas en el segundo entrenamiento	54
Gráfico 9-2: Diagrama de flujo para determinación del mejor umbral	55
Gráfico 10-2: Pruebas de imágenes para determinar el mejor umbral para la clasificación	57
Gráfico 11-2: Diagrama de flujo de función para obtener mejor valor de emparejamiento	61
Gráfico 12-2: Diagrama de flujo para determinar la denominación del billete detectado	64
Gráfico 13-2: Diagrama de flujo del funcionamiento del dispositivo	67
Gráfico 1-3: Identificaciones correctas e incorrectas a 16-17cm del billete al dispositivo	70
Gráfico 2-3: Identificaciones correctas e incorrectas a 18-19cm del billete al dispositivo	70
Gráfico 3-3: Identificaciones correctas e incorrectas a 20-21cm del billete al dispositivo	71
Gráfico 4-3: Identificaciones correctas e incorrectas a 22-23cm del billete al dispositivo	71
Gráfico 5-3: Identificaciones correctas e incorrectas a 24-25cm del billete al dispositivo	72
Gráfico 6-3: Identificaciones correctas e incorrectas a 26-27cm del billete al dispositivo	72
Gráfico 7-3: Identificaciones correctas e incorrectas a 28-29cm del billete al dispositivo	73
Gráfico 8-3: Identificaciones correctas e incorrectas a 30-31cm del billete al dispositivo	73
Gráfico 9-3: Proporciones obtenidas en los 8 rangos de análisis	75
Gráfico 10-3: Porcentaje de identificaciones correctas e incorrectas de 16 a 29cm	77
Gráfico 11-3: Pruebas de tiempo de respuesta y promedio en segundos	80

ÍNDICE DE TABLAS

Tabla 1-1: Especificaciones técnicas de la placa Raspberry pi 3 Modelo B	26
Tabla 2-1: Especificaciones técnicas más relevantes del módulo de cámara Raspberry v2	27
Tabla 3-1: Especificaciones técnicas de una batería externa de 3000mAh de 2 celdas	29
Tabla 1-2: Descripción de patrones uniformes y no uniformes	46
Tabla 1-3: Pruebas de identificación realizadas a distintas distancias del billete al dispositivo	70
Tabla 2-3: Tabla de proporciones dados intervalos de distancia	74
Tabla 3-3: Rango con las mejores identificaciones	76
Tabla 4-3: Tiempo invertido para la identificación de un billete	79
Tabla 5-3: Consumo de corriente del dispositivo en funcionamiento	81
Tabla 6-3: Consumo de corriente del dispositivo en reposo	82
Tabla 7-3: Costos de materiales y herramientas	84
Tabla 8-3: Costos de equipos	85
Tabla 9-3: Costos de implementación.....	85
Tabla 10-3: Costo total	86

RESUMEN

En el presente trabajo de titulación se desarrolló un dispositivo portátil capaz de detectar billetes y su denominación, orientando su uso a personas con discapacidad visual, emplea visión artificial y aprendizaje computacional. Para la realización de este prototipo se emplearon los métodos: deductivo, heurístico y experimental. Este cuenta con una fase de programación y otra de diseño en cuanto al hardware. La programación se llevó a cabo en Python, en él se desarrolló un clasificador lineal, regresión logística, consiguiendo diferenciar un billete del fondo de una imagen. Esto se logró con la extracción de características de un conjunto de entrenamiento de billetes y fondo, aplicando una variación del descriptor Local Binary Pattern que lo hace invariante a la rotación, luego se entrenó estos resultados usando regresión logística para obtener el modelo con más bajo coste error de todos los elementos del conjunto de entrenamiento. Dicho modelo se multiplica por cada ventana de la imagen aplicada el descriptor para comparar ese valor con cierto umbral y clasificarlas como billete o fondo. Una vez segmentado el billete de 1, 5, 10 o 20 dólares, se aplica la correlación para determinar el mejor emparejamiento entre el billete segmentado y las plantillas informando así la denominación del billete. En cuanto al hardware, se diseñó una carcasa que contiene una Raspberry pi 3, una cámara Raspberry pi, una batería de 3000mAh, un altavoz de 0.25W con su respectivo circuito, un switch y pulsador. El rango de distancia para obtener un buen número de identificaciones correctas está entre 16 y 29cm del billete a analizar al dispositivo, tomando un tiempo de procesamiento de 2,41 minutos aproximadamente. Es recomendable disminuir el tiempo empleado para la identificación de billetes sin afectar su desempeño así como adicionar otras técnicas o mecanismos para mejorar los resultados de la identificación.

Palabras clave: <TECNOLOGÍAS Y CIENCIAS DE LA INGENIERÍA>, <VISIÓN ARTIFICIAL>, <LOCAL BINARY PATTERN>, <APRENDIZAJE COMPUTACIONAL>, <REGRESIÓN LOGÍSTICA>, <CORRELACIÓN>

SUMMARY

In the current degree, work was developed in a portable device capable to detect bills and its denomination, guiding its use to people with visual impairment, employs artificial vision and computational learning. For the implementation of this prototype, the deductive, heuristic and experimental methods were applied. The same counts with a programming stage and other of design related to the hardware. The programming was carried out in Python, into it was developed a lineal classificator, logistic regression, obtaining to differentiate a bill from the bottom of an image. This was achieved with the extraction of characteristics of a set of training of bills and bottom, applying a variation of the descriptor Local Binary Pattern that do it invariant to the rotation, then these results were trained using logistic regression for obtaining the model with lower error cost of all the elements of training set. This model is multiplied by each window of applied image the descriptor in order to compare this value with certain threshold and classify them like bill or bottom. Once segmented the bill of 1,5,10 or 20 dollars, is applied the correlation for determining the better matching between the segmented bill and the templates informing in this way the bill denomination. Regarding to the hardware, a casing that contains a Raspberry pi 3 was designed, a Raspberry pi camera, a battery of 3000mAh, a loudspeaker of 0,25W with its respective circuit, a switch and button. The range of distance to get a good number of correct identifications is between 16 and 29 cm from the bill to analyse the device, taking a processing time of 2,41 minutes approximately. Is recommendable to reduce the employed time for the bills identification without affect its management such as to add other techniques or mechanisms to improve the identification results.

Key words: <TECHNOLOGY AND ENGINNERING SCIENCES>, <ARTIFICIAL VISION>, <LOCAL BINARY PATTERN>, <COMPUTATIONAL LEARNING>, <LOGISTIC REGRESSION>, <CORRELATION>

INTRODUCCIÓN

La vista es uno de los sentidos que posee el ser humano, se considera el más valioso, emplea los globos oculares para identificar, analizar, inspeccionar e interactuar con el entorno que nos rodea, sin embargo hay personas que no poseen la totalidad de su visión, según datos de la Organización Mundial de la Salud, en el 2014, alrededor de 285 millones de personas poseen discapacidad visual, este grupo se desagrega en personas ciegas y personas con baja visión. (OMS, 2014)

En Ecuador, según el Instituto Nacional de Estadísticas y Censos, INEC (Información Censal Cantonal) en el censo de población y vivienda del 2010, existen 186.117 personas con ceguera, 2.964 en la provincia de Chimborazo y 1.972 en Riobamba, por otro lado El Consejo Nacional para la Igualdad de Discapacidades en el 2016 informó que existen 51.600 personas con discapacidad visual en el país, 1.382 en Chimborazo y 724 en Riobamba. (CONADIS, 2016)

La inclusión de las personas con discapacidades, en áreas donde se desempeñan quienes no las tienen, es un punto de partida, para poder alcanzar igualdad de oportunidades y así explotar su potencial. Por ejemplo en el Ecuador, el Plan Nacional del Buen Vivir (2013-2017) estuvo compuesto por 12 objetivos, que buscaban el bienestar de la ciudadanía. “Objetivo 2. Auspiciar la igualdad, la cohesión, la inclusión y la equidad social y territorial en la diversidad”.

Las personas con discapacidad visual al estar vinculados en actividades de compra y/o venta de bienes y servicios, hacen uso del dinero, significándoles un desafío debido a su condición. Se han hecho algunos esfuerzos para ayudarlos como la modificación de monedas y billetes, o cursos para identificar el dinero mediante el tacto.

En países como México y Chile, los billetes han sido modificados en cuanto a longitud, color, marcas táctiles en cada denominación tornándolos identificables para personas con problemas de visión o ceguera total, mientras que el dólar (moneda oficial en Ecuador), la identificación de cada una de sus distintas denominaciones se torna difícil al no poseer características diferenciables, en cambio las monedas, al tener distinto diámetro, son más fáciles de identificar.

Esfuerzos para la identificación de billetes para personas con discapacidad visual ya se han desarrollado, en Colombia por ejemplo en el año 2008 se creó un dispositivo portátil basado en sensores magnéticos y ultravioleta, para la identificación de billetes de ese país y su autenticidad,

otros se han realizado mediante aplicaciones móviles, que ofrecen la identificación de cualquier billete de dólar usando la cámara del dispositivo, y expresan el resultado con comandos de voz.

A pesar del desarrollo de la tecnología, dispositivos y aplicaciones con accesibilidad para personas con problemas de visión, la adquisición de smartphones, tablets u otros dispositivos es un factor preponderante, pues para personas con recursos limitados (Según la OMS, las personas con discapacidades son vulnerables a la pobreza) la compra de estos dispositivos no es primordial.

El dispositivo portátil del cual se habla en el presente trabajo busca facilitar a las personas con problemas de visión o con ceguera, el reconocimiento de cada una de las denominaciones más usadas en el país: \$1, \$5, \$10 y \$20 dólares, empleando visión artificial y aprendizaje computacional, además de informar los resultados mediante comandos de voz.

JUSTIFICACIÓN TEÓRICA

Gobiernos, empresas y la industria en general se han interesado cada vez más por la inclusión de personas con discapacidades en el sector social y es que este interés puede y está cambiando la vida de muchas personas con esta condición, se han desarrollado sistemas operativos, dispositivos y otros, especialmente enfocados para el uso de personas con esta cualidad que les benefician, haciéndoles llevar una vida más inclusiva en el contexto de interacción con el entorno.

Un dispositivo llamado HandEyes creado en el Ecuador en el año 2016, por ejemplo, ganó el primer lugar del concurso realizado por The History Channel, Una Idea para cambiar la Historia, debido a su objetivo de implementación, pues permite mejorar la movilidad de personas con discapacidad visual.

En sistemas operativos como Android o iOS, usados en tablets o smartphones, buscan también ser cada vez más accesibles a las personas con discapacidades visuales, realizando aplicaciones exclusivas para ellos o acciones como lecturas de contenido, escribir mediante el uso de la voz o de identificación de dinero, no obstante, muchas personas con esta condición visual no pueden acceder a dichos dispositivos debido a recursos limitados o a desconocimiento.

El presente trabajo busca que las personas con discapacidad visual o aquellas con problemas de visión posean una alternativa para la identificación de la denominación de los billetes más utilizados en el país, billetes de \$1, \$5, \$10 y \$20 (Villalba Mateo, 2015), para ello se ha construido un dispositivo que utiliza visión artificial y procesamiento inteligente, con el propósito

de y mejorar las actividades financieras, tales manejo de dinero en efectivo en personas con problemas de visión.

JUSTIFICACIÓN APLICATIVA

Actualmente el campo de visión artificial, rama de la inteligencia artificial, se ha convertido en una herramienta poderosa para áreas tales como: alimentación, automatización industrial medicina, eléctrica, electrónica, entre otras, debido a sus aplicaciones, que permiten realizar distintas técnicas como identificación de objetos, medición de los mismos, identificador de colores, sensores de presencia, inspección de calidad, entre otros.

Además el implementar procesamiento inteligente o inteligencia artificial atrae varias ventajas tales como reducción de costos, ahorro de tiempo en tareas complejas o inclusión predicción de eventos.

En cuanto a la identificación de billetes a través de visión artificial, ya existen aplicaciones móviles tales como Google Goggles, Money Reader o LookTel Money Reader, que se pueden encontrar en Google Play, sin embargo hasta el momento no se ha identificado un dispositivo que use visión artificial, aprendizaje computacional y que esté orientado a personas con discapacidad visual.

El presente trabajo de titulación propone la realización de un dispositivo orientado a personas con problemas de visión, el cual es capaz de identificar la denominación de billetes de dólar, mediante el uso de visión artificial y procesamiento inteligente que le permite al dispositivo determinar las características del billete, para de este modo comunicar al usuario, luego de un análisis, la información necesaria acerca del billete.

OBJETIVOS

OBJETIVO GENERAL

Desarrollar un dispositivo portátil basado en visión artificial y procesamiento inteligente que permita la identificación de billetes orientado a personas con discapacidad visual

OBJETIVOS ESPECÍFICOS

- Emplear una metodología de procesamiento de Visión Artificial para realizar la identificación de un billete
- Optimizar el proceso de detección de billetes mediante una metodología de procesamiento inteligente
- Diseñar un dispositivo portátil que permita realizar la identificación de billetes orientado a personas con discapacidad visual a través de visión artificial y procesamiento inteligente

CAPÍTULO I

1. MARCO TEÓRICO REFERENCIAL

1.1 Discapacidad Visual

Según la Organización Mundial de la Salud (OMS) aproximadamente un 15% de la población mundial posee algún tipo de discapacidad, de los cuales se estima que 285 millones de personas poseen discapacidad visual, con una segmentación de 39 millones de personas consideradas no videntes y 246 millones con baja visión. (OMS,2014)

1.1.1 *Definición*

Antes de definir lo que es discapacidad visual, se debe entender lo que es la discapacidad, “Una discapacidad es una condición que hace que una persona sea considerada como discapacitada. Esto quiere decir que el sujeto en cuestión tendrá dificultades para desarrollar tareas cotidianas y corrientes que, al resto de los individuos, no les resultan complicadas. El origen de una discapacidad suele ser algún trastorno en las facultades físicas o mentales” (Pérez y Merino, 2012)

Esta definición refleja lo que se ha podido observar en la vida cotidiana, las personas con algún tipo de discapacidad poseen cierto grado de limitación o dificultad en el desarrollo de sus actividades diarias, sin embargo, a pesar de ser de conocimiento general de esta condición, no reciben los tratamientos o trato ideal y necesario para su normal desenvolvimiento.

Existen facultades físicas y mentales que condicionan los tipos de discapacidad, en el Ecuador se han dividido en 6 tipos de discapacidades: auditiva, física, intelectual, lenguaje, psicosocial y visual, cada una de estas se subdivide en 4 grados: leve, moderado, grave y muy grave. Cabe mencionar que la discapacidad, puede adquirirse desde el nacimiento o a lo largo de la vida.

La discapacidad visual también es conocida como déficit visual o visión reducida, es la disminución de agudeza visual, condición que afecta de mayor o menor grado a los ojos, según la Dirección Nacional de Discapacidades hay que diferenciar entre una enfermedad y la discapacidad en sí, la enfermedad posee, mientras que para la discapacidad no existen tratamientos que puedan curarla, por lo que se vuelve permanente. (Dirección Nacional de Discapacidades)

1.1.2 Problemas y dificultades de las personas con discapacidad visual con el entorno

La misma definición de discapacidad expresa que las personas que la sufren, estarán ligadas a experimentar dificultades o limitaciones en las actividades de la vida cotidiana, por lo que se puede tener una idea de cómo las personas con discapacidad visual perciben el mundo, el cual presenta muchísimos obstáculos, debido a que el entorno está diseñado y construido mayormente para personas que no poseen discapacidad, llevando a discapacitados a una adaptación obligada.

Actualmente existen políticas de inclusión para personas con discapacidades, tales como la Ley Orgánica de Discapacidades, las misiones: Las manuelas y las Joaquinas, la primera se orienta a ayudas técnicas y bonos a personas con discapacidad, mientras que la segunda busca brindar capacitaciones, bono a la persona al cuidado del discapacitado y ayudas técnicas a personas con discapacidad muy grave. (Secretaría Técnica plan Toda una Vida)

El mundo al estar dotado de gran cantidad de información visual, hace difícil la inclusión de personas con discapacidad visual en el desenvolvimiento de actividades cotidianas, tales como movilidad, orientación, comunicación, reconocimiento de objetos, entre otros.

1.1.2.1 Movilidad

La movilidad es una de las facultades que poseen muchos seres vivos, el ser humano utiliza sus sentidos y extremidades para moverse o desplazarse de un lado a otro, sin embargo al tener una discapacidad visual o problemas de visión, la movilidad se vuelve un reto (Terven et al, 2013: p.21), por ello estas personas deben ser entrenadas lo más pronto posible teniendo que preparar otros sentidos como la audición y tacto para sortear obstáculos y llevar una vida autónoma.

1.1.2.2 Orientación

Las personas necesitan moverse utilizando la orientación, pues esta les servirá para saber a dónde dirigirse tomando en cuenta su posición en el entorno, memorizando ciertas señales o lugares para poder acceder al objetivo, ya sea el llegar a cierta dirección o conseguir algún objeto.

El tener alguna discapacidad o problema visual, dificulta poder orientarse, por lo que muchas de las personas no videntes preguntan por información si están en el camino correcto, qué camino seguir para llegar a cierto lugar o conseguir cierto objeto.

1.1.2.3 Comunicación

La información es la herramienta fundamental para conocer y darnos cuentas del mundo que nos rodea, permite analizar situaciones y encontrar conclusiones. La información mundial, nacional, provincial y de diversos temas está presente en diarios, revistas, posters, artículos, pero ¿Cuántos de estos documentos son orientados a personas con discapacidad visual y han estado presentes en puntos de venta donde normalmente cualquier persona puede tener acceso?

Según la información recogida por el artículo de (Terven et al, 2013: p.22), la mayoría de elementos que brindan información escrita no están en braille, sistema de escritura y lectura especialmente orientada a personas con discapacidad visual debido a que utiliza el tacto, otro dato preocupante que se menciona es que solo un 10% de niños no videntes aprenden braille, limitando aún más la absorción de información.

En cuanto a la interacción social existen claras desventajas, pues al entablar conversaciones con una o varias personas, no sólo las expresiones verbales son usadas sino también aquellas que interviene la mímica, con movimientos y gesticulaciones, para poder enfatizar el mensaje a enviar.

Esto le representa un problema a la persona no vidente o con algún tipo de problema de visión, para personas con problemas de visión o con ceguera pues no se puede captar la idea del mensaje en su totalidad.

1.1.2.4 Reconocimiento de objetos

El reconocimiento de objetos es otro de los retos a los cuales deben enfrentarse las personas con discapacidad visual o problemas de visión dada la necesidad de interactuar con el entorno (Terven et al, 2013: p.22), por ejemplo al momento de vestirse, bañarse y limpiar, es decir, al realizar actividades cotidianas se debe tener en mente la forma, sensación, sonido o aroma que tienen cada una de las herramientas a usarse, también en la, artículos de higiene personal o limpieza.

Claramente ciertos objetos son más fáciles de reconocer que otros, por su aroma o textura, sin embargo otros son más complejos de identificar debido a la similitud de característica, un ejemplo claro es el dinero, monedas y billetes

El dinero, medio legal de pago o intercambio para obtener bienes o servicios, es utilizado en todo el mundo, en el Ecuador la moneda oficial es el dólar estadounidense, el cual cuenta con monedas y billetes. Las monedas en circulación cuentan con 6 denominaciones: \$0.01, \$0.05, \$0.10, \$0.25,

\$0.50 y \$1, teniendo cada una de ellas distinto diámetro, por esta razón son más fáciles de diferenciar entre una y otra para personas con discapacidad visual.

En cuanto a los billetes sus denominaciones son: \$1, \$2, \$5, \$10, \$20, \$50 y \$100, todas estas poseen una dimensión de: 155.956 mm x 66.294 mm de ancho, además de colores muy parecidos lo que provoca dificultad de reconocimiento para personas con baja o nula visión.

En otros países como México y Chile, se han implementado técnicas que ayudan a personas con discapacidad visual a identificar la denominación de sus billetes, cambiando en cada una, sus características tales como: colores brillantes, diferencias en los tamaños de cada denominación y marcas táctiles.

Otra opción que ha realizado el Banco de México es que en caso de que la persona no pueda determinar de qué billete se trata es utilizar una plantilla, esta lo que hará es que al doblar el billete, indicará mediante braille la denominación correcta del billete.



Figura 1-1: Tablilla de billetes para personas con discapacidad

Fuente: <https://goo.gl/xUwKZw>

Una forma que permite a una persona identificar determinado billete de dólar, según la Fundación Americana para ciegos, es doblando a cada denominación de distinto modo, así \$1 se mantendrá desplegado, el de \$5 se doblará a lo largo, \$10 a lo ancho, \$20 se dobla dos veces, sin embargo necesitarán ayuda de personas honestas para poder organizar el dinero, quitándole autonomía.

Una alternativa para hacer a los dólares inclusivos, sería rediseñarlos no obstante esto sería realmente costoso, debido a lo cual esta opción ha sido descartada hasta el momento, por el Departamento de Tesoros de Estados Unidos. (Hasanuzzaman, Yang y Tian, 2012, p. 1022).

1.1.3 *Dispositivos de detección de dinero orientado personas con discapacidad visual*

Para disminuir los problemas de reconocimiento de dinero se han desarrollado aplicaciones móviles, sin embargo estas necesitan dispositivos inteligentes para poder ejecutarse, un ejemplo es Money Reader y Recognizer (Terven, Salas y Raducanu, 2013, p.22), dispositivos portátiles que se dediquen únicamente a la detección de dinero para el uso de personas con problemas visuales existen pocos.

En el 2008 en Colombia se creó un dispositivo portátil capaz de reconocer denominaciones y analizar si se trata de un posible billete falso, ha sido construido mediante sensores de luz y utiliza el audio para informar al usuario, según se explica en el artículo, una persona no vidente consultada elogia el uso de la tecnología y señala que no está de más utilizarla en sus actividades comerciales. (Lancheros, 2008).

En Canadá se desarrolló un dispositivo portátil llamado Note Teller 2, el cual permite detectar billetes, en este hay una ranura donde debe situarse el billete para ser identificado teniendo una precisión del 80% debido a que se pueden presentar problemas con billetes desgastados y arrugadas. (Hasanuzzaman, Yang y Tian, 2012, p. 1023).



Figura 2-1: Note Teller 2

Fuente: <https://goo.gl/UgVgZ6>

En el 2012 en Estados Unidos de América se dio a conocer un dispositivo portátil que utilizaba visión artificial para la detección de los billetes utilizando el algoritmo Speeded-Up Robust Features (SURF) o funciones robustas aceleradas con un nivel bastante alto de precisión.

Se segmenta a los billetes según las características más relevantes en cada uno, estas son las que van a ser comparadas en cada billete, además este propone facilidad de uso a personas con

discapacidad visual pues se emplea en un entorno no controlado. (Hasanuzzaman, Yang y Tian, 2012, p. 1023)

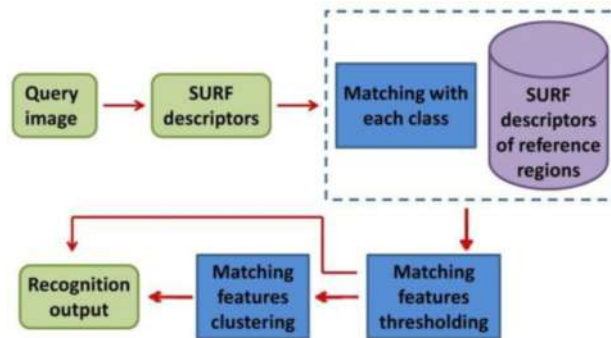


Gráfico 1-1: Flujograma de un sistema de reconocimiento de dinero

Fuente: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3363999/bin/nihms-369665-f0001.jpg>

1.2 Inteligencia artificial

Frente a algún problema o dificultad, el ser humano es capaz de tomar decisiones, basadas en el análisis de los posibles escenarios a los cuales conllevarían dichas acciones, para según ese estudio tomar la mejor alternativa y aplicarla, es decir tomar el mejor camino a seguir. Esto es lo que se conoce como inteligencia.

1.2.1 Definición

La inteligencia artificial basándose en como el ser humano soluciona un problema, busca dotar a máquinas, dispositivos o programas de cierta inteligencia, de tal modo que estos puedan tomar decisiones basándose en la experiencia o entrenamiento, además de poder interactuar con el entorno para la toma de decisiones en distintos procesos.

La inteligencia artificial engloba otras áreas tales como: la visión artificial que trabaja en torno a datos provenientes de imágenes para su análisis y extracción de información, el aprendizaje computacional que usado en imágenes tiene como objetivo la clasificación de distintos grupos de datos y muchos más.

1.2.2 Visión Artificial

Uno de los sentidos más importantes que posee el ser humano y muchos organismos vivos es la visión, que permite captar información del entorno para posteriormente entenderla e interactuar

con este. Muchas de las actividades en la vida cotidiana emplean representaciones gráficas para describir o explicar información, por ejemplo, esquemas de circuitos, señalética o señales de tránsito que son captadas por los ojos del ser humano y entendidos por su cerebro.

La visión artificial toma como modelo aquel comportamiento visual de la anatomía humana e intenta imitar este sentido de forma computacional, para procesar imágenes de forma similar a la del ser humano y de este modo actuar de acuerdo a requerimientos determinados. Esta herramienta simulando el funcionamiento de la vista, permite extraer información de imágenes captadas por cámaras que serían los “ojos”, mediante los cuales una máquina podría ver.

Las imágenes captadas por las cámaras de una escena contienen información objetivo a analizar y fondo, dependiendo de la aplicación, es claro que no se necesitan todos los datos de la imagen, sino los relevantes, es por ello que se debe aplicar una serie de procedimientos para estudiar únicamente la información objetivo. En sí se pueden realizar varias aplicaciones usando este proceso, ya sea el reconocimiento de objetos, interpretación de escenas, o conteo por ejemplo.

1.2.2.1 Sistema de visión artificial

La adquisición de imágenes es la primera de las dos fases que generalmente se constituye un sistema de visión artificial para realizar cualquier tipo de aplicación en este campo.

La adquisición de imágenes tiene como objetivo obtener una imagen digitalizada de la escena que se va a analizar, esta debe ser de buena calidad para poder obtener buenos resultados. Esto dependerá de un sensor o dispositivo, habitualmente se trata de una cámara, además de otras operaciones. (Fases de un sistema de V.A, p. 4)

La iluminación es la condición más relevante dentro de cualquier sistema de visión artificial, pues determinará la calidad de la imagen obtenida. La cámara captará la luz reflejada en cada objeto de la escena tal manera que pueda resaltar aspectos importantes que se van a llevar al análisis. Lo que se debería evitar es la existencia de sombras, bajos contrastes o reflexiones.

Algunos ejemplos de iluminación se muestran en la figura 3-1, estos específicamente aplicados en el campo industrial, que iluminan al objeto a través de: luz directa, luz en ángulo y luz en ángulo bajo.

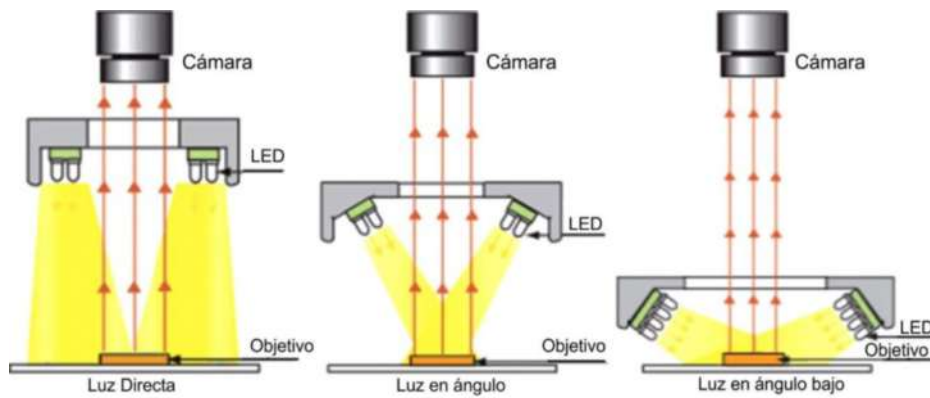


Figura 3-1: Ejemplos de iluminación en el campo industrial

Fuente: http://logicelectronics.net/images/portfolio-details/TECNOLOGIA_tipo_luz.jpg

El sensor de imagen, es un transductor sensible a la luz, lo que quiere decir que la señal eléctrica que envía dependerá de la luz que recibe, permitiendo la adquisición de las características de la escena. La cámara es usada como sensor de imagen, pues trabaja de forma similar a la del ojo humano, para uso en aplicaciones de visión artificial existen diversas con distintas características tales como control de tiempos, señales, sensibilidad, infrarrojas, entre otras.

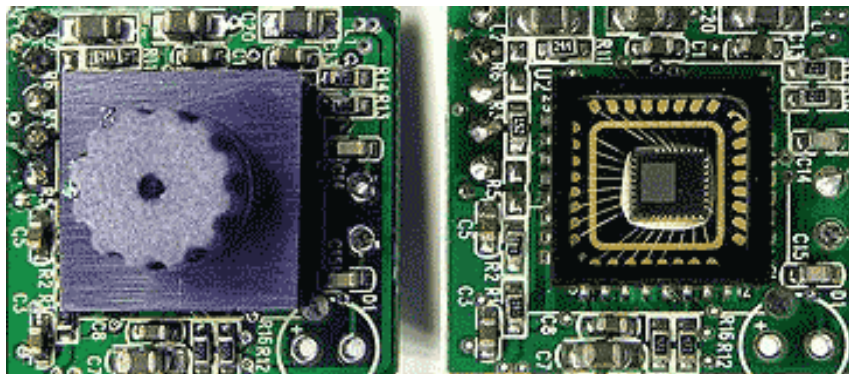


Figura 4-1: Sensor de imagen con y sin lente

Fuente: http://iknowit.ru/image_base/2009/pimg_463_1162.png

Otro aspecto a considerar es la digitalización, ya que las imágenes obtenidas son continuas y se necesita digitalizarlas de tal manera que puedan ser procesadas por el computador. En la figura 5-1 se puede visualizar el proceso de digitalización, teniendo una imagen a procesar, el objetivo, un filtro Red Green Blue (RGB), esta pasa por un sensor CCD (Charge Coupled Device) o CMOS (Complementary Metal Oxide Semiconductor) para obtener la imagen digitalizada

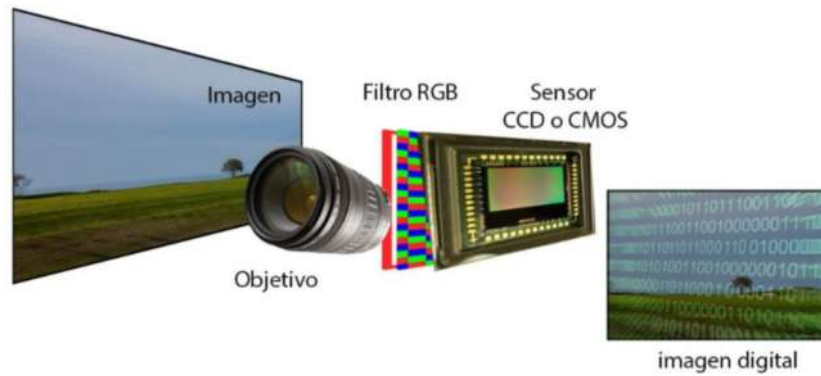


Figura 5-1: Proceso de digitalización de imágenes

Fuente: <http://www.fundacionctic.org/sites/sat/files/images/Captura%20imagen%20digital.jpg>

La segunda fase de un sistema de visión artificial es el análisis de imágenes. Consiste en todas las técnicas y algoritmos utilizados para procesar imágenes que permitirán obtener información importante de la escena. (Fases de un sistema de V.A, pp. 5-7)

El preprocesado o filtrado es utilizado para mejorar la imagen obtenida en la primera fase. Esto permitirá realzar ciertas características, disminuir el ruido, mejorar el contraste, facilitando posteriores etapas de procesado de la imagen. Algunas de las operaciones utilizadas son: Escala de grises, binarización (imagen a blanco y negro), eliminar el ruido causado por cámaras y transformaciones geométricas tales como rotar y trasladar, sin modificar sus valores originales.



Figura 6-1: Imágenes aplicando escala de grises y disminución de ruido

Fuente: https://claudiovz.github.io/scipy-lecture-notes-ES/_images/plot_sharpen_1.png

La segmentación se encarga de dividir o tomar ciertas áreas de la imagen donde se encuentren objetos importantes para el análisis y reconocimiento. Estas secciones se emplearán en el siguiente proceso que es la descripción y son las que llevan toda la información necesaria a analizar.

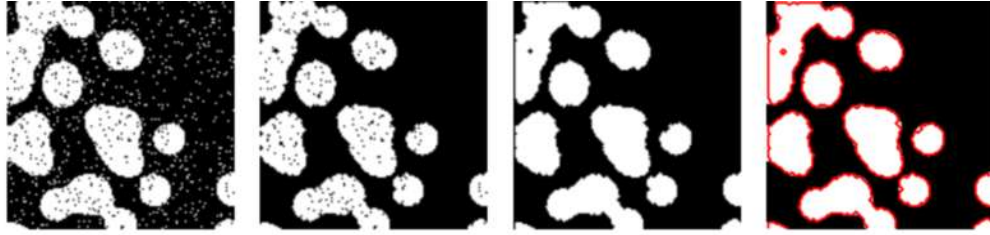


Figura 7-1: Ejemplo de segmentación

Fuente: https://claudiovz.github.io/scipy-lecture-notes-ES/_images/plot_clean_morpho_1.png

La descripción busca determinar las características o información suficiente de cada una de las secciones extraídas en el proceso anterior, con el objetivo de diferenciar cada una de ellas.

El reconocimiento es la última etapa, y se centra básicamente en clasificar a todos los objetos encontrados usando algoritmos de inteligencia artificial entre otros. Se encerrará y etiquetará a cada uno de los objetos.

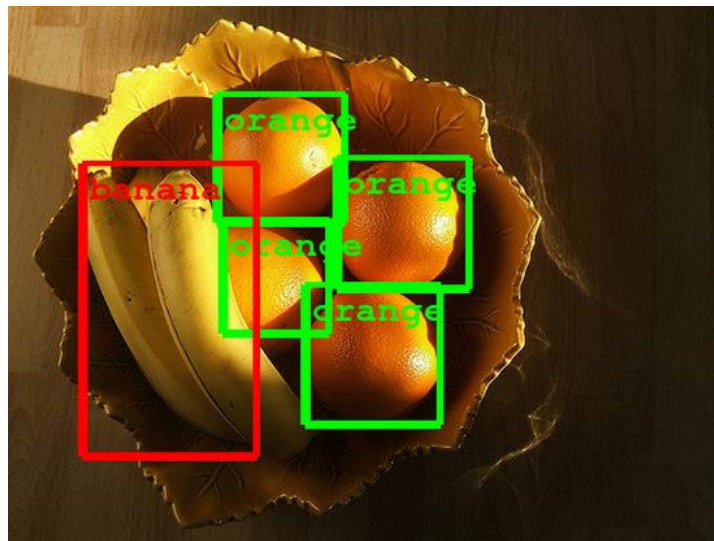


Figura 8-1: Ejemplo de reconocimiento de objetos

Fuente: <https://omicronno.elespanol.com/wp-content/uploads/2014/09/google-reconocimiento-imagen-2.jpg>

1.2.2.2 *Detección de objetos*

En muchas aplicaciones de la visión artificial se busca identificar ciertos objetos, elementos tangibles con características similares entre sí, para posteriormente analizarlos e indicar información sobre ellos. La detección de objetos es la extracción de dichos elementos presentes en una imagen o de una secuencia de imágenes. Esta tarea consta de dos partes, la primera es la extracción de las características de la imagen, y la segunda es búsqueda de los objetos.

Cabe resaltar que las características obtenidas de la extracción se denominan descriptores, para desarrollarlos se necesita de procesos matemáticos, dependiendo del tipo de descriptor que se haya elegido usar, es así que se puede emplear histogramas de color, template matching, patrones binarios locales o Local binary pattern, entre otros. Existen diferentes factores que pueden influir en el desempeño del descriptor tales como la iluminación, la distancia o su orientación y forma.

En cuanto a la búsqueda, se debe realizar una serie de procesos para poder escoger a los posibles candidatos que cumplan con las características del objeto, para ello será necesario emplear una pirámide de imágenes y una ventana deslizante de tamaño canónico que recorra cada una de las imágenes apiramidadas. Una vez obtenidos los posibles candidatos es necesario hacer una clasificación, esto puede realizarse mediante aprendizaje automático.

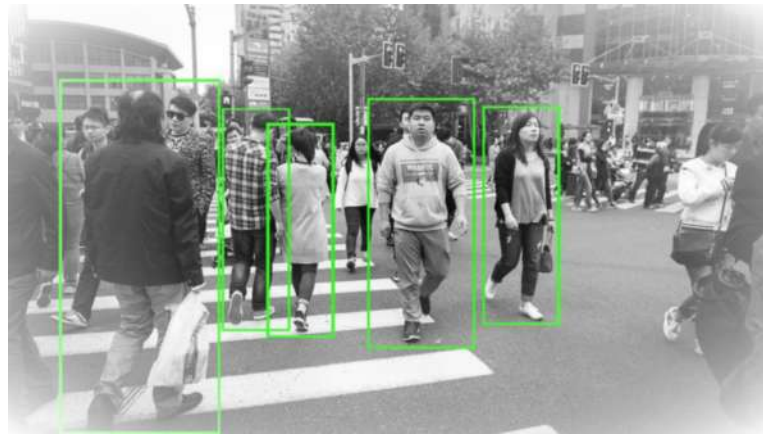


Figura 9-1: Reconocimiento de peatones

Fuente: <https://goo.gl/5yLhPC>

1.2.2.3 *Local Binary Pattern*

A continuación se describen los descriptores Local Binary Pattern (LBP), Uniform Local Binary Pattern ULBP, y una variación de este que es el descriptor invariante a la rotación, este último se realiza con el previo análisis de los dos primeros.

Antes de hablar del patrón binario local uniforme, es necesario entender al patrón binario local o LBP. Este es un descriptor de textura que se caracteriza por su robustez ante cambios de iluminación. El proceso para obtener dicho descriptor es transformar la imagen a escala de grises, luego se toma información de todos los vecinos de cada pixel, comparando el pixel central con los de su alrededor y en base a un umbral se obtiene un número binario.

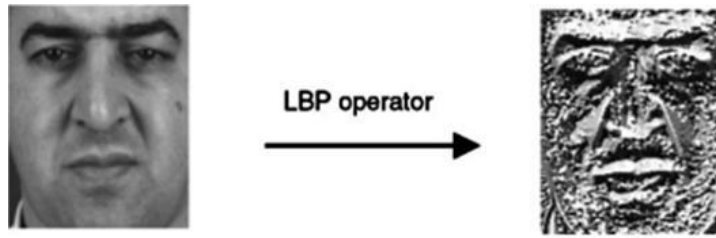


Figura 10-1: Aplicación del descriptor LBP a una imagen de intensidad

Fuente: <https://goo.gl/ywdj4k>

A pesar de que los LBP hayan sido desarrollados para el análisis de texturas, su bajo costo computacional y su poder de discriminación lo han hecho ser un potente descriptor, es más se ha aplicado en reconocimiento de rostros, peatones, entre otros. Para iniciar con el proceso de obtención de características se necesita tener una imagen a color, la cual se transforma a escala de grises. (Descripción de textura en imágenes utilizando Local Binary Pattern (LBP), p. 118)

A continuación se describirá el proceso de obtención del descriptor:

Como primer paso para determinar los valores, se debe centrar en cada pixel de la imagen y realizar una comparación de intensidad de grises con sus vecinos. Además se determinará cierto umbral para la obtención del número binario.

La determinación de umbral se representa mediante esta ecuación:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p, \quad s(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases} \quad \text{Ecuación 1.1}$$

Donde:

P = número de vecinos a comparar.

R = tamaño del vecindario.

g_c = valor del pixel central.

g_p = pixel vecino. p es el número de cada vecino.

Un ejemplo de extracción se visualiza en la figura 11-1. En donde se muestra el proceso de obtención del descriptor LBP dada una imagen, es importante recalcar que se toman en cuenta la intensidad de los pixeles vecinos dado un píxel central ya que estos dos valores van a ser comparados. El valor obtenido luego del proceso es asignado al píxel central, y se continúa con la realización del mismo procedimiento para el resto de los píxeles de la imagen.

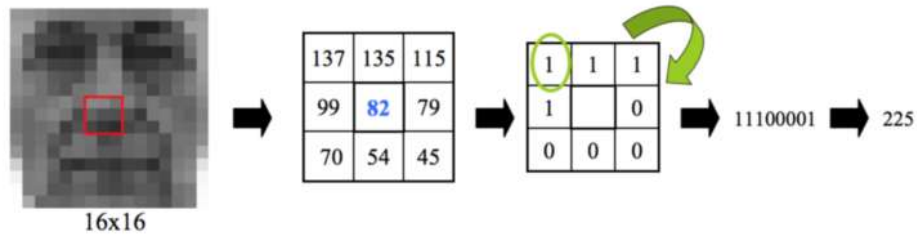


Figura 11-1: Proceso de obtención de los valores de Local Binary Pattern

Fuente: <https://goo.gl/DrjiTq>

Se toma de la imagen cada pixel, en este caso el elemento a analizar será el 82, ahora se comparará si es menor o mayor con relación a los otros pixeles, si el valor central es mayor se asignará el 1, si es menor será 0. Una vez obtenido el número binario se debe elevar el número de la celda con base 2, y se obtiene el nuevo valor del descriptor. Es importante aclarar que las posiciones de las celdas, en este caso de 0 a 8, se recorren arbitrariamente.

Los patrones binarios locales uniformes son una variación de LBP, a pesar de tener similar desarrollo, la diferencia está en que sólo toma ciertos valores para su estructura, mientras a los restantes les asigna un valor diferente, reduciendo así el vector a obtener como descriptor de 256 elementos a 59, obteniendo invariancia rotación y robustez ante cambios de iluminación.. (Local Binary Patterns applied to Face Detection and Recognition,2010,pp. 7-11)

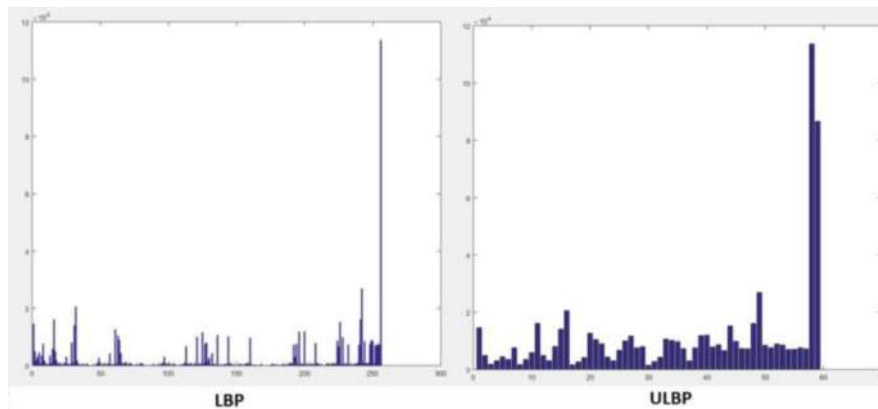


Gráfico 2-1: Histograma de una imagen aplicando LBP y ULBP

Fuente: <https://cesartroyasherdek.files.wordpress.com/2016/02/lbph.jpg?w=840>

Un patrón LBP es uniforme, siempre y cuando en su estructura circular binaria sólo haya de 0 a 2 transiciones (cambios de 0 a 1, o de 1 a 0). Se pueden obtener 58 patrones uniformes diferentes, si los valores no coinciden con estos patrones, entonces se le asignará un valor diferente, obteniendo en su totalidad 59 elementos. En la figura 12-1, se puede apreciar todos los patrones binarios locales uniformes, los círculos oscuros representan un 1, mientras que los blancos un 0.

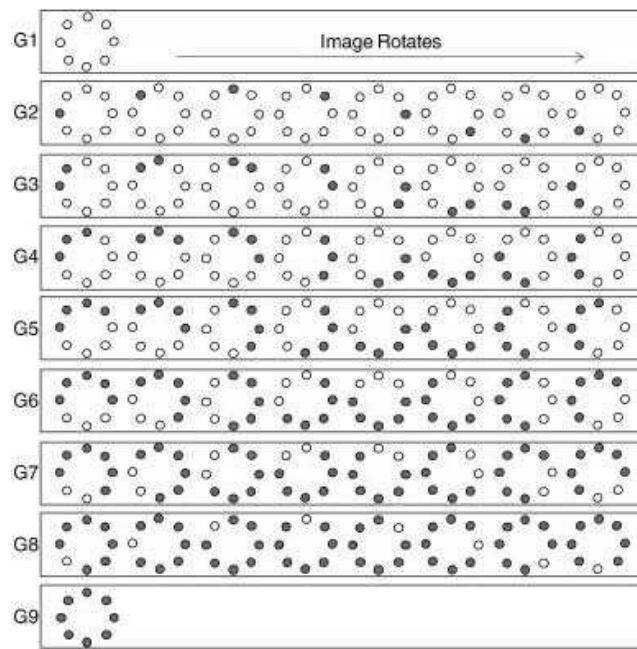


Figura 12-1: Patrones binarios locales Uniformes

Fuente: <https://ars.els-cdn.com/content/image/1-s2.0-S0262885615000992-gr2.jpg>

En la figura 13-1 se pueden observar algunos de los ejemplos correspondientes a la uniformidad y no uniformidad de los patrones según los LBP.

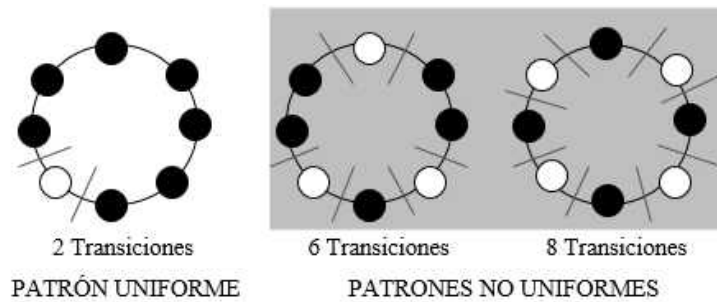


Figura 13-1: Ejemplos de patrones LBP uniforme y no uniforme

Fuente: <https://goo.gl/DrjiTq>

Existe además un método que permite que el descriptor mejore su respuesta ante la rotación. Una vez que se obtiene el número binario necesario para calcular el LBP o LBPU, se procederá a rotar el mismo, hasta obtener el valor mínimo de dichas rotaciones, este es el que será asignado como valor del pixel. (Descripción de textura en imágenes utilizando Local Binary Pattern (LBP), pp. 122-123)

1.2.2.4 *Template Matching*

El template matching, comparación de plantillas o correspondencia de patrones es un procedimiento que consiste en buscar la ubicación de un objeto o plantilla en una imagen más grande. En sí se compara el patrón con toda la imagen, la comparación se realiza por número de píxeles coincidentes y no coincidentes que darán la información de un buen o mal emparejamiento.

Para poder llevar a cabo esta técnica es necesario conocer los métodos que se pueden emplear tales como medidas de disimilitud y medidas de similitud. El primer método se basa en la disimilitud, es decir la plantilla pasa por la imagen y se resta del valor de la ventana, dicho valor se asignará en el valor central de la imagen resultante. De este modo se obtendrá un mapa de mínimos en donde la sección más oscura determinará el mejor emparejamiento.

Los métodos que emplea OpenCV para poder determinar las disimilitudes entre plantillas y la imagen son: `cv.TM_SQDIFF` y `cv.TM_SQDIFF_NORMED`.

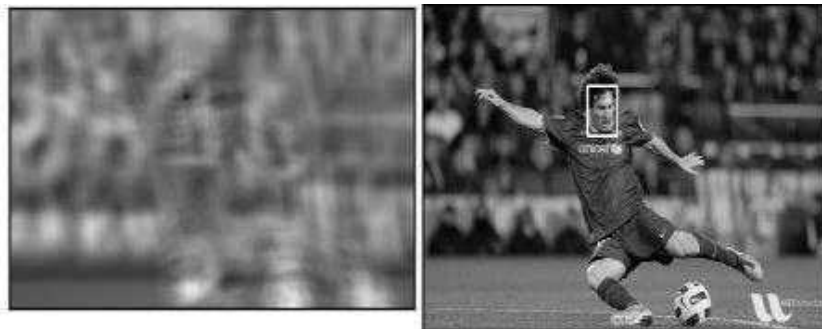


Figura 14-1: Ejemplo de template matching con medidas de disimilitud

Fuente: https://docs.opencv.org/trunk/template_sqdiff_5.jpg

Otro método que emplea el template matching es el cálculo de similitud mediante la correlación, para ello se multiplican el número de coincidencias por los valores la ventana y de la plantilla, este valor se asigna al pixel central de cada ventana. El mejor emparejamiento corresponde a la sección más clara de la imagen resultante.

Las funciones que emplea OpenCV para poder determinar las similitudes entre plantillas y la imagen son: `cv.TM_COEFF`, `cv.TM_CCOEFF_NORMED`, `cv.TM_CCORR` y `cv.TM_CORR_NORMED`. A continuación se puede apreciar un ejemplo del cálculo de similitud:

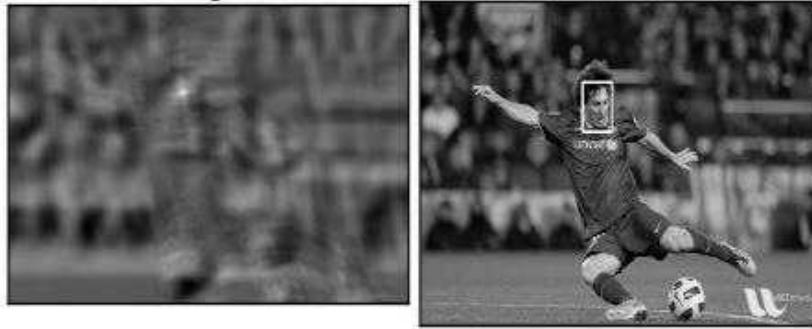


Figura 15-1: Ejemplo de template matching con medidas de similitud

Fuente: https://docs.opencv.org/trunk/template_ccoeff_1.jpg

En librerías de procesamiento de imágenes como OpenCV, se usan varios métodos en template matching derivados del cálculo de disimilitud y de similitud. Estos se deben aplicar dependiendo de las circunstancias de análisis de la imagen, así como la forma del objeto.

Hay que tomar en cuenta también que esta técnica busca el patrón tal y como se plantea, es decir que no se obtendrá buenos resultados ante rotación, cambio de tamaño, oclusiones, cambio de iluminación entre otros, de los patrones que se encuentren en la imagen global, para solucionar este problema se pueden aplicar otros procedimientos a la imagen o usar una variedad de imágenes del mismo patrón.

1.2.3 *Aprendizaje automático*

También llamado machine learning, es una rama de la inteligencia artificial que tiene como objetivo darle al computador la capacidad de aprender, dada información basada en antecedentes, brindándole así la facultad de decisión.

1.2.3.1 *Aprendizaje supervisado y no supervisado*

Básicamente existen dos formas por las cuales un computador podrá aprender, el aprendizaje supervisado y el no supervisado, pudiendo haber ciertas variantes también, pero se tomará en cuenta en particular a estas a continuación:

El aprendizaje supervisado se da cuando se conoce a que clase o agrupación pertenece cada elemento del conjunto de entrenamiento, esto quiere decir que para entrenar al clasificador de dicho conjunto se separará a cada uno de ellos dependiendo de la clase a la cual pertenezcan. Se pueden aplicar clasificadores paramétricos y no paramétricos de acuerdo a la separabilidad entre cada clase.

Un clasificador se basa en superficies o regiones de decisión para separar las clases, cuando estas son separables se emplean clasificadores paramétricos basados en las distribuciones de probabilidad de cada clase. Entre los clasificadores paramétricos se encuentran la regresión logística, Support Vector Machine (Máquinas de soporte), entre otros.

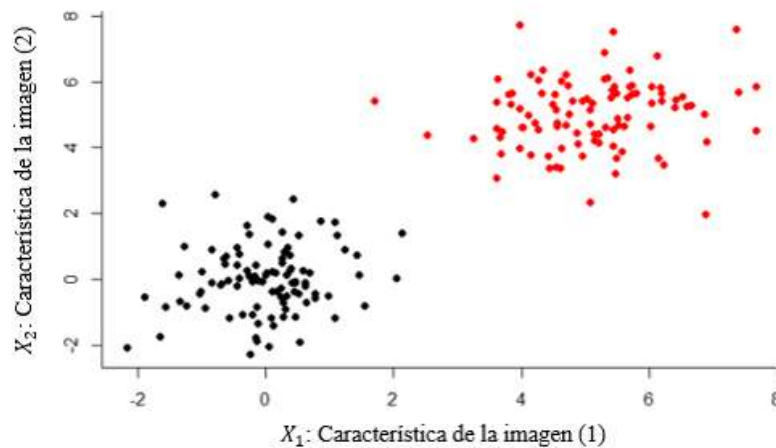


Figura 16-1: Ejemplo de dos clases separables

Fuente: <https://caminosaleatorios.files.wordpress.com/2017/11/ejemplo.png?w=640>

Si en cambio las clases no son separables linealmente se emplean otros clasificadores llamados no paramétricos tales como Adaboost.

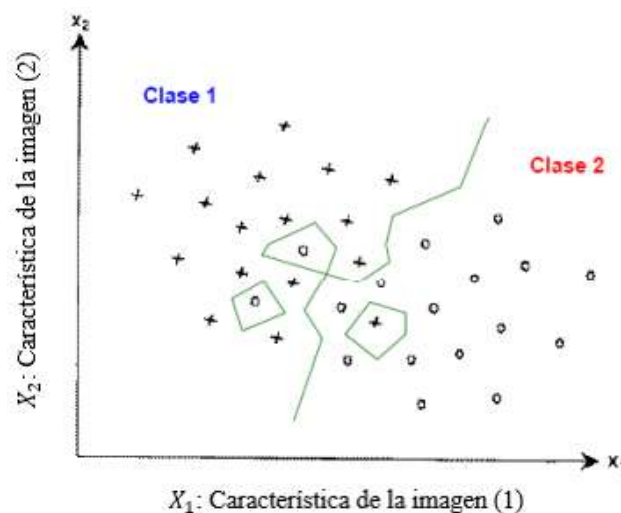


Figura 17-1: Ejemplo de clases no separables linealmente

Fuente: http://isa.umh.es/asignaturas/iarp/transparencias/teoria_RP.pdf

El aprendizaje no supervisado en cambio tiene como conjunto de entrenamiento patrones de los cuales no se conoce a que clase pertenecen, teniendo como objetivo la agrupación de dichos elementos en clases.

1.2.3.2 Regresión Logística

Esta es una herramienta estadística que permite clasificar patrones o muestras dado un conjunto de entrenamiento dividido en dos clases. De estos datos se podrá determinar una recta que dividirá a ambas clases para posteriormente clasificar nuevos datos ingresados entre una u otra alternativa. La regresión logística usa la función sigmoidea o logística para el proceso de aprendizaje y como umbral para la detección del objeto, de allí su nombre.

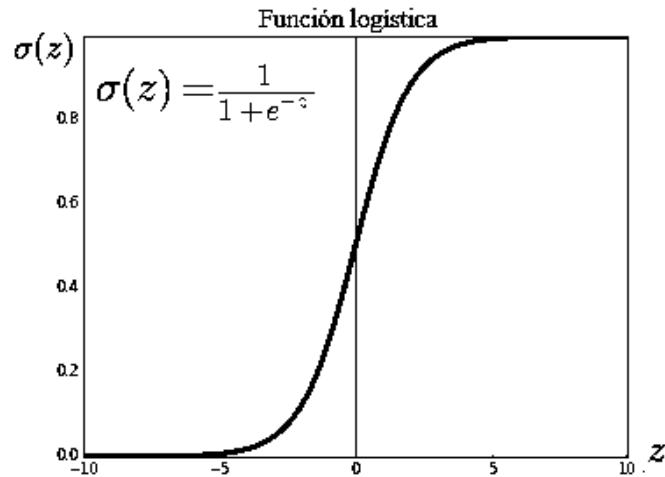


Gráfico 3-1: Función logística o sigmoide

Fuente: <https://ml4a.github.io/images/figures/sigmoid.png>

Se usa esta función debido a que existe un rango de flexibilidad al escoger entre dos grupos cuyos valores pertenecerán a 0 o a 1, para determinar un umbral. Esto no se podría conseguir con una función escalón, ya que sólo se tiene un 0 o 1 y no permite introducir un umbral o valores entre estos límites.

La regresión logística es muy empleada en procesos de aprendizaje automático supervisado debido a su rapidez y buenos resultados al clasificar. En cuanto a la clasificación de objetos, en visión artificial, el proceso para conseguir este clasificador empieza con el descriptor, este se usará para obtener las características de dos grupos de datos. De cada muestra de dicho grupo se deberá obtener un modelo usando descenso de gradiente.

Una vez que se tiene cada modelo, se debe obtener mediante la función de costo, el menor valor del error, del grupo de datos correspondiente a la clasificación. Este le corresponderá al mejor modelo obtenido de todas las muestras y se usará para poder determinar el umbral de clasificación. A continuación se explica el proceso matemático:

Cabe resaltar que la longitud del modelo será la del descriptor más uno, y al descriptor se le antepone un uno para todo el proceso, y su forma compacta se expresa:

$$w^t x = 0 \quad \text{Ecuación 2.1}$$

$$\text{Modelo: } w = (w_0, w_1, \dots, w_n)^t \quad \text{Ecuación 3.1}$$

$$\text{Descriptor: } x = (1, x_1, \dots, x_n)^t \quad \text{Ecuación 4.1}$$

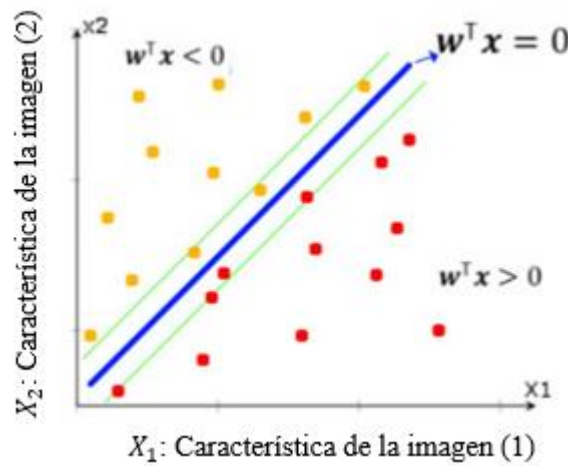


Figura 18-1: Clasificación de dos clases con regresión logística

Fuente: https://cesartroyasherdek.files.wordpress.com/2016/02/screenshot_3.jpg?w=395&h=293

En la figura 18-1 $w^t x$, representa el umbral, Para poder obtener cada modelo (w), se necesita un grupo de entrenamiento y etiquetas, estas serán 0 o 1 dependiendo del grupo al que pertenezcan.

$$S = \{(x^1, y^1), \dots, (x^M, y^M)\} \quad \text{Ecuación 5.1}$$

S es el conjunto de entrenamiento, que está compuesto por: descriptores, representados por la letra x , y las etiquetas por la letra y , esta última será 0 o 1, dependiendo de la clase a la que pertenezca la muestra. La totalidad de las muestras corresponde a M . Posteriormente w , será representado como el modelo.

Es necesario que la función de costo que se vaya a utilizar sea convexa, esto quiere decir que tenga un único punto mínimo para de esta manera buscar el menor error con cada modelo. La función de costo a usar es la siguiente:

$$J(w) = -\frac{1}{M} \sum_{j=1}^M y^j \ln(h_w(x^j)) + (1 - y^j) \ln(1 - h_w(x^j)) \quad \text{Ecuación 6.1}$$

El término $h_w(x^j)$, corresponde a la función sigmoidea del modelo por el descriptor, dicho de otra manera:

$$h_w(x^j) = \frac{1}{1 + e^{-w^T x}} \quad \text{Ecuación 7.1}$$

El aprendizaje de cada modelo (w) se llevará a cabo tomando el valor anterior del modelo y la derivada de la función de costo. Además se puede añadir un nuevo término, α , para acelerar el proceso de aprendizaje. Entonces,

$$w_n^{(k)} = w_n^{(k-1)} - \alpha \frac{\partial}{\partial w_n} J(w^{k-1}) \quad \text{Ecuación 8.1}$$

$$\frac{\partial}{\partial w_i} J(w) = \frac{1}{M} \sum_{j=1}^M (h_w(x^j) - y^j) x_i^j \quad \text{Ecuación 9.1}$$

$$w_n^{(k)} = w_n^{(k-1)} - \alpha \frac{1}{M} \sum_{j=1}^M (h_w(x^j) - y^j) x_n^j \quad \text{Ecuación 10.1}$$

Hasta este punto ya se tiene el mejor modelo del grupo de entrenamiento, ahora, en vez de la forma compacta: $w^t x = 0$, se debe determinar el umbral (T) en vez de 0 que determinará la clasificación, para ello:

$$\text{Clasificador} = C_{w,T} = \begin{cases} 1 & \text{si } h_w(x) < T \\ 0 & \text{en otro caso} \end{cases} \quad \text{Ecuación 11.1}$$

Este clasificador posee una estructura de una neurona artificial, como se verá a continuación:

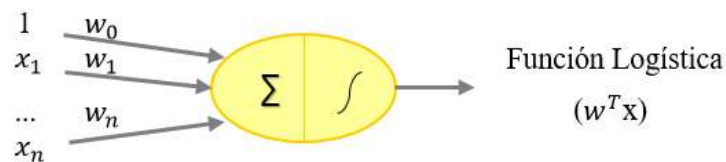


Figura 19-1: Regresión logística como neurona artificial

Fuente: <https://www.coursera.org/learn/deteccion-objetos/lecture/bojHF/12-3-a-regresion-logistica-clasificacion-i>

En la representación de la clasificación con regresión logística como neurona artificial, las entradas serán cada elemento del descriptor, que son multiplicados por pesos en este caso cada elemento del modelo, posteriormente cada valor va a ser sumado y aplicado la función logística de donde se obtiene un solo valor o umbral a comparar.

1.3 Hardware

A continuación se describirán los principales elementos o herramientas físicas utilizadas para el desarrollo de este dispositivo portátil, tales como: la placa raspberry pi 3, cámara raspberry v2, altavoz y una batería de litio-ion recargable.

1.3.1 Raspberry pi 3

Esta es una pequeña placa que contiene un computador en su interior, pues se pueden conectar otros dispositivos propios de una computadora tales como pantalla, teclado, mouse, cámara entre otros. Dado a su pequeño tamaño es muy fácil de transportar y usar además de ser muy poderosa para desarrollar distintos proyectos.

Su versión más reciente es la placa Raspberry Pi 3 Modelo B, diferenciándose con sus antecesoras por tener Wi-Fi y Bluetooth, brindándole aún más comodidad al momento de programarla y ya en aplicaciones finalizadas pues elimina la necesidad de un cable de red. Puede alimentarse también con baterías sin tener la necesidad de estar conectada a la red eléctrica. En la figura 20-1 se puede apreciar la placa y las partes que la componen.

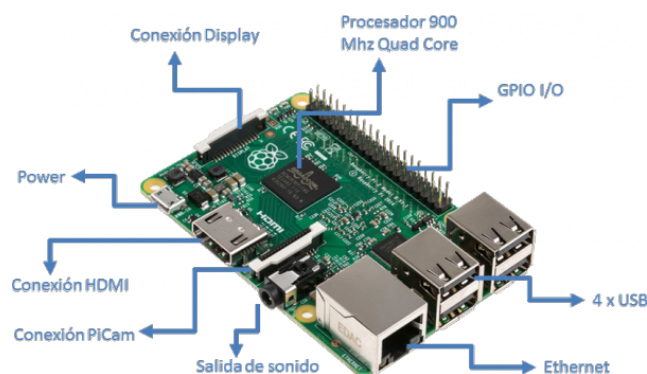


Figura 20-1: Raspberry pi 3 Modelo B

Fuente: <https://bit.ly/2MuszD7>

1.3.1.1 Características

Las características que posee la placa Raspberry pi 3 Modelo B son las siguientes:

Tabla 1-1: Especificaciones técnicas de la placa Raspberry pi 3 Modelo B

Características	Descripción
Procesador	Broadcom BCM2837 64Bit Quad Core @ 1.2GHz
GPU	Videocore IV
RAM	1GB SDRAM @400MHz
Almacenamiento	MicroSD
USB 2.0	4 Puertos USB
Consumo máximo de voltaje	2.5A – 5V
GPIO	40 pines
PuertoEthernet	10/100 Ethernet (RJ-45)
Wi-Fi	Integrado. 803.11n
Bluetooth BLE	Integrado. Bluetooth 4.1

Realizado por: Gabriela Solano. 2018

Fuente: Raspberry pi.2016.

Un factor a tomar en cuenta es la clase que posea la memoria SD, que hace las veces de disco duro en la raspberry pi, pues al ser mayor su clase, más será la velocidad de lectura y grabación.

1.3.2 Cámara Raspberry Pi

La placas Raspberry pi han destinado en su arquitectura una pequeña sección dedicada a una cámara propia de su marca, haciendo en su conjunto una herramienta compacta para el uso de proyectos en los que se requiera un sistema de visión, a diferencia del espacio que se utilizaría con una webcam. En la figura 21-1 se aprecia el pequeño circuito que compone la cámara de Raspberry pi.

Existen dos versiones de estas cámaras, la primera de 5 megapíxeles, mientras que la más reciente lanzada en 2016 cuenta con 8 megapíxeles. Estas se conectan a través de un bus de datos a la placa brindándole comodidad en su uso y poseen las mismas medidas. Además los módulos necesarios para usar estas cámaras ya vienen cargados por defecto en la raspberry pi, con el lenguaje de programación Python.

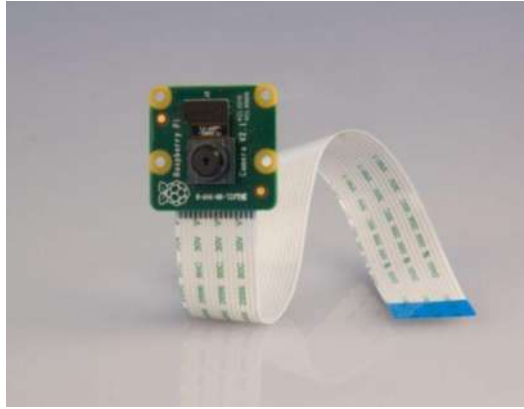


Figura 21-1: Cámara Raspberry pi v2

Fuente: https://www.raspberrypi.org/app/uploads/2016/04/camera_v2-500x386.jpg

1.3.2.1 Características

Las características que a continuación se describen, son aquellas de la cámara v2 de 8 megapíxeles.

Tabla 2-1: Especificaciones técnicas más relevantes del módulo de cámara Raspberry v2

Características	Descripción
Dimensiones	Alrededor de 25x24x9 mm
Peso	3g
Resolución	8 Megapíxeles
Sensor	Sony IMX219
Resolución del sensor	3280 x2464 píxeles
Video Modes	1080p30, 720p60 y 640x480p60/90

Realizado por: Gabriela Solano. 2018

Fuente: Raspberry pi.2016.

1.3.3 Altavoz

Un altavoz es un dispositivo capaz de convertir energía eléctrica en energía acústica para la reproducción de sonido. Para que cumpla su objetivo, se necesitan de dos etapas, la primera se da al momento en que la señal eléctrica produce movimiento en el diafragma del parlante, produciendo a su vez sonido.

La potencia de cada altavoz es la potencia eléctrica máxima permitida por el parlante antes que este pueda averiarse o se pierda calidad de audio. A continuación en la figura 22-1 se puede apreciar un pequeño parlante frontal y posteriormente, este cuenta con una potencia de 0.25W.



Figura 22-1: Altavoz de 0.25W

Fuente: https://images-na.ssl-images-amazon.com/images/I/61nJ6onM-mL._SX425_.jpg

1.3.4 *Batería externa*

Una batería externa o también llamada power bank, es aquella que permite cargar un dispositivo en cualquier lugar sin necesidad de conexiones a la red eléctrica. Consta de una batería de litio ion recargable y un circuito electrónico, este último cuenta con un conector USB que entrega energía a determinado dispositivo y un conector micro USB, mediante el cual se carga la batería.

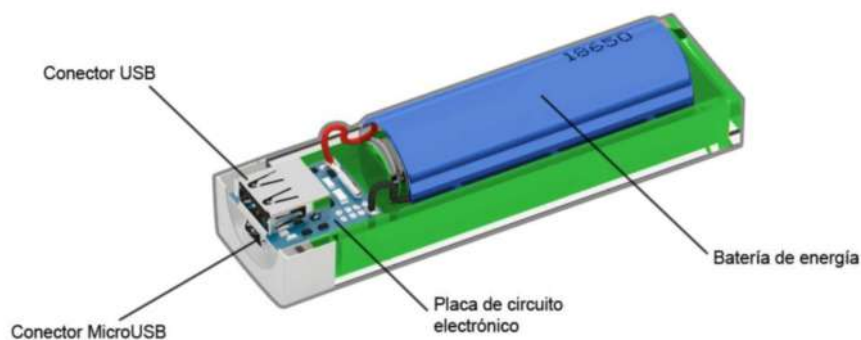


Figura 23-1: Ejemplo de batería externa y sus partes

Fuente: <http://www.powerbankevacolor.com/public/modules/page/images/Partes%20de%20Powerbank4.jpg>

La velocidad de carga que entrega la batería externa es determinada por el fabricante de acuerdo a los componentes y materiales que consta su estructura, es así que hay aquellas de: 2000mAh, 3000mAh, 3500mAh, entre otras. Estas mantienen 5V de salida, así como 1Ah por cada hora nominalmente. Además estas baterías externas son muy utilizadas en el mercado, debido a su fácil manejo y pequeña estructura

1.3.4.1 Características de una batería externa de 3000mAh

Tabla 3-1: Especificaciones técnicas de una batería externa de 3000mAh de 2 celdas

Características	Descripción
Número de celdas	2
Fabricación de la batería	Litio-Ion
Entrada	Micro USB 5V/1 A
Salida	USB 5V/1 A
Capacidad	3000mAh
Dimensión	44mm de alto x95mm de largo x22mm de alto

Realizado por: Gabriela Solano. 2018

Fuente: <https://www.amazon.com/3000mAh-External-Battery-Pack-Power/dp/B00NVN2Z1K>

1.4 Software

1.4.1 Sistema Operativo Raspbian

El sistema operativo Raspbian, especialmente desarrollado para el manejo de una raspberry pi es el predilecto a la hora de manejar y programar en la placa. Posee un entorno amigable con el usuario que facilita su interacción, además de ser un software libre. La placa también soporta otros sistemas operativos, sin embargo este es el usado por defecto.

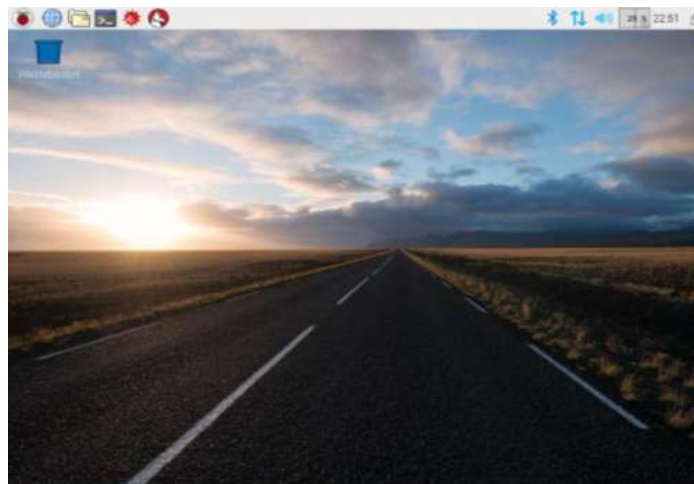


Figura 24-1: Escritorio del sistema operativo Raspbian

Fuente: <https://cdn.makeuseof.com/wp-content/uploads/2016/10/muo-linux-raspbian-pixel-desktop.png>

1.4.2 *Python*

Python es un lenguaje de programación gratuito, multiplataforma orientado a objetos, al cual se pueden integrar gran cantidad de librerías para el desarrollo de múltiples aplicaciones y usado por defecto en la Raspberry pi. Entre sus distribuciones, la 2.7 es la más usada sin embargo no es la más reciente.



Figura 25-1: Logo del lenguaje de programación Python

Fuente: <https://goo.gl/m7Qyng>

Python permite importar varios módulos que permiten desarrollar distintos proyectos y aplicaciones. Por ejemplo numpy, una biblioteca que permite realizar funciones matemáticas en vectores y matrices, os, biblioteca que permite manipular archivos o carpetas del sistema, pickle, biblioteca que transforma un objeto a una cadena, y una cadena a objeto permitiendo manejar gran cantidad de datos de manera más rápida.

En el presente trabajo se ha utilizado Python en su versión 2-7 para el desarrollo de los programas de visión artificial y procesamiento inteligente.

1.4.3 *OpenCV*

OpenCV (Open Source Computer Vision), es una librería de código abierto desarrollada por Intel, está destinada a la visión artificial, además es multiplataforma y usada en diferentes lenguajes de programación, lo que la hace muy popular. Entre sus algoritmos se destacan la implementación de visión artificial y aprendizaje computacional permitiendo realizar diversos proyectos y aplicaciones.

Los años de estudio y optimización de sus algoritmos desarrollados en C, lo han convertido en una herramienta muy útil y eficiente incluso en aplicaciones en tiempo real, además de ser compatible con varios sistemas operativos, tales como Windows, Linux o MAC OS X.



Figura 26-1: Logo OpenCV

Fuente: <https://goo.gl/CHcbRU>

1.4.4 *SolidWorks*

Este es un software para realizar diseños, simulaciones en 2D y 3D, se basa en diseño asistido por computadora (CAD). Permite modelar cualquier tipo de objetos de forma sencilla y rápida, dando una vista previa digital de cómo se obtendrá el resultado final del objeto.



Figura 27-1: Logo SolidWorks

Fuente: <https://i.servimg.com/u/f11/18/18/55/83/solidw10.jpg>

CAPÍTULO II

2. MARCO METODOLÓGICO

En el presente capítulo se explica cómo ha sido creado el sistema de visión e inteligencia artificial usando aprendizaje automático, para el desarrollo de la identificación del billete de una imagen y posterior clasificación de cada denominación, así como el diseño y construcción del dispositivo. Esto se llevó a cabo usando los siguientes métodos: deductivo, heurístico y experimental.

El método deductivo busca llegar a conclusiones particulares a partir de generalidades. De este modo se ha buscado información acerca de diversos métodos y herramientas para desarrollar el sistema de visión artificial y de procesamiento inteligente. De entre todas las opciones se escogió aquellas que poseían mejor rendimiento y que llevan también a una respuesta óptima.

En cuanto al método heurístico se identifica y presenta el problema de la clasificación de billetes y diferenciación de denominaciones, de ambos se plantean y ejecutan distintos procedimientos para posteriormente evaluarlos y poder tomar las mejores estrategias para cumplir con la solución del problema.

El método experimental está presente en diversas etapas del proyecto, principalmente en el desarrollo del procesamiento inteligente y en pruebas de funcionamiento global del dispositivo. Para ello se manipula distintas variables o datos para observar el comportamiento que tenían en el software y hardware.

2.1 Hardware

A continuación se describen todos los componentes, circuitos y procesos realizados para el diseño y construcción del dispositivo detector de billetes.

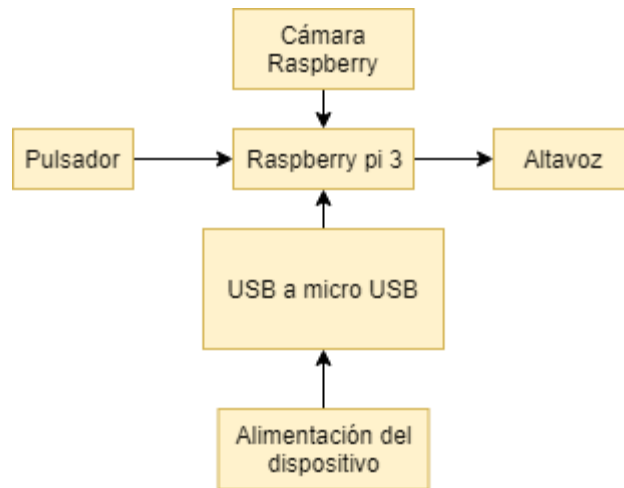


Gráfico 1-2: Diagrama de bloques del dispositivo

Realizado por: Solano, G. 2018

2.1.1 Componentes

La raspberry pi 3 es la placa base del dispositivo pues en esta se desarrolla todo el software para la detección de billetes con la ayuda de la cámara raspberry v2 para la toma de captura de imágenes. Con estos dos componentes ya se puede realizar las detecciones pero es importante darlos a conocer, entonces se empleó un parlante de 0.25W para que emita todos los comandos de voz.

Ya que este es un dispositivo portátil es necesaria una batería por lo que se elige una de 3000mAh debido al poco espacio que tiene su estructura y a que se puede usar el dispositivo por aproximadamente 3 horas. El switch es utilizado para encender el dispositivo, es decir para permitir el paso de energía desde la batería hasta la raspberry pi 3 mientras que el pulsador únicamente captura la imagen a ser analizada.

En sí el hardware usado para la construcción del dispositivo cuenta principalmente con los siguientes componentes:

- Raspberry pi 3
- Cámara raspberry pi v2
- Bus de datos para la cámara
- Altavoz de 0.25W
- Circuito para el altavoz
- Batería de 3000mAh
- Conectores para batería y la raspberry pi 3

- Switch
- Pulsador
- Conector USB
- Conector micro USB
- Cable

Todos estos componentes deben ser conectados a la placa, en su mayoría ya se encuentran desarrollados a excepción del circuito para el altavoz y pulsador, estos son explicados a continuación:

2.1.2 Circuitos electrónicos

En este apartado se describen todos los circuitos que intervienen en la estructura electrónica del dispositivo detector de billetes.

2.1.2.1 Alimentación del dispositivo

Para la elección de la fuente de alimentación adecuada de este dispositivo se tomó en cuenta la alimentación de la placa raspberry que requiere de un voltaje constante de 5V para su correcto funcionamiento. Se escogió una power bank de 3000mAh debido a su estructura pequeña en comparación a otras con mayor mili amperios hora.

Su estructura cuenta con un conector USB hembra. Este debe ser conectado a un cable USB-micro USB acondicionado con un switch para permitir o no el paso de energía, por último el micro USB (macho) debe ser conectado al micro USB (hembra) correspondiente a la alimentación de la placa raspberry pi 3.

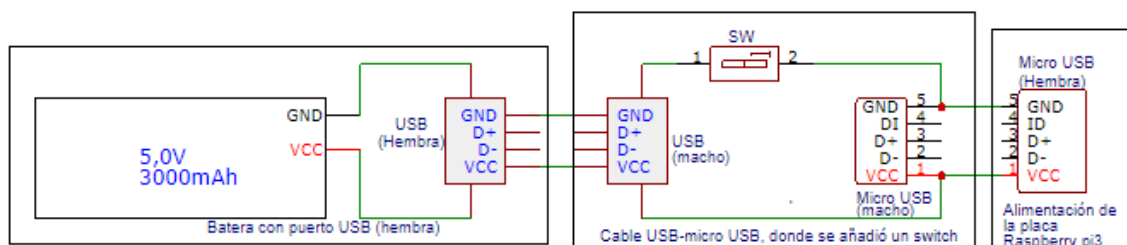


Figura 1-2: Diagrama esquemático de la alimentación del dispositivo

Realizado por: Solano, G. 2018

Cabe destacar que las conexiones entre los conectores han sido únicamente conectadas en VCC y GND, esto debido a que sólo se necesita entregar a la raspberry pi la alimentación y no datos.

2.1.2.2 Circuito pulsador

Para que el usuario pueda capturar la imagen del billete a analizar fue necesario implementar un circuito en el que intervenga un pulsador, como se muestra en la figura 2-2, este se planteó con una alimentación de 3.3V, además el estado de este pulsador es percibido por el pin GPIO 24.

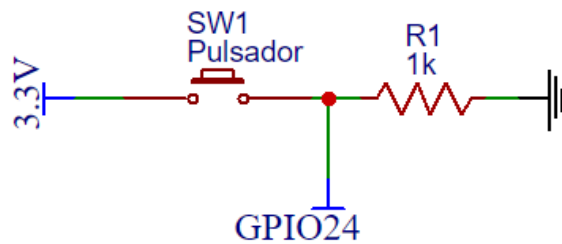


Figura 2-2: Circuito pulsador

Realizado por: Solano, G. 2018

2.1.2.3 Circuito altavoz

Para informar al usuario acerca de la denominación del billete analizado es importante tener un emisor de audio por ello se realiza un pequeño circuito capaz de para amplificar el sonido desde el puerto de audio de la raspberry, como se muestra a continuación:

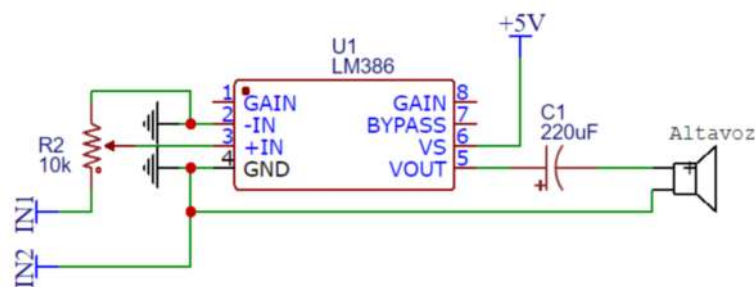


Figura 3-2: Circuito amplificador de audio

Realizado por: Solano, G. 2018

Este circuito de la figura 3-2 cuenta con un amplificador de audio LM386, a este se lo puede alimentar de 4V a 12V, pero se eligió alimentarlo con 5V que da la raspberry pi en su puerto GPIO. Este circuito también necesita de una entrada de audio, IN1 e IN2, que son dadas igualmente por la placa a través del puerto de audio.

Como salida de audio ha empleado un altavoz de 0.25W, debido a su estructura pequeña, esto permitire que el dispositivo final sea compacto a diferencia de si se utilizara uno con mayor vatiaje. Hay que tomar en cuenta que el volumen del audio emitido no es manipulable por el usuario, está pre establecido.

2.1.2.4 Esquema de conexiones al puerto GPIO de la raspberry pi 3

Se integraron los circuitos del pulsador así como del altavoz en uno sólo, pues estos son conectados a los puertos GPIO de la placa raspberry pi 3. Es así que para el altavoz se usan solamente los pines GPIO 4 y 6 para la alimentación, mientras que para el pulsador se usan los pines GPIO 14, 17 y 18 así como se visualiza a continuación:

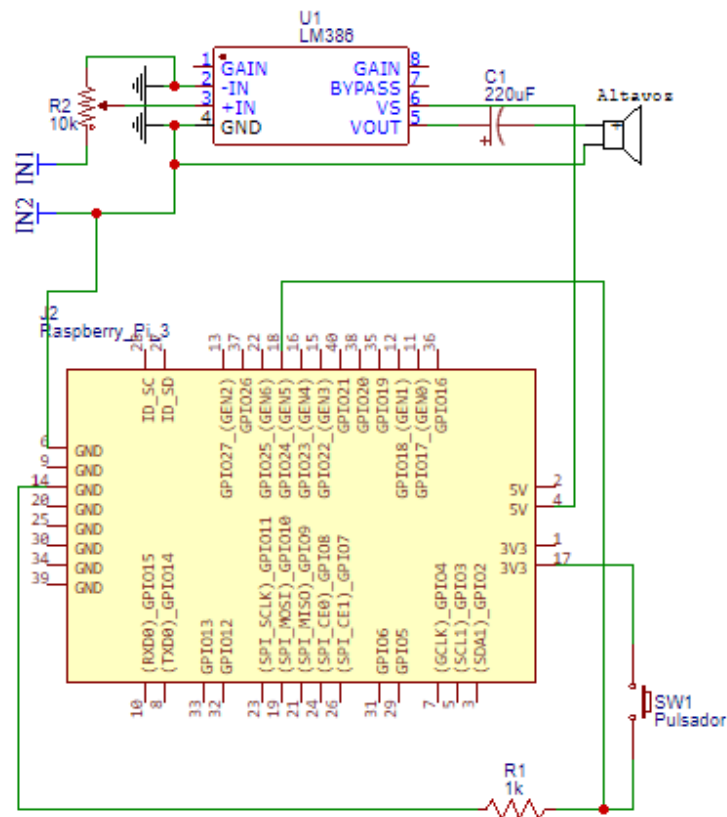


Figura 4-2: Esquema de conexiones al puerto GPIO

Realizado por: Solano, G. 2018

2.1.2.5 Esquema electrónico del dispositivo 3D

A continuación se visualizan todas las conexiones explicadas previamente, en un esquema 3D:

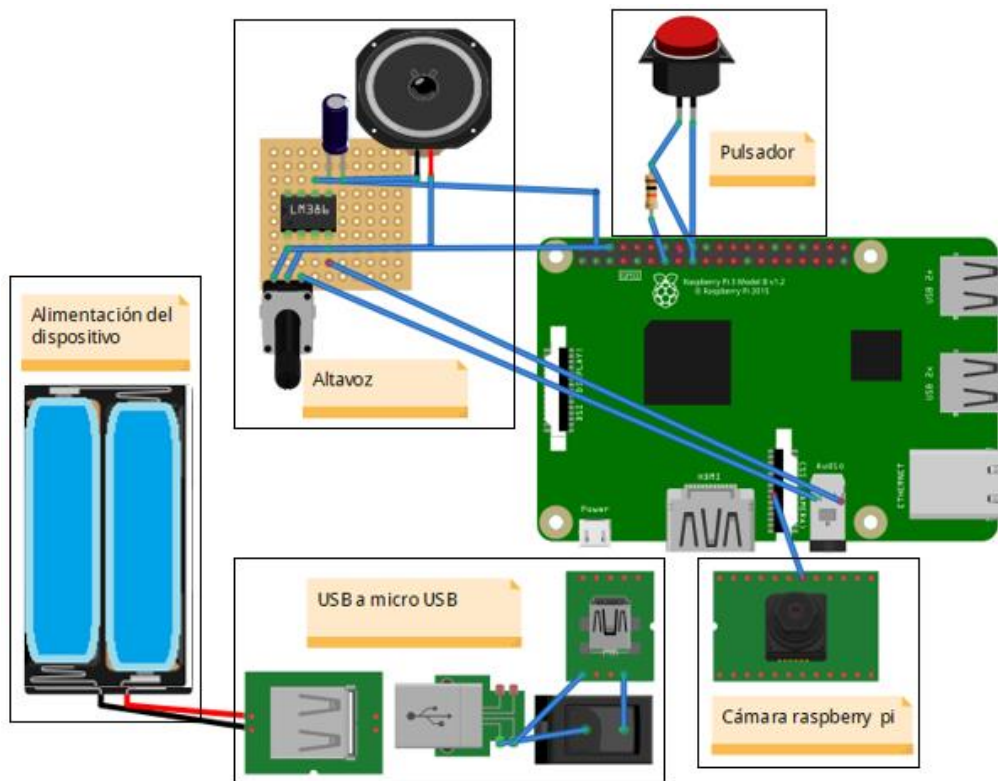


Figura 5-2: Esquema electrónico del sistema en 3D

Realizado por: Solano, G. 2018

2.1.3 *Diseño de la estructura del dispositivo*

El diseño de los componentes del dispositivo se lleva a cabo en Solid Works debido a que esta herramienta fue impartida en un semestre de estudio de la carrera Ingeniería Electrónica en Control y Redes Industriales, además a partir del diseño 3D, se pudo construir el dispositivo con PLA como material de impresión.

Una vez que se cuenta con todos los componentes necesarios para armar el dispositivo, se procede al diseño del case o carcasa en donde son ubicados cada uno de ellos con el fin de protegerlos y permitir la manipulación el dispositivo en conjunto. Ahora bien, el paso consecutivo es la medición de todos los componentes y ubicarlos en lugares aproximados de cómo se quisiera que se obtuviese el resultado final. De este modo se puede ya establecer un diseño.

El diseño cuenta con varios compartimentos en los cuales va ubicado cada componente. La Raspberry pi posee uno el cual junto a la base del dispositivo permite ubicar la placa y añadirle a esta sus demás elementos para formar el dispositivo.

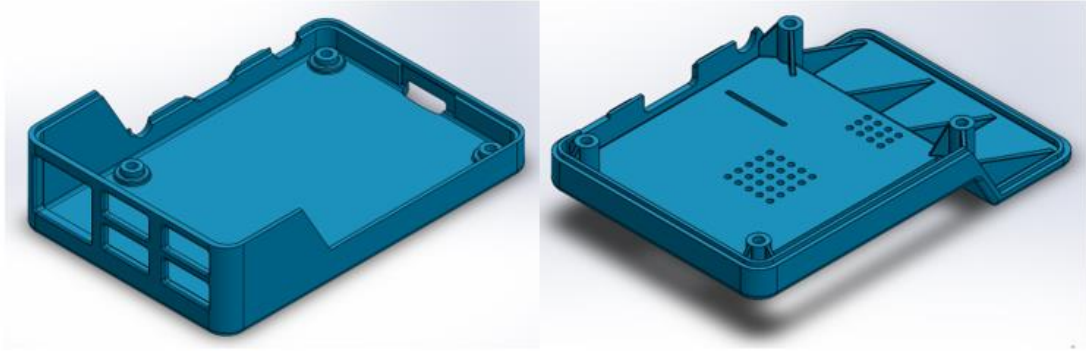


Figura 6-2: Diseño del compartimento para la placa Raspberry pi

Realizado por: Solano, G. 2018

Se desarrolla también una estructura únicamente para la batería, esta cuenta con un puerto USB y microUSB. El compartimento se ubica a presión y es extraíble, permitiendo conectar la batería a la raspberry mediante el USB mientras se usa el dispositivo y cuando esta se agote, se extrae el compartimento y se carga, evitando inconvenientes como dañar la placa debido a la alimentación al cargar la batería mientras se usa el dispositivo.

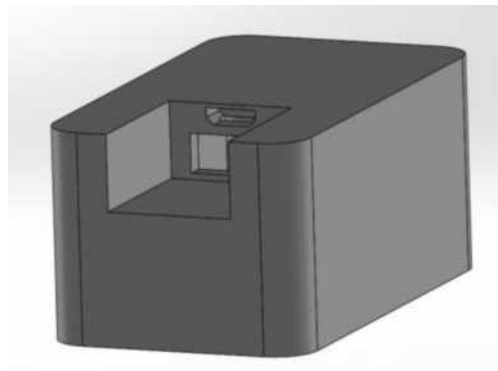


Figura 7-2: Compartimento para la batería del dispositivo

Realizado por: Solano, G. 2018

La cámara también cuenta con su carcasa para protegerla y ubicarla en el dispositivo. En cuanto al altavoz y su circuito, el primero va ubicado al lado de la cámara, y para que el sonido pueda ser emitido, la tapa del dispositivo cuenta con agujeros para el parlante, mientras que el circuito será ubicado sobre la carcasa de la raspberry. La base del dispositivo es una estructura plana rectangular.

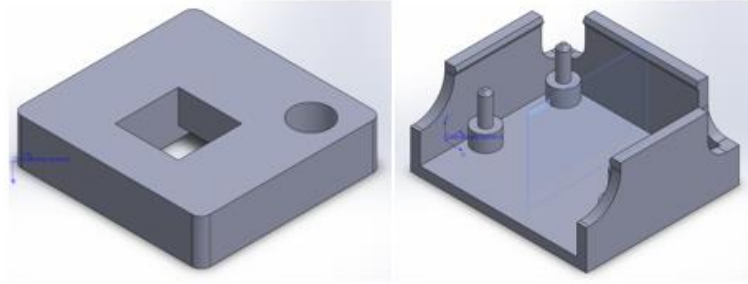


Figura 8-2: Diseño del compartimiento para la cámara Raspberry

Realizado por: Solano, G. 2018

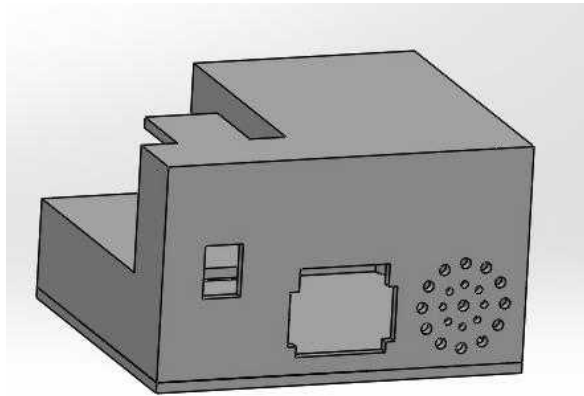


Figura 9-2: Diseño del dispositivo

Realizado por: Solano, G. 2018

Adicionalmente se diseñó dos pequeños sujetadores que sirven como soporte para una banda que pasa por el dispositivo, esto con el objetivo de que el usuario pueda manipular de mejor manera la estructura.

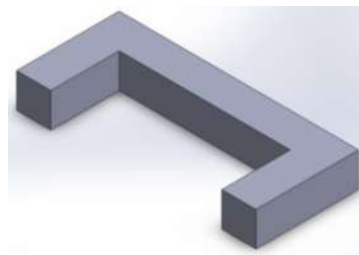


Figura 10-2: Diseño del sujetador de banda

Realizado por: Solano, G. 2018

Pues bien, todas las piezas desarrolladas tienen una ubicación en cuanto a su estructura, por ello en las siguientes figuras se muestra el montaje y resultado final del dispositivo diseñado

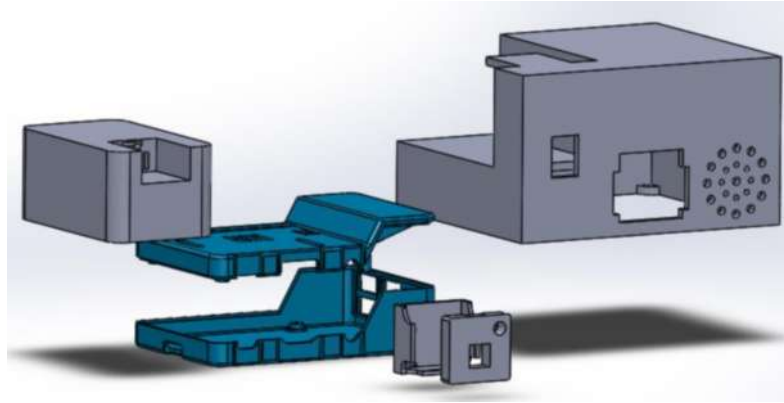


Figura 11-2: Montaje del dispositivo

Realizado por: Solano, G. 2018

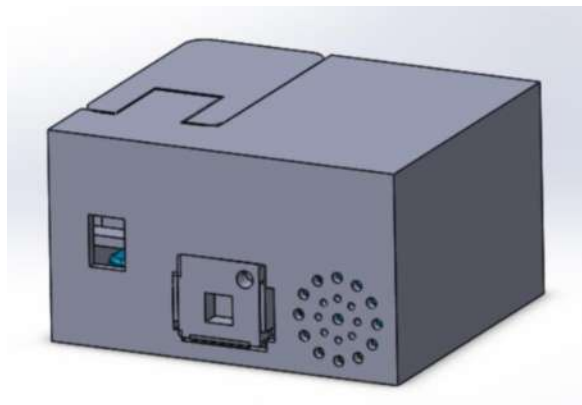


Figura 12-2: Dispositivo armado

Realizado por: Solano, G. 2018

En su estructura final el dispositivo cuenta con 92mm de ancho, 106mm de largo y 61mm de alto. Véase en el Anexo C, el montaje del dispositivo.

2.2 Software

En este apartado se describe el sistema de visión artificial además del procesamiento inteligente a través de un clasificador lineal. El gráfico 2-2 ilustra el proceso global del software que se emplea para la detección de billete y la identificación de cada una de sus denominaciones:

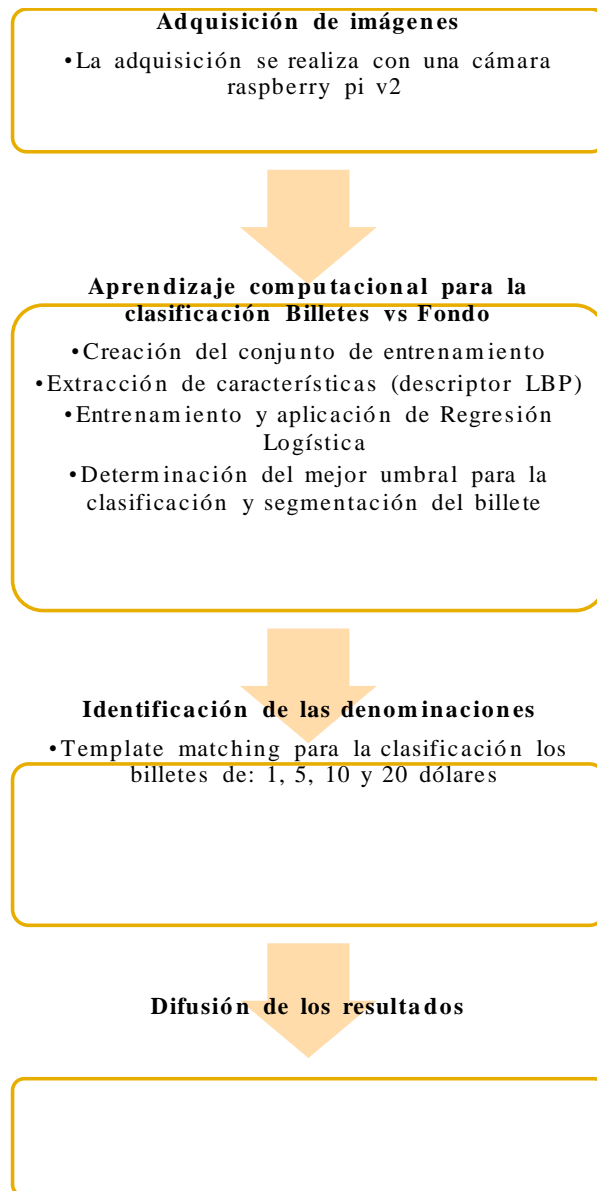


Gráfico 2-2: Desarrollo del sistema detección de billetes y su denominación

Realizado por: Solano, G. 2018

2.2.1 *Adquisición de imágenes*

La adquisición de imágenes se realiza con una cámara raspberry pi v2. Esto es importante de resaltar ya que para la clasificación del billete vs fondo, específicamente el conjunto de entrenamiento se desarrolla con capturas de dicha cámara, permitiendo una mejor clasificación de lo que se haría con imágenes externas obtenidas desde la web.

La resolución que se utiliza para captar la imagen a analizar es de 704 píxeles de ancho x 672 de alto. Esto debido a que sólo se utiliza una imagen apiramidada para poder disminuir el tiempo de ejecución. Para el conjunto de entrenamiento se ha determinado una imagen de 320x170 píxeles.

2.2.2 *Aprendizaje computacional para la clasificación Billete vs Fondo*

Las denominaciones de los billetes de dólar a identificar son de 1, 5, 10 y 20 dólares, puesto que pertenecen al grupo de más utilizados en el Ecuador (Villalba Mateo, 2015). Por ello se las usa de forma frontal y posterior como imágenes positivas para el entrenamiento de un clasificador.



Figura 13-2: Billetes de dólar de 1, 5, 10 y 20 dólares, frontales y posteriores

Realizado por: Solano, G. 2018

Como primer paso para la detección del billete, se debe separar los billetes del fondo de la imagen, para luego poder evaluar sus características más representativas y diferenciar cada uno de ellos en cada una de sus denominaciones. Este proceso de clasificación se logra a través de la regresión logística que a continuación se explica:

2.2.2.1 *Creación del conjunto de entrenamiento*

La regresión logística pertenece al llamado aprendizaje supervisado, por ello, lo primero que se debe hacer es extraer imágenes de acuerdo a su clase, en este caso las clases serían billete y fondo, estas dos deben estar bien diferenciadas, es decir en el conjunto billetes, únicamente debe haber

imágenes de billetes, todo lo contrario para el conjunto fondo. Se las separa en carpetas diferentes para la extracción de características y posterior entrenamiento.

Las imágenes correspondientes a billetes son llamadas imágenes positivas, mientras que las correspondientes al fondo son llamadas imágenes negativas. Es importante mencionar que las positivas deben conservar cierto marco del fondo, ya que esto conlleva mejores resultados en la clasificación. Tanto imágenes positivas como negativas deben tener el mismo tamaño, en este caso el conjunto de entrenamiento tiene 320 píxeles de ancho y 170 de alto.



Figura 14-2: Ejemplos de imágenes positivas

Realizado por: Solano, G. 2018



Figura 15-2: Ejemplos de imágenes negativas o fondo

Realizado por: Solano, G. 2018

Nótese que las imágenes positivas y negativas han sido tomadas con la cámara en entornos no controlados esto para tener mejores resultados ya que el dispositivo planteado actúa en dicho entorno. Dichas imágenes además contienen a billetes con diferente orientación e iluminación, para tener un conjunto de entrenamiento diverso.

La cantidad de imágenes positivas usadas es de 155, mientras que de negativas 349. Aunque se suelen usar miles de imágenes para este proceso de aprendizaje, es importante entender también que se obtiene un mejor resultado con imágenes reales, aunque sean en menor número. Es más se

podiera realizar este proceso teniendo como base 100 imágenes positivas considerando escenarios aplicativos reales y diversidad entre imágenes. (PUTTEMANS, Steven)

Se han realizado también pruebas con conjuntos de entrenamiento con imágenes de 160x90 y 230x135 píxeles, pero no se obtuvo buenos resultados al momento de clasificar entre fondo y billete debido al tamaño reducido de las imágenes y por lo tanto de los billetes capturados.

2.2.2.2 *Extracción de características del conjunto de entrenamiento*

Para la extracción de características se ha escogido a una variación de LBP como descriptor, este es el LBP invariante a la rotación, por a sus buenos resultados ante cambio de iluminación y bajo costo computacional. Además el entorno no controlado hace de este descriptor una muy buena herramienta, ya que no importa si el billete esta recto o con algún ángulo de giro, que esté derecho o de cabeza.



Figura 16-2: Imagen en escala de grises y aplicado LBP invariante a la rotación

Realizado por: Solano, G. 2018

En la figura 16-2 se puede apreciar una imagen en escala de grises a la cual se le ha aplicado el descriptor LBP invariante a la rotación. Como se puede visualizar en la imagen resultante se obtienen valores cercanos a 0 que se plasman en tonos oscuros.

En el gráfico 3-2 se describe el proceso de obtención del descriptor LBP invariante a la rotación, este es aplicado a todos los elementos del conjunto de entrenamiento y posteriormente para la segmentación del billete en una imagen. A través de este proceso se reducen los elementos del vector de una imagen con 256 posibles valores de una imagen en escala de grises o intensidad a 59.

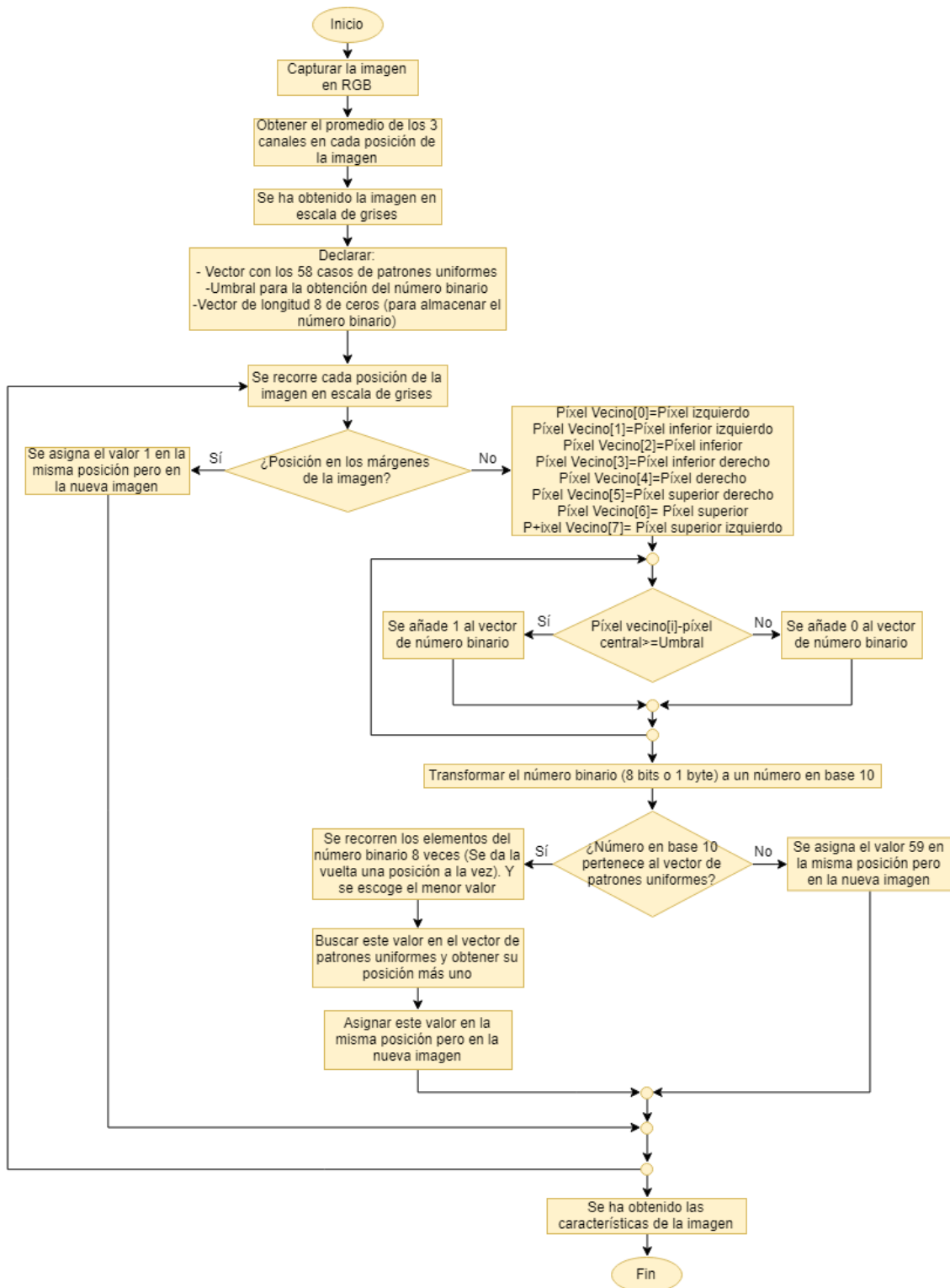


Gráfico 3-2: Diagrama de flujo del descriptor LBP invariante a la rotación

Realizado por: Solano, G. 2018

Como primer paso se captura la imagen en RGB, y se aplica escala de grises, este proceso se consigue obteniendo el promedio de los 3 canales, reduciendo la imagen a un solo. Luego se aplica LBP uniforme, este descriptor se basa en la intensidad de sus píxeles vecinos y depende de

cierto umbral, por ello se declara: un vector con los 58 casos de patrones uniformes, el umbral para obtener el número binario y un vector donde se almacenará cada dígito del número binario.

Se considera un patrón uniforme cuando analizando cada dígito de su número binario existen hasta dos 2 transiciones que pueden ser de 0 a 1 o de 1 a 0, según esto se pueden dar 58 posibilidades que traducido a un número en base 10 son:

Tabla 1-2: Descripción de patrones uniformes y no uniformes

Patrones uniformes (58 casos)	Patrones no uniformes
0, 1, 2, 3, 4, 6, 7, 8, 12, 14, 15, 16, 24, 28, 30, 31, 32, 48, 56, 60, 62, 63, 64, 96, 112, 120, 124, 126, 127, 128, 129, 131, 135, 143, 159, 191, 192, 193, 195, 199, 207, 223, 224, 225, 227, 231, 239, 240, 241, 243, 247, 248, 249, 251, 252, 253, 254, 255	Cualquier otro valor

Realizado por: Solano, G. 2018

El siguiente paso es recorrer cada posición de la imagen, por diseño del algoritmo se ha asignado a los márgenes de la nueva imagen (descriptor) un único valor puesto que no tendrán todos los vecinos a comparar. Para aquellos casos en los que sí existan todos sus vecinos se irán comparando de modo anti horario.

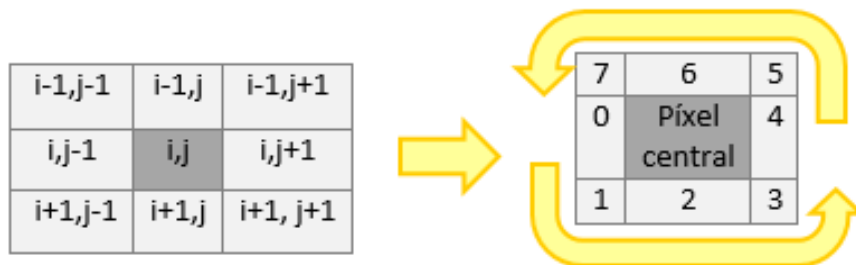


Figura 17-2: Posiciones de los píxeles y orden de comparación

Realizado por: Solano, G. 2018

En la figura 17-2, se pueden apreciar las posiciones de los píxeles vecinos con relación al píxel central de posición i,j , también se muestra el orden en el que se lleva a cabo la comparación, cabe resaltar que este es un orden arbitrario, es decir se puede empezar en cualquier posición siempre y cuando se aplique en toda la imagen. Bien, ahora se resta cada píxel vecino menos el píxel central, y se compara con un umbral arbitrario también, en este caso 8.

Comparando cada diferencia con el umbral se obtienen 8 dígitos (de 1 y 0) asignados a las mismas posiciones de la comparación, este número binario se convierte a número en base 10 y se busca si pertenece al vector de patrones uniformes. Si no pertenece quiere decir que se trata de un patrón no uniforme, entonces se asigna el valor 59. Si pertenece al vector se procede a rotar los ocho dígitos del número binario, uno a la vez.

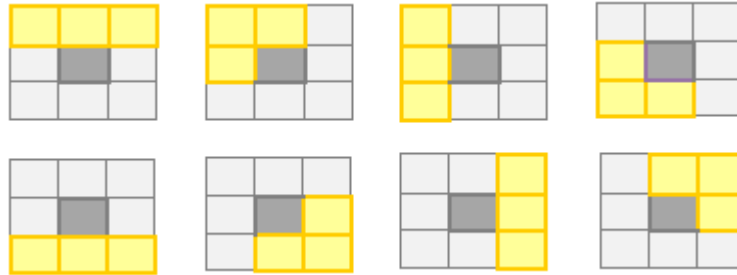


Figura 18-2: Ejemplo de rotación del número binario

Realizado por: Solano, G. 2018

La figura 18-2 muestra un ejemplo de cómo se rota el número binario. Los recuadros de color amarillo pueden representar unos o ceros, independientemente del valor que corresponda, lo importante es que tienen dos transiciones, por lo tanto son patrones uniformes. Se desplazan dichos valores 8 veces, de estos se transforma el número binario a base 10 y se escoge el menor valor, este se busca en el vector de patrones uniformes del cual se extrae la posición más uno.

De esta forma se obtienen los valores más bajos de los patrones uniformes y se incrementa la invariabilidad a la rotación de las imágenes a analizar, pues los histogramas obtenidos para dichas imágenes son idénticos independientemente de la orientación que tome el objeto.

A continuación un ejemplo de la invariabilidad ante la rotación de una imagen y su respectivo histograma en la imagen 19-2, para ello se ha tomado una imagen horizontal y se aplicó LBP invariante a la rotación, así mismo se gira la imagen horizontal aplicando igualmente el descriptor. Los histogramas obtenidos para ambos casos son prácticamente iguales.

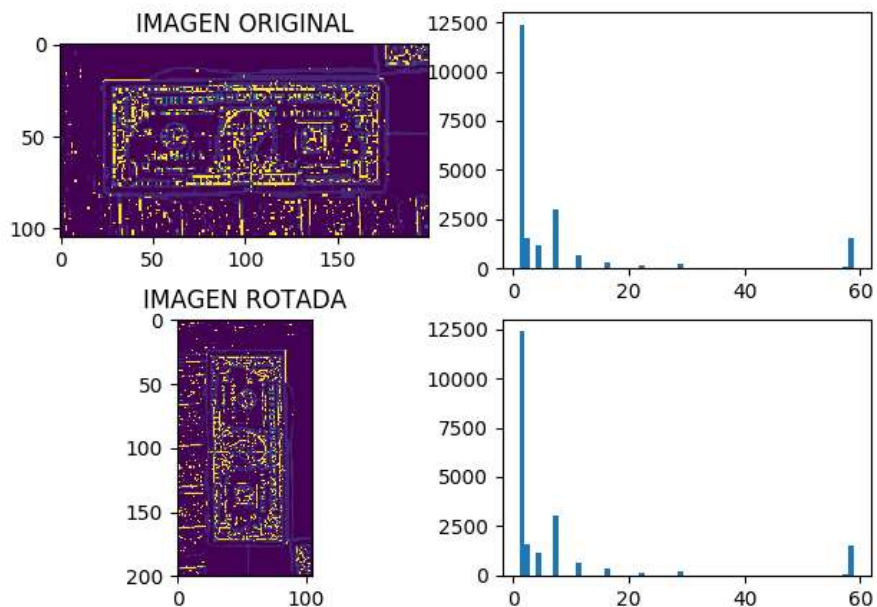


Figura 19-2: Ejemplo de LBP invariante a la rotación

Realizado por: Solano, G. 2018

A través de este proceso se obtiene el descriptor con 59 elementos, mucho menos que los 256 que se tenía con la escala de grises, logrando que los pasos posteriores se ejecuten rápidamente a comparación de una imagen de intensidad. La figura 20-2 plasma lo planteado, pues al analizar el histograma de la imagen en escala de grises y la del descriptor LBP, el primero cuenta con más elementos que el segundo.

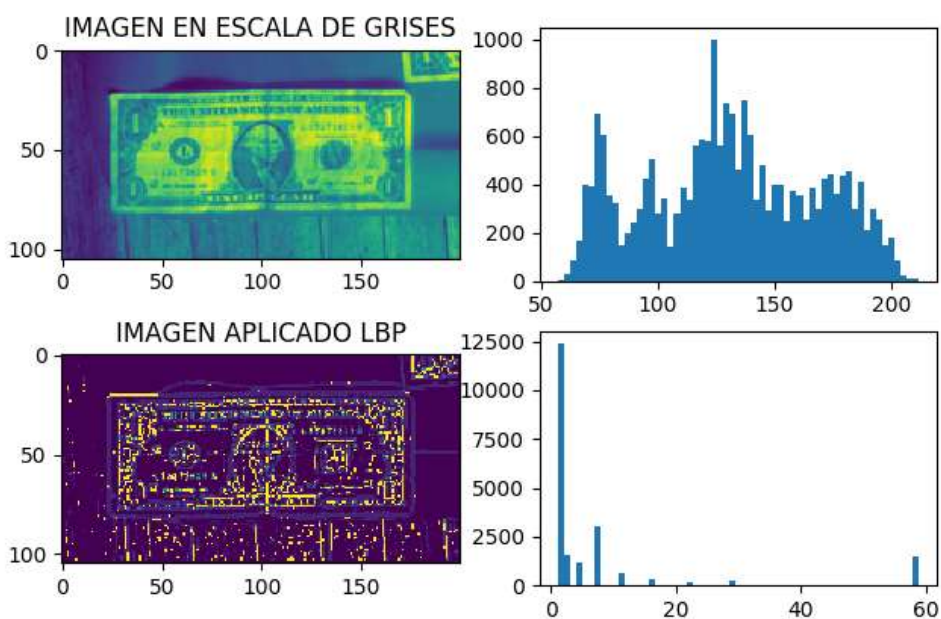


Figura 20-2: Imagen e histograma en escala de grises y aplicando LBP

Realizado por: Solano, G. 2018

Este proceso es programado usando la biblioteca Numpy para la manipulación de matrices y operaciones numéricas, mientras que OpenCV únicamente para leer la imagen a procesar. Una vez que se programa el descriptor hay que aplicar este a cada una de las imágenes positivas y negativas del conjunto de entrenamiento, con el objetivo de extraer características que son usadas para entrenar al clasificador que se explica posteriormente.

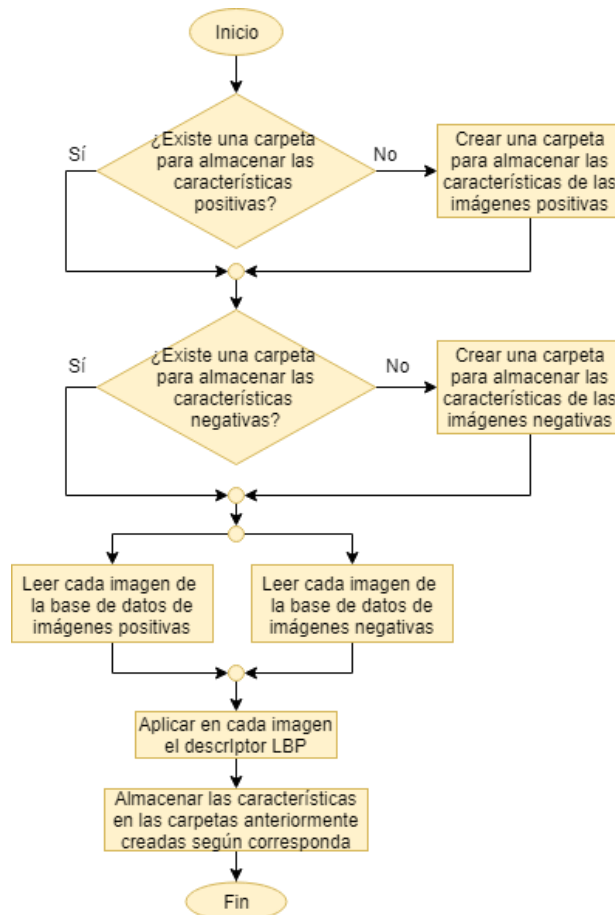


Gráfico 4-2: Diagrama de flujo del proceso de extracción de características

Realizado por: Solano, G. 2018

Ahora se inicia el proceso de extracción de características como lo muestra en el gráfico 4-2, para ello se debe crear una carpeta en donde vayan a almacenarse aquellas extraídas de las imágenes positivas e imágenes negativas, esto se logra a través de la biblioteca Os. Luego se lee cada imagen tanto de la base de datos positivas que corresponde a billetes, y la base de datos de negativas que corresponde al fondo.

Seguidamente se les aplica el descriptor LBP a todas las imágenes almacenándolas por separado, las características de las imágenes positivas en una carpeta y las negativas en otra. El almacenamiento se logra usando la biblioteca Pickle.

2.2.2.3 Entrenamiento y aplicación de Regresión Logística

Como clasificador se ha elegido la regresión logística debido a su rapidez, que es un factor importante en esta aplicación y las buenas respuestas al clasificar. Puesto que se tiene dos clases (billetes y fondo), se puede aplicar este método para su separación.

A continuación el desarrollo del clasificador lineal llamado Regresión Logística:

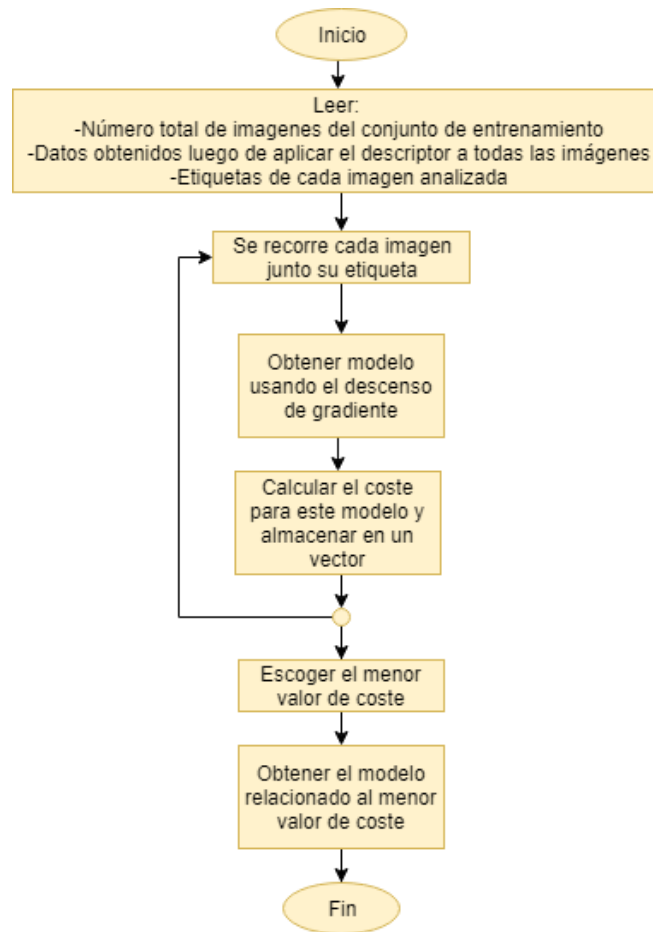


Gráfico 5-2: Diagrama de flujo de la regresión logística

Realizado por: Solano, G. 2018

La regresión logística es un clasificador lineal que permite separar dos clases en este caso los billetes del fondo. Debido a que los dólares tienen características y colores muy similares en todas sus denominaciones es posible agrupar todas ellas para entrenar al clasificador y diferenciarlo del fondo, pero antes de iniciar el proceso con el conjunto de entrenamiento es necesario programar en sí la regresión logística.

El objetivo de este clasificador es determinar el modelo con el menor coste de error para multiplicarlo por el descriptor y según cierto umbral determinar si dicho descriptor es o no un billete. Pues bien se comienza con el proceso de obtención de este clasificador. El primer paso es leer la información del conjunto de entrenamiento, es decir: el número total de imágenes, el conjunto de entrenamiento junto sus etiquetas.

Si una imagen es positiva se le asocia una etiqueta de 1, mientras que si es negativa una etiqueta de 0, lo que ayuda a diferenciar las clases. El siguiente paso es leer cada elemento con su etiqueta. Cabe recordar que los elementos de los que hace referencia son de los obtenidos luego de la extracción de características. Entonces el conjunto de entrenamiento viene dado por la ecuación 5.1, donde x son los descriptores, e y las etiquetas.

Luego lo que se debe hacer es anteponer un 1 a cada vector del descriptor, para obtener un modelo de la misma longitud, es decir ya no será de 59 sino de 60 elementos. Con esta información se procede a encontrar el modelo para ese descriptor, para ello se utiliza el descenso de gradiente de una función de costo, para entender esto se debe explicar dicha función en primer lugar.

La función de costo determina la cantidad de error ante una aproximación, en este caso se debe determinar si es un billete 1, caso contrario 0, siendo estos valores ideales. Ahora bien se necesita el valor real o aproximado que viene dado por la ecuación 7.1, correspondiente a la función logística o sigmoide. Para entrenar el clasificador la x es la multiplicación de los vectores: descriptor x , y el modelo w : $w^T x$, obteniendo un valor escalar.

Una función de costo sería la etiqueta menos el aproximado, entonces se calcularía por separado para billete y fondo, pero se puede usar una función compacta en la que intervengan ambas etiquetas tal como la ecuación 6.1.

Donde $h_w(x^j)$ es función logística cuyo argumento es $w^T x$, y M la totalidad de muestras. Cabe destacar que esta función de costo compacta es convexa, es decir que tiene un único valor mínimo al cual se debe acercarse más el modelo luego del entrenamiento, esto quiere decir que se encuentra el modelo con el menor coste de error.

Para entrenar el modelo se necesita la derivada de esta función de costo aplicando descenso de gradiente para poder acercarse más al punto mínimo de la función. El descenso de gradiente corresponde a la ecuación 8.1, para así obtener la ecuación 10.1, donde α es el valor de aprendizaje que donde es asignado 0,5.

Del modelo obtenido de longitud 60, se procede a calcular su función de costo y se almacena dicho valor en un vector. Se obtendrán tantos modelos como elementos del conjunto de entrenamiento, luego de todos los valores de coste almacenados y aplicado valor absoluto de donde se elige al de menor valor que corresponde al mejor modelo.

Una vez desarrollada la regresión logística, se sigue con el entrenamiento con las características positivas y negativas anteriormente extraídas. Todo este proceso se programó con la biblioteca Numpy.

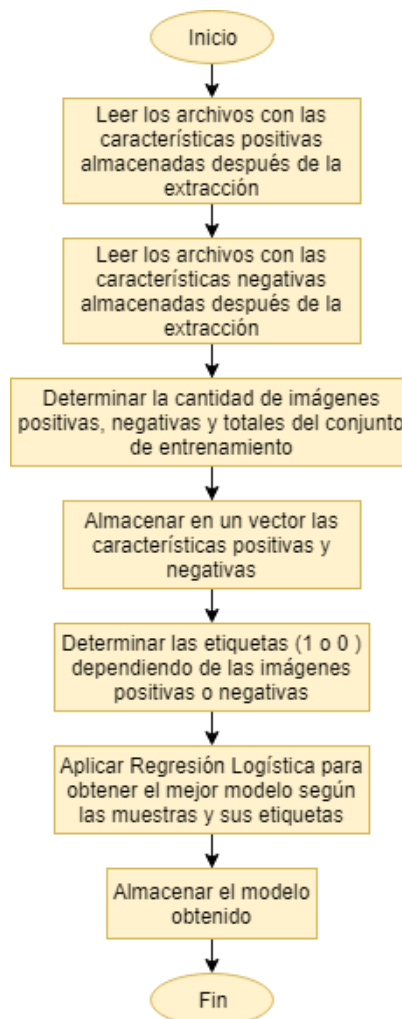


Gráfico 6-2: Diagrama de flujo del entrenamiento del clasificador

Realizado por: Solano, G. 2018

Ahora el proceso de entrenamiento, en primer lugar se lee las características de las imágenes positivas y negativas usando la biblioteca Os. Entonces se determina la cantidad de imágenes positivas, negativas y la suma de estas dos que es la totalidad de elementos del conjunto de entrenamiento.

Luego se declara un vector con información de ambos sub conjuntos y otro con unos y ceros, la cantidad corresponde al número de elementos de cada grupo. De este modo los vectores se corresponderán al orden dado.

Ahora se alimenta la regresión logística con los 3 valores que necesita para determinar el mejor modelo: características positivas, negativas y el número total de elementos. Una vez que se obtenga el modelo con el menor coste, se almacena en una nueva carpeta, aplicando las bibliotecas Os y Pickle.

2.2.2.4 Determinación del mejor umbral para la clasificación y segmentación del billete

Una vez obtenido el modelo es necesario determinar el umbral para compararlo y determinar la existencia o no un billete según la clasificación, analizando los datos obtenidos del conjunto de entrenamiento. El valor que se va a comparar es el obtenido por la ecuación 7.1, que es:

$$h_w(x^j) = \frac{1}{1 + e^{-w^T x}}$$

En un principio se entrenó el modelo con un conjunto de entrenamiento de: 215 muestras positivas o billetes y 49 negativas. El modelo obtenido no tuvo buenos resultados pues hubo un número elevado de falsos positivos.

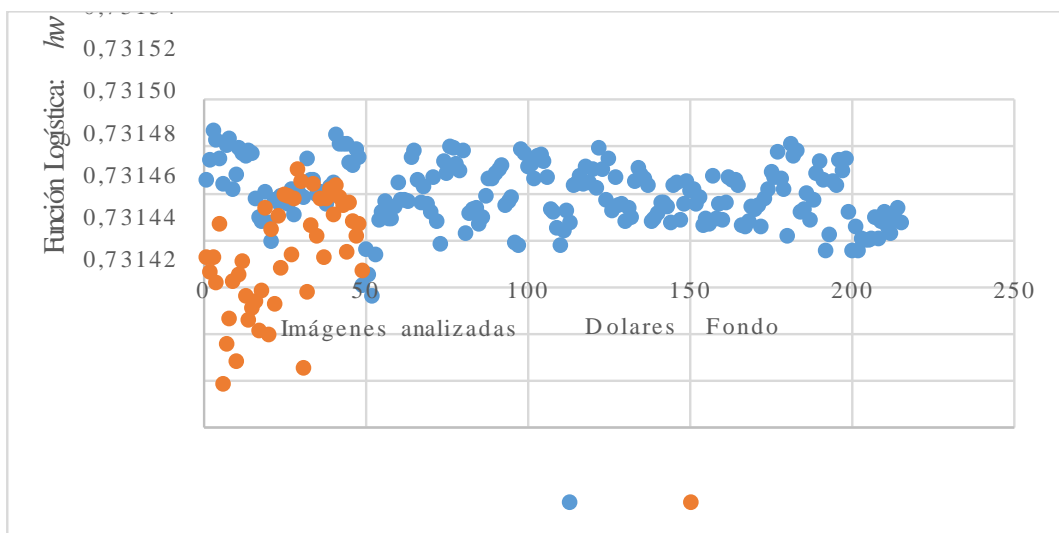


Gráfico 7-2: Muestras positivas y negativas en el primer entrenamiento

Realizado por: Solano, G. 2018

Dados estos resultados se realiza un segundo entrenamiento, tomando ciertas muestras de las que si había acertado en el primer entrenamiento. En este se plateó como conjunto de 155 imágenes positivas y 349 negativas y la información que brindaron fue:

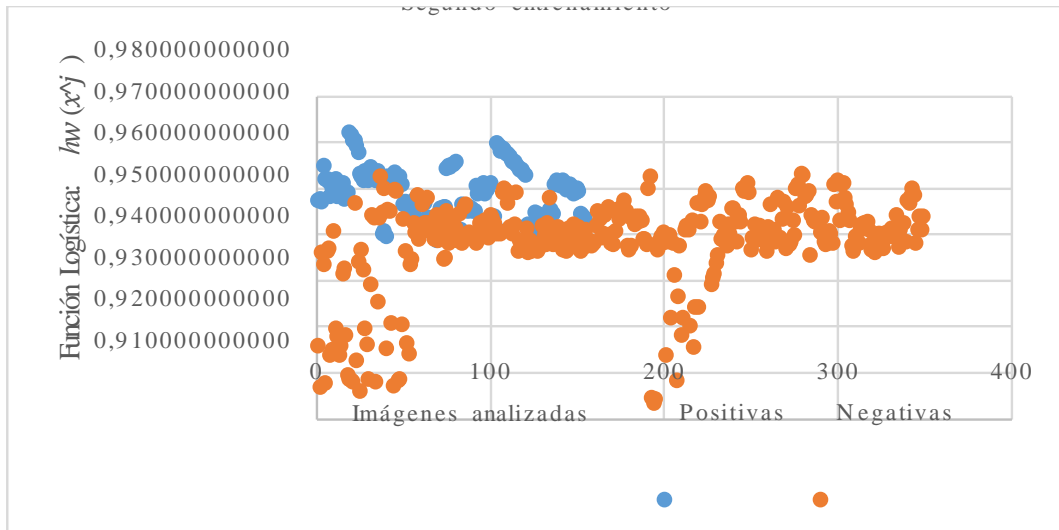


Gráfico 8-2: Imágenes positivas y negativas en el segundo entrenamiento

Realizado por: Solano, G. 2018

Esta información no es la que determina el umbral, pues es con la que se entrenó y se estaría calculando nuevamente el error del modelo obtenido, simplemente sirve para poder tener una guía de donde posiblemente está ubicado dicho umbral, que es aproximadamente entre 0,95 y 0,96, luego a través de varias pruebas con distintas imágenes de billetes de forma frontal y posterior a distinta orientación se establece el mejor umbral.

Para determinar dicho umbral lo que se debe hacer son pruebas reales o test de imágenes (con imágenes de 704x672 píxeles), de este modo se consiguen mejores resultados ya que se trabaja sobre escenarios en los que podría trabajar el dispositivo. En el gráfico 9-2 se describe el proceso realizado que ayuda a determinar el mejor umbral para la clasificación entre billete y el fondo en una imagen.

La razón por la cual se decide usar imágenes de tamaño real en vez de los tamaños de las muestras fue para encontrar el mejor umbral ya aplicando los posibles escenarios donde puede ser usado el dispositivo, y así disminuir el error al momento de clasificar, pues si se toman imágenes únicamente de 320x170 se ingresa ya de por sí una imagen procesada ya sea por recorte y/o redimensionamiento.

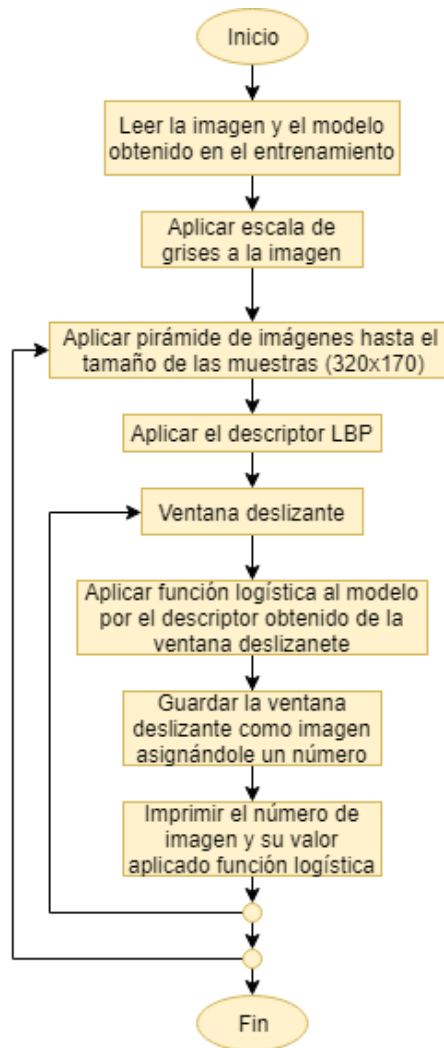


Gráfico 9-2: Diagrama de flujo para determinación del mejor umbral

Realizado por: Solano, G. 2018

Se obtiene la imagen en RGB y se lee el modelo anteriormente almacenado. Luego se aplica escala de grises a la imagen, ahora uno de los nuevos pasos que se aplica es la pirámide de imágenes, esta consiste en redimensionar la imagen conservando su ratio o proporción, para luego aplicar una ventana deslizante de tamaño 320x170.



Figura 21-2: Ejemplo de pirámide de imágenes

Realizado por: Solano, G. 2018

Pueden existir muchísimas imágenes a obtener de la pirámide de imágenes, estas dependen de la resolución de la imagen a analizar. Se hicieron pruebas con 2 y 3 imágenes apiramidadas pero debido al gran tiempo invertido en cada una de ellas se determina usar una imagen de 704x672 píxeles y tomar la segunda pirámide (447x469 píxeles) para con esta aplicar el detector de billetes.

Entonces se captura una imagen con resolución 704x672 y se redimensiona a 447x469 píxeles, a esta última se le aplica el descriptor LBP, para con ello usar ventana deslizante, esta es un recuadro por así decirlo de 320x170 píxeles (del mismo tamaño con que se plantearon las imágenes del conjunto de entrenamiento) que recorre cada imagen obtenida, cuya finalidad es encontrar el billete en una imagen.

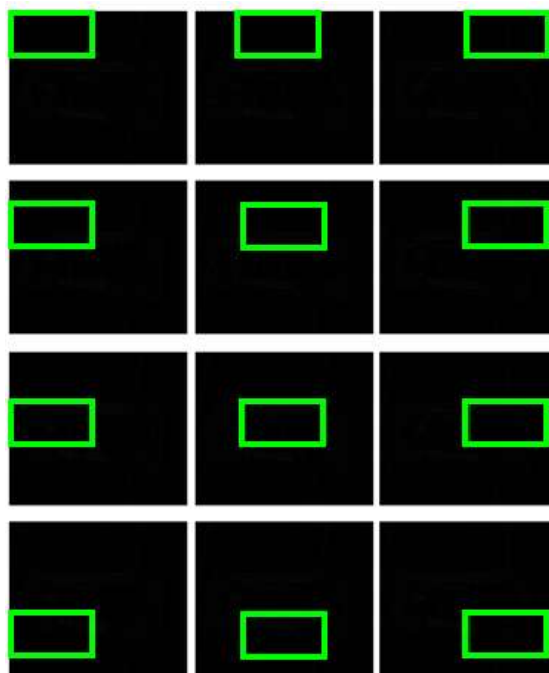


Figura 22-2: Ventana deslizante sobre una imagen aplicada LBP

Realizado por: Solano, G. 2018

Como se puede ver en la Figura 22-2, dada una imagen que se ha aplicado el descriptor LBP invariante a la rotación, se aplica una ventana deslizante representada por un rectángulo de color verde, esta recorre toda la imagen de arriba hacia abajo tratando de encontrar un billete para poder clasificarlo y así diferenciarlo del fondo.

De cada ventana deslizante se extrae información del descriptor, se multiplica por el modelo y es allí donde se compara con el umbral. Tanto la ventana deslizante como las imágenes en pirámide se pueden obtener utilizando la biblioteca Imutils o programando directamente en Numpy.

Aplicando la ventana deslizante a la imagen con un paso de 28 píxeles se obtienen 276 imágenes a multiplicar por el modelo, de las cuales la clasificación debe ser capaz de separar billete del fondo.

Una vez realizados estos pasos, es hora de hacer pruebas con varias imágenes para determinar el mejor umbral, en este caso se tomaron en cuenta 8 imágenes para visualizar el comportamiento de las imágenes positivas y negativas, para cada caso se tomó 10 datos. A continuación se representarán estos datos en el próximo gráfico:

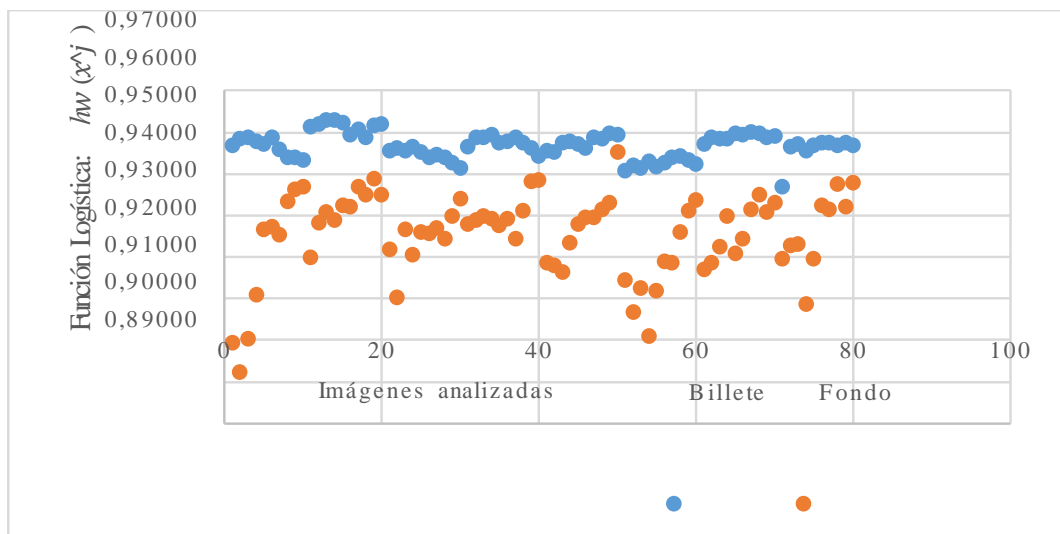


Gráfico 10-2: Pruebas de imágenes para determinar el mejor umbral para la clasificación

Realizado por: Solano, G. 2018

Analizando las imágenes entre billete y fondo, se nota que las imágenes correspondientes a billetes tienen valores más altos que aquellas en donde se encuentran fondos, además tomando en cuenta los valores aplicando función logística se determina que el umbral se encuentra entre 0,9532 y 0,97, de tal modo que estos valores son catalogados como billetes mientras los valores menores como fondo.

Cabe resaltar que la iluminación es un factor muy importante en la clasificación, pues a pesar de buenas detecciones con cambio de iluminación, imágenes muy oscuras de billetes podrían ser clasificadas como fondo. Otro factor a tomar en cuenta es el fondo desordenado, pues se podrían obtener malas detecciones ya que puede clasificar un espacio del fondo como billete.

Una vez determinado el umbral de clasificación se encuentra con un inconveniente, van a existir varias detecciones en la misma imagen del mismo billete, a continuación se analiza algunas de las detecciones de la figura 23-2:



Figura 23-2: Ejemplo de una captura de imagen

Realizado por: Solano, G. 2018

Algunas detecciones encontradas a continuación:



Figura 24-1: Ejemplo de varias detecciones en una misma imagen

Realizado por: Solano, G. 2018

Para poder obtener únicamente una imagen de todo el conjunto de imágenes clasificadas como billete lo que se hizo es almacenar los valores de hw, es decir la multiplicación del modelo por la información de cada ventana, se ordenan estos valores de forma descendente y se elige a la de valor mayor o la posición cero del vector ordenado, con esto se obtiene sólo la mejor detección de todas, segmentando de este modo el billete de la imagen.

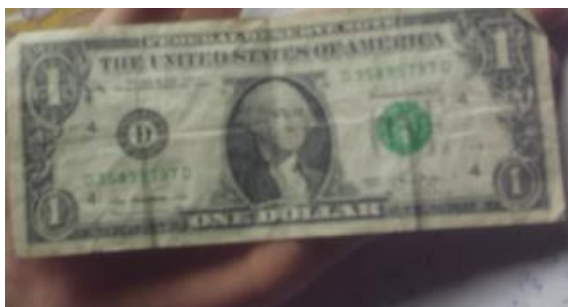


Figura 25-2: Ejemplo de segmentación del billete

Realizado por: Solano, G. 2018

La figura 25-2, representa la segmentación del billete dada una imagen capturada (figura 23-2), es decir esta imagen de 320x170 píxeles, es el resultado de la clasificación entre billete y fondo. Es ahora donde se procede a determinar la denominación de dicho billete, para ello es necesario diferenciar entre los billetes de 1, 5,10 y 20 dólares. Si por el contrario no se detecta un billete en la imagen se debe tomar una nueva captura.

2.2.3 *Identificación de las denominaciones*

Una vez obtenida la segmentación de un billete en una imagen, se plantea la pregunta ¿A qué denominación pertenece el billete?, pues bien para ello se ha utilizado template matching, una técnica que permite determinar el grado de similitud o disimilitud entre una imagen y una plantilla o template, esta trabaja comparando un conjunto de plantillas establecidas por toda la imagen y arroja el mejor valor de emparejamiento de cada una con la imagen.

La razón detrás de la elección de esta técnica para la diferenciación de las denominaciones se debió a que se llevaron pruebas usando detección de bordes, de esquinas y asemejarlos a un rectángulo, sin embargo no se obtuvo buenos resultados debido al desgaste de esquinas, billetes doblados o arrugados. En la figura 26-2, se puede apreciar que los bordes de los billetes son discontinuos haciendo difícil asemejarlas a un polígono.



Figura 26-2: Imagen en RGB y detección de bordes

Realizado por: Solano, G. 2018

Otra técnica que se probó fue las máscaras de color, con estas se puede encontrar determinados colores en una imagen dando como resultado una binarizada en donde el blanco muestra el área donde está presente un color previamente establecido, mientras que el negro muestra la ausencia de este color, sin embargo tampoco cumplió con la diferenciación a causa del entorno no controlado, pues la percepción de los colores puede variar ante la iluminación.

Debido a la manipulación de los billetes en nuestro entorno tampoco se pudo aplicar OCR (Optical Character Recognition), ya que este necesita de caracteres que estén dispuestos de forma recta y esto no se puede obtener en un entorno no controlado, además se necesita discriminar únicamente el billete usando sus bordes y como se explicó en el párrafo anterior tampoco se logró que funcionara.

Otra de las alternativas que se intentaron fue la de usar la técnica One vs All, empleando la regresión logística. Esta consiste en diferenciar cada clase (denominación) y tomar como positivos a cada denominación y fondo a los demás billetes no usados como positivos dando lugar a varios modelos, sin embargo debido a las características tan similares de los billetes no se pudo lograr la diferenciación.

Volviendo al empleo del template matching, se emplea la biblioteca OpenCV, y el método que tuvo las mejores identificaciones fue: `cv2.TM_CCoeff` que es la correlación. Se usaron 7 plantillas para cada denominación de forma frontal y posterior, de las cuales están distribuidas en 0° , 5° , -5° , 10° , -10° , 15° y -15° grados, los cuales se escalan varias veces.

Cada una de las plantillas recorre toda la imagen buscando el mejor emparejamiento el cual es representado con un valor obtenido del número de coincidencias de los píxeles, esto no necesariamente quiere decir que dicho emparejamiento determine la denominación del billete, se deben aplicar todas las plantillas en sus distintas orientaciones, para determinar el mejor resultado.

El gráfico 11-2 muestra cómo se realiza la función para extraer los mejores valores de emparejamiento de una imagen con las plantillas. Para ello se lee la imagen detectada luego de la clasificación con regresión logística, el directorio global de cada denominación, los nombres de las plantillas esto para encontrar su ubicación, el método de template matching, número de veces que se escalan las plantillas y el tamaño a disminuir del tamaño original de las plantillas.

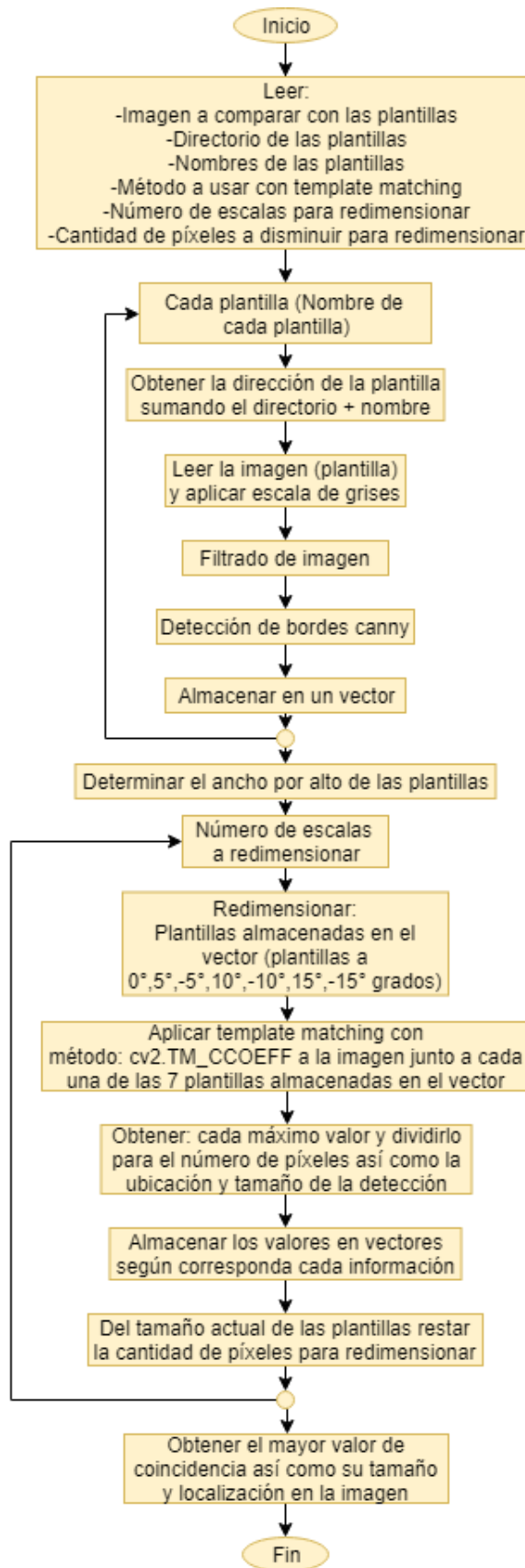


Gráfico 11-2: Diagrama de flujo de función para obtener mejor valor de emparejamiento

Realizado por: Solano, G. 2018

Cada denominación posee imágenes frontales y posteriores diferentes, por lo cual la función que describe el gráfico 11-2 únicamente se alimenta de plantillas frontales o posteriores de 1, 5, 10 y 20 dólares, una a la vez. Para obtener estas se analizan las porciones de los billetes en ambos lados que son diferenciables de las demás denominaciones, además de hacer pruebas sobre ellas. También se determinaron los máximos valores en cuanto a dimensión para cada plantilla.



Figura 27-2: Plantillas usadas para la diferenciación de denominaciones

Realizado por: Solano, G. 2018

En la figura 27-2, muestran las plantillas usadas a comparar con la imagen, a cada una de estas se la giró en distintos grados, además de procesarlas aplicando: escala de grises, filtrado y detección de bordes canny, finalmente se almacena este último dato un vector. Ahora se compara cada una de las plantillas con la imagen y se almacenan todos los valores de las comparaciones.

Hay que tomar en cuenta que los tamaños de cada plantilla fueron determinados tomando como referencia imágenes de 704x672 en donde se muestra la totalidad del billete a una distancia mínima entre el billete y el dispositivo (16cm), y se realiza el redimensionamiento hasta que en el billete segmentado puedan ser identificables cada una de las plantillas sin que este consista un área muy pequeña.

La distancia de 16cm a la que fueron determinadas las plantillas influye en la distancia a las que los billetes van a ser detectados, pues esta es tomada como distancia base para determinar el desempeño del dispositivo.



Figura 28-2: Plantillas de 1, 5, 10 y 20 dólares frontales y posteriores a 0°, 5°,-5°,10°,-10°,15°, -15° grados

Realizado por: Solano, G. 2018

Cada plantilla de denominación frontal o posterior cuenta con 7 plantillas a distintos grados, como lo muestra la figura 28-2, además cada grupo de plantillas posee dimensiones máximas distintas por lo cual se realizan distintos números de escalas, sin embargo para todos se disminuye cierta cantidad de píxeles para realizar las comparaciones nuevamente en cada iteración. Tomando esto en cuenta para todas las denominaciones se realizan entre 50 y 80 comparaciones.

Una vez almacenados todos los valores y guardados en un vector, se ordenan y se elige el máximo valor, determinando el mejor emparejamiento de un grupo de plantillas sobre una imagen. Ahora es cuestión de comparar los 8 valores obtenidos de las comparaciones.

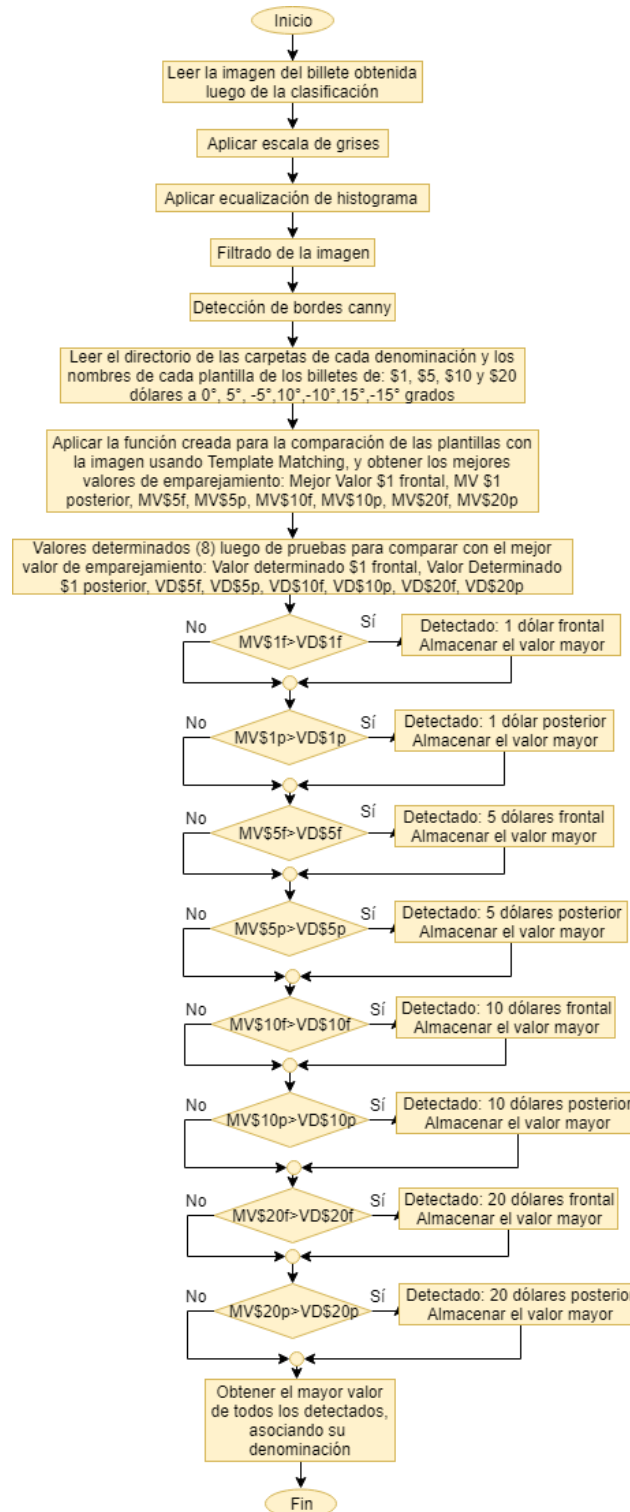


Gráfico 12-2: Diagrama de flujo para determinar la denominación del billete detectado

Realizado por: Solano, G. 2018

El gráfico 12-2 muestra el proceso para obtener la denominación del billete segmentado luego de la detección, este utiliza la función descrita en el gráfico 11-2 que determina el mejor valor de emparejamiento para cada grupo de plantillas. Bien como primer punto se tiene la lectura de la imagen (billete) a comparar, a esta se le aplica escala de grises y eualización de histograma que busca distribuir uniformemente el histograma de la imagen mejorando su contraste.

El siguiente paso es filtrar la imagen usando `cv2.filter2D` que la suaviza, aplicar la detección de bordes canny obteniendo una imagen binarizada (blanco y negro). Se leen los directorios de las plantillas y se aplica la función del gráfico 11.2, se compara los valores determinados (VD\$1f, VD\$5f, VD\$10f, VD\$20f, VD\$1p, VD\$5p, VD\$10p, VD\$20p) con los valores reales (MV\$1f, MV\$5f, MV\$10f, MV\$20f, MV\$1p, MV\$5p, MV\$10p, MV\$20p) de emparejamiento.

Por ejemplo, si el valor máximo de 1 dólar frontal (MV\$1f) obtenido aplicando la función del gráfico 12-2, es mayor a un valor determinado (VD\$1f) entonces se ha detectado un billete de 1 dólar frontal así que se almacena el MV\$1f, y se continúa con las comparaciones entre (MV\$1p) y (VD\$1p), hasta cumplir con la comparación de todas las plantillas.

Se realiza la comparación entre los valores determinados y reales para tratar de eliminar falsas detecciones, almacenando únicamente los valores mayores a los determinados y de todos esto se obtiene un máximo global que corresponde a la denominación del billete, siendo así se puede informar el resultado de la detección.

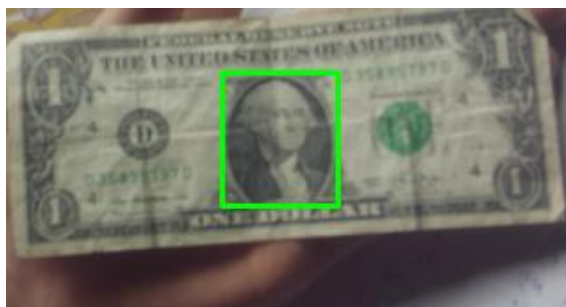


Figura 29-2: Ejemplo de detección de 1 dólar frontal

Realizado por: Solano, G. 2018

Al terminar con las comparaciones si se determina el valor máximo de la comparación de todas las plantillas se procede a informar el resultado de dicha detección, junto a la localización y tamaño para poder visualizar el resultado, no obstante para el dispositivo sólo se necesita informar la denominación. En caso de que no se obtenga ninguna detección se gira la imagen a comparar 180° volviendo a realizar el proceso del gráfico 12-2.

Si no se obtiene ningún buen emparejamiento o valor máximo se debe a capturar una nueva imagen y discriminar el billete del fondo. Es importante mencionar que la técnica de template matching es sensible a cambios de iluminación por lo que pueden haber buenas imágenes obtenidas de la clasificación y aun así es posible que no detecte a que denominación pertenece el billete.

2.2.4 *Difusión de los resultados mediante comandos de voz*

Una vez obtenidos los resultados es necesario informar al usuario acerca de que billete se analizó para ello es preciso usar un módulo de voz de python, denominado Pyaudio. Este permite dar a conocer el resultado mediante archivos wav, que es un formato de audio digital.

Entonces los posibles resultados a obtener dado que se considera billetes que poseen dos estructuras totalmente diferentes (frontal y posterior) en su misma unidad, pero se refiere a un solo billete, por ello se podrá escuchar: “Un dólar”, “Cinco dólares”, “Diez dólares” y “Veinte dólares”.

Además como forma de aviso, el programa al momento de capturar una imagen reproduce un “Click”, para que el usuario sepa que se ha proceder con la ejecución, asimismo se escuchará “Captura nueva imagen”, si ya ha informado el resultado de la denominación o si por el contrario no lo ha encontrado y se necesite una nueva imagen a analizar se reproduce “Captura nuevamente”.

Adicionalmente el usuario sabrá que el dispositivo está listo para usarse una vez encendido mediante el mensaje “Encendido”.

Todos estos comandos de voz están bajo el formato .wav y se reproducirán en distintos momentos de la ejecución según se requieran.

2.2.5 *Diagrama de flujo del funcionamiento del dispositivo*

A continuación se visualiza el funcionamiento general del dispositivo a través del siguiente diagrama de bloques:

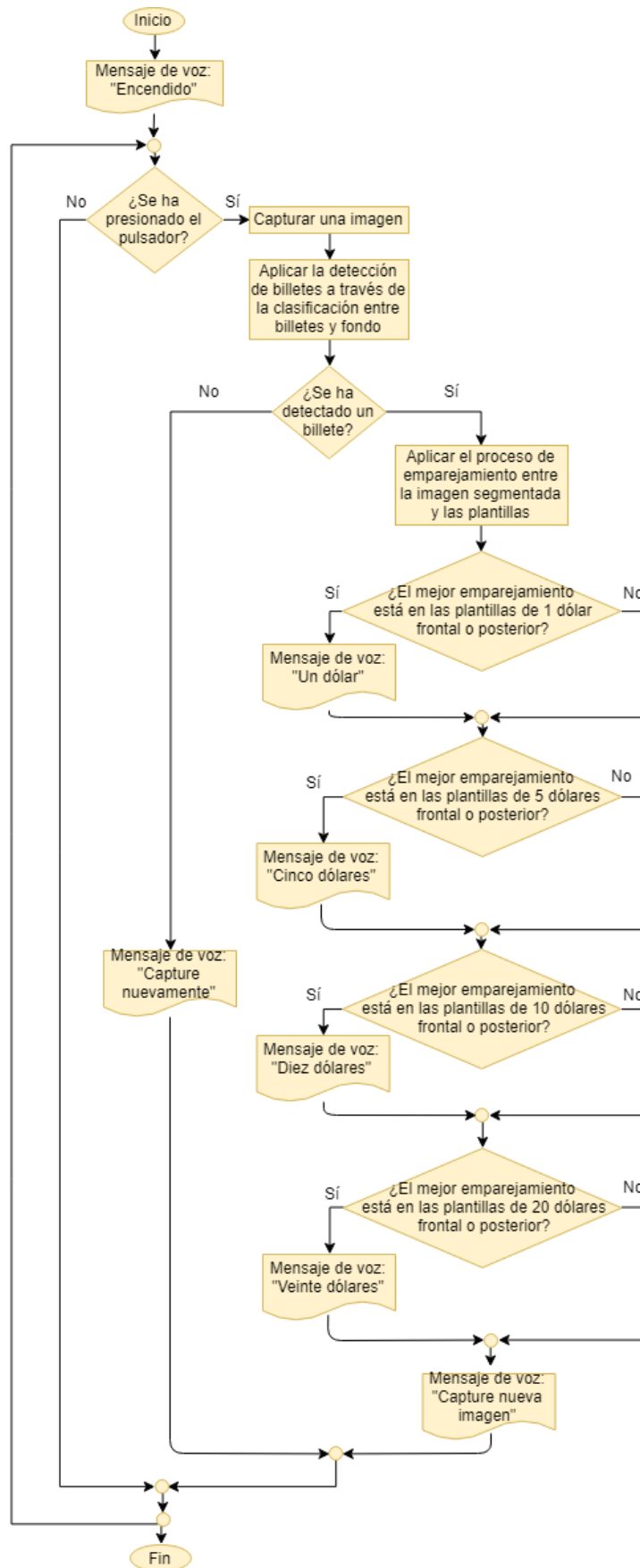


Gráfico 13-2: Diagrama de flujo del funcionamiento del dispositivo

Realizado por: Solano, G. 2018

CAPÍTULO III

3. MARCO DE RESULTADOS

Se ha definido al dispositivo para la detección de billetes como una estructura rectangular, que puede ser empuñado y es portátil, con el mismo se apuntará la cámara en dirección del billete para iniciar la detección, el objetivo del presente capítulo es evidenciar los resultados obtenidos de las pruebas realizadas. Han sido consideradas las denominaciones de 1, 5, 10 y 20 dólares americanos por ser de mayor circulación.

Para lograr una óptima detección se toman en consideración dos factores preponderantes, la distancia del objeto a identificar, así como; el tiempo que se demora en el procesamiento de las imágenes, el dispositivo posee una distancia óptima de detección, menor o mayor a esta la detección se vuelve incierta.

Para realizar las detecciones a distintas distancias, se desarrolló un escenario de prueba constituido por billetes, el dispositivo, un flexómetro y diversos tipos de fondos, las pruebas iniciaron a una distancia de 16cm hasta 31 cm. Para que el dispositivo sea portátil se le acondicionó una batería o power bank y con esta se llevó cabo la medición de tiempo de respuesta.

3.1 Desempeño de la identificación de denominación de billetes

Para determinar si el dispositivo desarrollado cumple con el objetivo de identificación de las denominaciones de los billetes de 1, 5, 10 y 20 dólares se realizó un estudio estadístico basado en contraste sobre la igualdad de varias proporciones. Para ello se evaluaron imágenes de billetes tomadas por el dispositivo a distintas distancias de este.

3.1.1 *Características de las imágenes tomadas para las pruebas*

Para el conjunto de evaluación se tomaron imágenes con billetes de 1, 5, 10 y 20 dólares de forma frontal y posterior. Además cada una de las imágenes posee fondos diferentes, dado que el dispositivo es aplicado en entornos no controlados, cabe resaltar que se capturó también imágenes de billetes con una orientación máxima de -15° y 15° .

A más de estas características para tener un conjunto de imágenes diverso se analizaron billetes con distinto trato en su estructura, esto quiere decir entre ellos estaban billetes casi nuevos, manchados, arrugados y desgastados.



Figura 1-3: Algunas imágenes de prueba

Realizado por: Solano, G. 2018

Las distancias a las que fueron capturadas las imágenes poseen intervalos de 2 cm desde 16 a 31cm de distancia del billete al dispositivo, teniendo en total de 8 intervalos. La cantidad de imágenes que se consideró fue de 120 por intervalo, como se muestra en la tabla 1-3. Se inició a 16cm debido a que las plantillas para la identificación de las denominaciones fueron extraídas de los billetes a esa distancia.

Cabe destacar que de las pruebas realizadas, únicamente se tomaron en cuenta aquellas las identificaciones correctas e incorrectas, se debe aclarar esto debido a que puede darse el caso que el dispositivo no detecte el billete aun capturando la imagen, esto puede suceder ya que la coincidencia no fue buena, por lo tanto no se emite ningún mensaje acerca de la denominación del billete, sino un “capture nuevamente”.

Tabla 1-3: Pruebas de identificación realizadas a distintas distancias del billete al dispositivo

Distancia (cm)	16-17	18-19	20-21	22-23	24-25	26-27	28-29	30-31	Total
Identificaciones correctas	110	98	94	99	101	92	88	71	753
Identificaciones incorrectas	10	22	26	21	19	28	32	49	207
Total	120	120	120	120	120	120	120	120	960

Realizado por: Solano, G. 2018

De los valores descritos en la tabla 1-3, se desglosan en los siguientes gráficos:

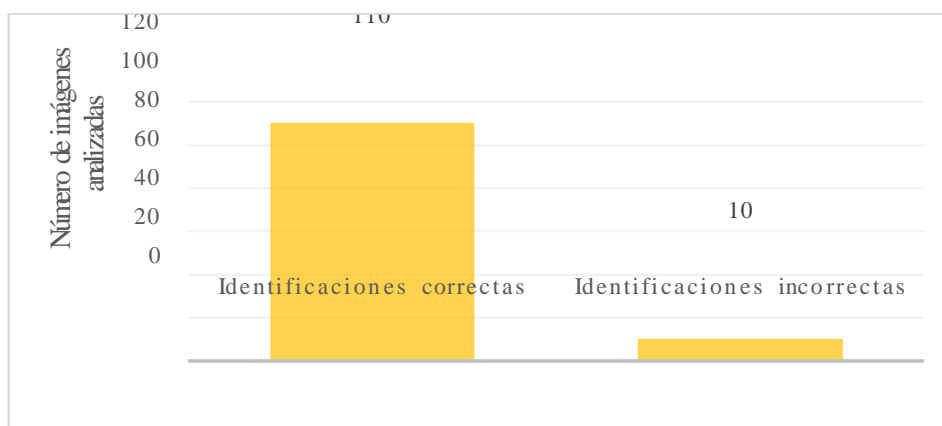


Gráfico 1-3: Identificaciones correctas e incorrectas a 16-17cm del billete al dispositivo

Realizado por: Solano, G. 2018

El gráfico 1-3, ilustra que existen de 120 imágenes con billetes analizados, de los cuales 110 se han identificado correctamente, mientras que 10 lo han hecho de forma incorrecta.

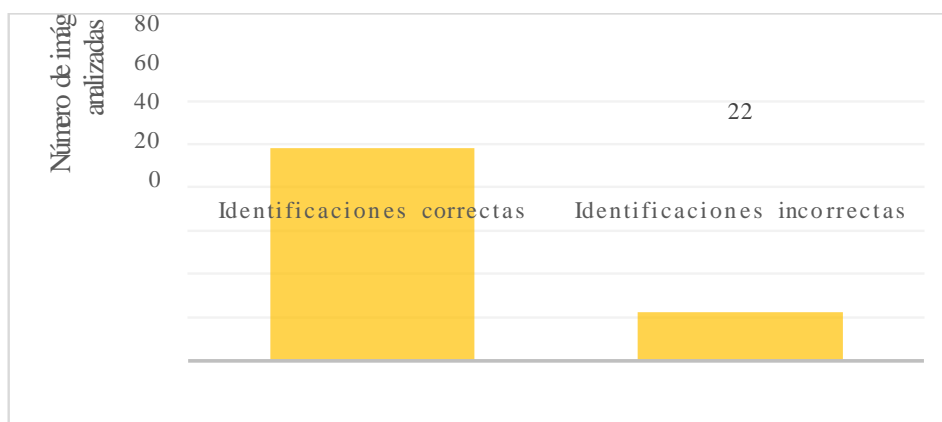


Gráfico 2-3: Identificaciones correctas e incorrectas a 18-19cm del billete al dispositivo

Realizado por: Solano, G. 2018

El gráfico 2-3, ilustra que existen de 120 imágenes con billetes analizados, de los cuales 98 se han identificado correctamente, mientras que 22 lo han hecho de forma incorrecta.

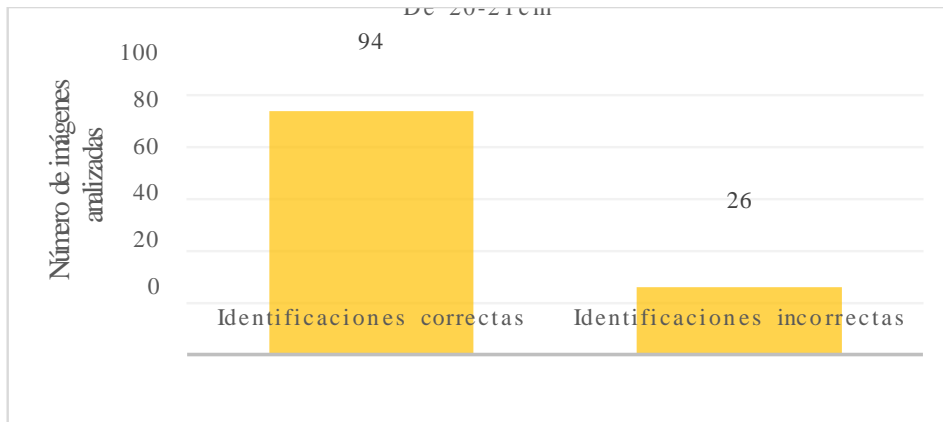


Gráfico 3-3: Identificaciones correctas e incorrectas a 20-21cm del billete al dispositivo

Realizado por: Solano, G. 2018

El gráfico 3-3, ilustra que existen de 120 imágenes con billetes analizados, de los cuales 94 se han identificado correctamente, mientras que 26 lo han hecho de forma incorrecta.

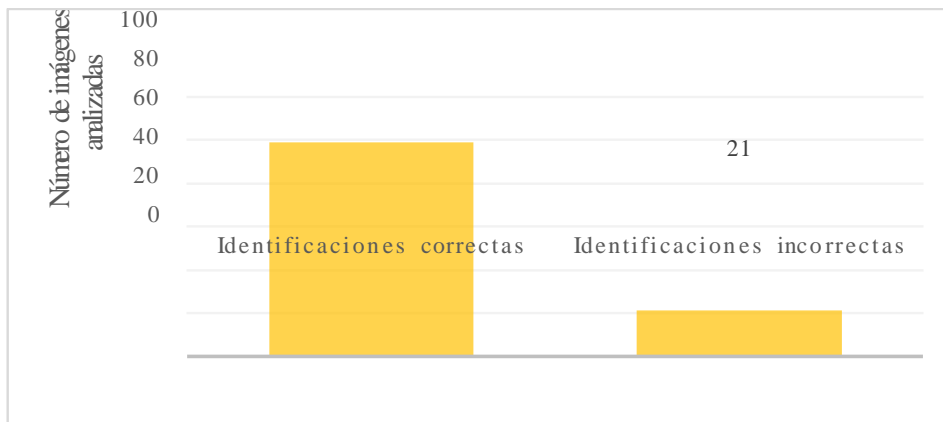


Gráfico 4-3: Identificaciones correctas e incorrectas a 22-23cm del billete al dispositivo

Realizado por: Solano, G. 2018

El gráfico 4-3, ilustra que existen de 120 imágenes con billetes analizados, de los cuales 99 se han identificado correctamente, mientras que 21 lo han hecho de forma incorrecta.

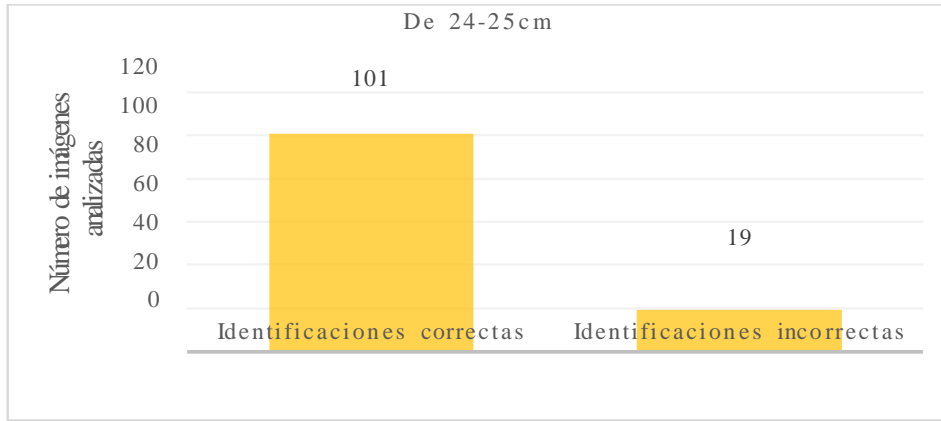


Gráfico 5-3: Identificaciones correctas e incorrectas a 24-25cm del billete al dispositivo

Realizado por: Solano, G. 2018

El gráfico 5-3, ilustra que existen de 120 imágenes con billetes analizados, de los cuales 101 se han identificado correctamente, mientras que 19 lo han hecho de forma incorrecta.

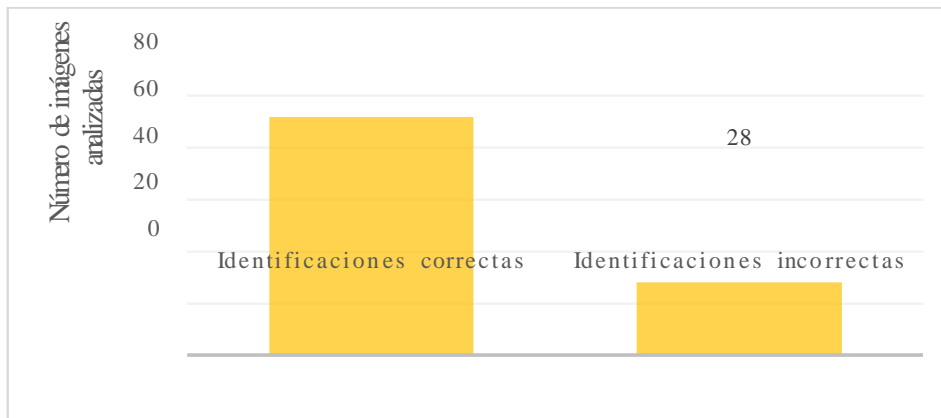


Gráfico 6-3: Identificaciones correctas e incorrectas a 26-27cm del billete al dispositivo

Realizado por: Solano, G. 2018

El gráfico 6-3, ilustra que existen de 120 imágenes con billetes analizados, de los cuales 92 se han identificado correctamente, mientras que 28 lo han hecho de forma incorrecta.

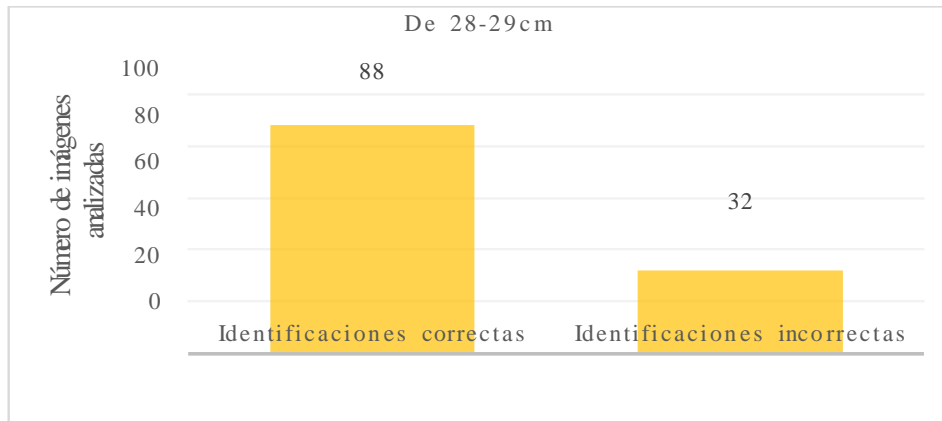


Gráfico 7-3: Identificaciones correctas e incorrectas a 28-29cm del billete al dispositivo

Realizado por: Solano, G. 2018

El gráfico 7-3, ilustra que existen de 120 imágenes con billetes analizados, de los cuales 88 se han identificado correctamente, mientras que 32 lo han hecho de forma incorrecta.

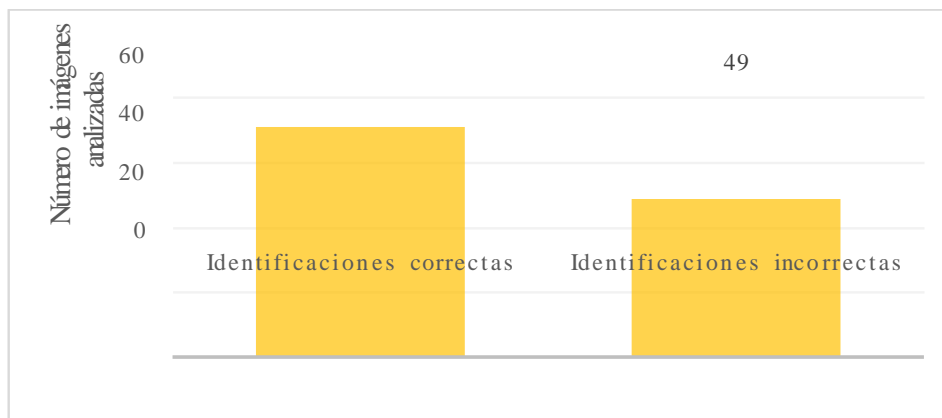


Gráfico 8-3: Identificaciones correctas e incorrectas a 30-31cm del billete al dispositivo

Realizado por: Solano, G. 2018

El gráfico 8-3, ilustra que existen de 120 imágenes con billetes analizados, de los cuales 71 se han identificado correctamente, mientras que 49 lo han hecho de forma incorrecta.

3.1.2 *Contraste sobre la igualdad de varias proporciones*

El contraste de igualdad de varias proporciones determina la homogeneidad o no de los datos, y de este modo encontrar el mejor intervalo de desempeño del dispositivo, tomando en cuenta los datos de la tabla 1-3. Para resolver el problema planteado se realizaron los 5 pasos para aceptación o no de hipótesis, que son: Planteamiento de hipótesis nula y alternativa, significancia, criterio y decisión.

Para la realización de los cálculos se ha utilizado el software estadístico R Studio, debido a que esta herramienta fue impartida en un semestre de la carrera de Ingeniería Electrónica en Control y Redes Industriales.

Se plantea probar si la probabilidad de que se obtengan identificaciones correctas al analizar billetes es la misma para todas las distancias con una significancia de 0,05.

1. **Hipótesis nula:** $p_{\text{Distancia1}} = p_{\text{Distancia2}} = p_{\text{Distancia3}} = p_{\text{Distancia4}} = p_{\text{Distancia5}} = p_{\text{Distancia6}} = p_{\text{Distancia7}} = p_{\text{Distancia8}}$. La probabilidad de identificaciones correctas es la misma para todas las distancias.

Hipótesis alternativa: Al menos una de las probabilidades de identificaciones correctas es diferente.

2. Nivel de significancia: 0,05

3. Criterio: Se rechaza la hipótesis nula si $p\text{-value} < 0,05$, con el estadístico X^2

4. Cálculos (R Studio)

```
correctas=c(110,98,94,99,101,92,88,71)
```

```
prop.test(correctas,n=c(120,120,120,120,120,120,120,120))
```

```
# data: correctas out of c(120, 120, 120, 120, 120, 120, 120, 120)
```

```
# X-squared = 45.0773, df = 7, p-value = 1.321e-07
```

5. Decisión: Dado que p-value es menor que 0,05, **la hipótesis nula se rechaza**, es decir que la probabilidad de identificaciones correctas no es la misma para todos los intervalos de distancia con un nivel de confianza de 95%.

Además el software arroja los siguientes valores de proporción:

Tabla 2-3: Tabla de proporciones dados intervalos de distancia

Distancia (cm)	16-17	18-19	20-21	22-23	24-25	26-27	28-29	30-31
Proporciones	0,91667	0,81667	0,78333	0,825	0,84167	0,76667	0,7333	0,59167

Realizado por: Solano, G. 2018

En la tabla 2-3 se tienen las distintas proporciones de acuerdo al rango de las distancias que se ha puesto a prueba. Siendo el 1 un 100%, cada una de ellas refleja el grado de respuestas correctas que se han obtenido del sistema.

El siguiente gráfico ilustra las proporciones obtenidas:

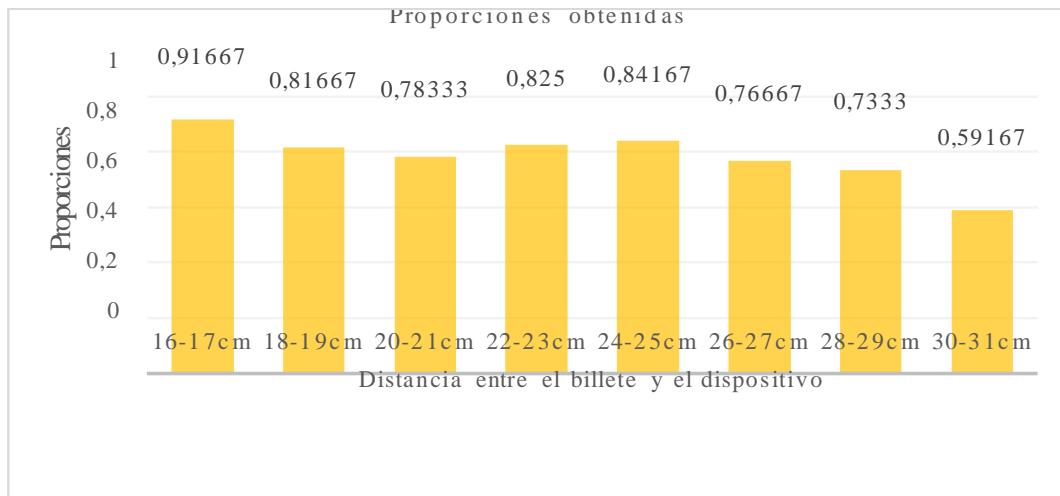


Gráfico 9-3: Proporciones obtenidas en los 8 rangos de análisis

Realizado por: Solano, G. 2018

En el gráfico 9-3 se puede apreciar de mejor manera el grado de proporción de las identificaciones correctas, además de la diferencia que tienen cada una de ellas en respuesta de acuerdo a la distancia. Es claro que de todas estas proporciones, la última es de menor valor con respecto a las anteriores, esto es debido a que de una distancia de 30 a 31cm se han obtenido mayor cantidad de respuestas negativas, que en las demás distancias.

3.1.2.1 Distancias de no funcionamiento del dispositivo

Estadísticamente se puede determinar el buen funcionamiento del dispositivo cuando las identificaciones correctas se ubiquen al menos en un 60% o de una proporción de 0,6. De la tabla 2-3 de proporciones obtenidas, se puede apreciar que la mayoría de intervalos de distancias cumplen con esta condición a excepción de la distancia 30-31cm, que sólo cuenta con una proporción de 0,59167.

Por esta razón se descarta este último intervalo, puesto que no se produce un adecuado funcionamiento del dispositivo. Además las imágenes tomadas a dicha distancia cuentan con una segmentación de billetes mucho más pequeños a analizar, entonces las plantillas para determinar estos también debería ser pequeñas, sin embargo a otras distancias estas pueden desencadenar en malos emparejamientos.

Tomando este análisis como referencia, se prevé también que el dispositivo no tendrá buenos resultados ante distancias mayores a 31 cm, ya que no se determinaron plantillas más pequeñas

para la detección. Esto también pasa con imágenes tomadas a una distancia menor que 16cm, ya que las plantillas planteadas fueron determinadas únicamente desde 16cm de distancia.

3.1.2.2 Distancias de funcionamiento del dispositivo

Una vez descartada la distancia en este caso, de no adecuado funcionamiento, se procede a analizar la proporción de todos rangos de distancia que si superaron el 60%. Esto a través de un contraste sobre una proporción. Para realizar dicho análisis es necesario sumar todos aquellos valores obtenidos de las identificaciones correctas, esto es:

Tabla 3-3: Rango con las mejores identificaciones

Distancias (cm)	16-29
Identificaciones correctas	682
Identificaciones incorrectas	158
Total	840

Realizado por: Solano, G. 2018

En la tabla 3-3 se puede apreciar los valores totales de las identificaciones correctas e incorrectas de 16 a 29cm, pues la distancia 30 a 31cm se descartó por no cumplir con un 60% de buenas identificaciones. Con estos datos, se procede a plantear lo siguiente:

De un total de 840 imágenes analizadas a una distancia de 16-29cm, el número de identificaciones correctas fue de 682. ¿Respalda este resultado la hipótesis de que la proporción de identificaciones correctas sea de un 60%?

1. **Hipótesis nula:** $p=0,6$ (Proporción de identificaciones correctas)
Hipótesis alternativa: $p>0,6$.
2. Nivel de significancia 0,05
3. Criterio: Se rechaza la hipótesis nula si $p\text{-value}<0,05$, con el estadístico X^2
4. Cálculos (R Studio)

```
prop.test(x=682,n=840,p=0.6)
# data: 682 out of 840, null probability 0.6
# X-squared = 156.281, df = 1, p-value < 2.2e-16
```

6. Decisión: Dado que p-value es mayor que 0,05, **la hipótesis nula se rechaza**, es decir que la proporción de identificaciones correctas es mayor a un 60% con un nivel de confianza de 95%.

Un dato adicional que nos brinda este análisis es la proporción de positivos que es de 0,8119048, o dicho de otra forma, el porcentaje de identificaciones correctas en un rango de 16 a 29cm es de 81,19048%.

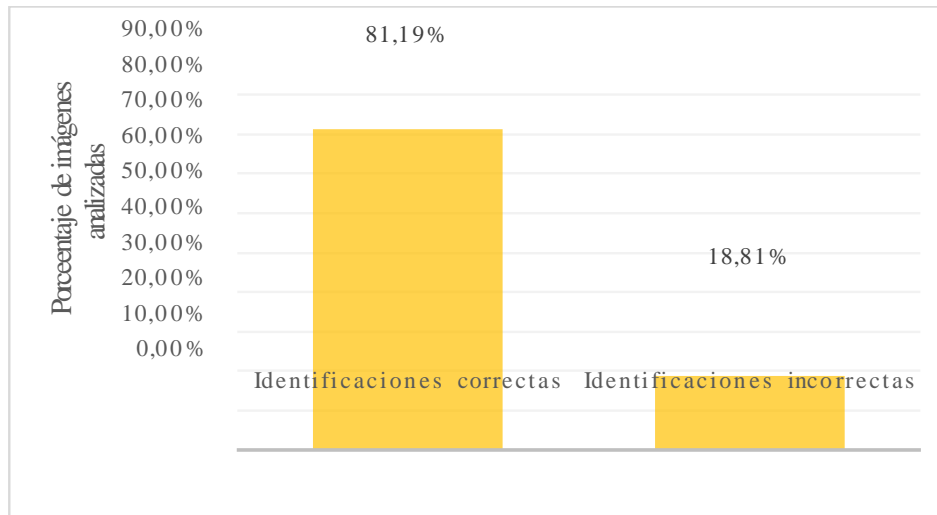


Gráfico 10-3: Porcentaje de identificaciones correctas e incorrectas de 16 a 29cm

Realizado por: Solano, G. 2018

En el gráfico 10-3 se muestra el porcentaje de identificaciones correctas, 81,19% que es mucho mayor a la de identificaciones incorrectas con un 18,81%.

Entonces el estudio de contraste sobre una proporción realizado ha determinado que el dispositivo tiene un buen desempeño de 16 a 29cm de distancia entre el billete y el mismo, habiendo un 81,19% de identificaciones correctas de un total de 840 imágenes, además este análisis cuenta con un nivel de confianza del 95%.

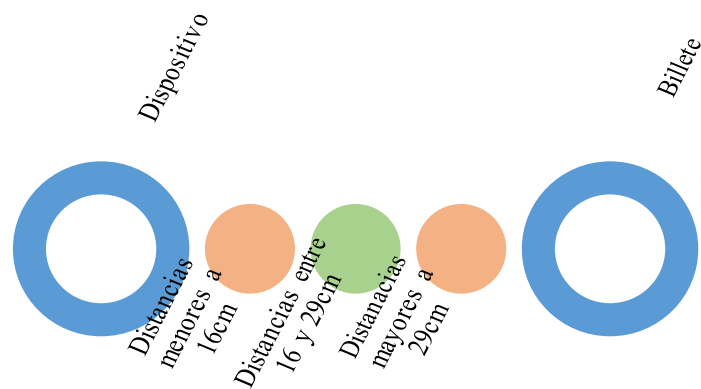


Figura 2-3: Funcionamiento de las distancias entre el dispositivo y el billete a analizar
Realizado por: Solano, G. 2018

La figura 2-3, muestra en resumen las distancias de sí y no funcionamiento o buenos resultados entre el dispositivo y el billete que desea analizar. En este caso a distancias menores a 16cm así como a distancias mayores a 30cm no se obtendrán buenos resultados, mientras que el intervalo de buen funcionamiento del dispositivo está entre 16 y 29cm.

Si se analizan las identificaciones correctas e incorrectas del intervalo de funcionamiento óptimo (entre 16 y 29cm), se puede apreciar que las detecciones dan mejor resultado ante fondos no desordenados o con patrones no recurrentes. La presencia de dichos fondos en la imagen puede acarrear malas detecciones, pues al aplicar el detector de bordes canny estos pueden interpretarse como billetes en zonas donde no los hay.

La iluminación también es un factor importante ya que si se tiene un exceso de luz sobre los billetes no se obtiene una buena identificación puesto que las características de los mismos no se logran apreciar, esto también ocurre con poca iluminación. Este problema radica especialmente en el uso del template matching, pues el método de por sí es sensible ante este factor, mientras que para la segmentación del billete por el descriptor utilizado se obtienen mejores resultados.

Otra problema que se evidenció es que al momento de determinar la denominación del billete de una imagen segmentada correctamente se podría tener fallos. Esto especialmente en imágenes posteriores entre 10 y 20 dólares, el problema radica en la similitud del diseño de los billetes, entonces al momento de identificar estos se presenta cierta confusión para el sistema.

A pesar de usar distintas plantillas para la diferenciación entre billetes de 10 y 20 dólares posteriores se seguía teniendo el mismo error. Por lo que se tomaron las plantillas que menos error se obtiene. Cabe destacar que este problema se da mayormente es estos billetes en la parte posterior, por otro lado en billetes en el lado frontal se tiene una mejor diferenciación por sus diseños en cuanto a los retratos.

3.2 Tiempo de respuesta

Una vez obtenido el rango con mejores resultados en la identificación de las denominaciones de los billetes, se procede a extraer el tiempo invertido de dicho procesamiento. Las imágenes usadas para este análisis han sido tomadas aleatoriamente de las identificaciones correctas de las pruebas de funcionamiento planteadas anteriormente.

Se tomaron 32 datos en total, 4 imágenes con: 1 dólar frontal, 1 dólar posterior, 5 dólares frontal, 5 dólares posterior, 10 dólares frontal, 10 dólares posterior, 20 dólares frontal y 20 dólares posterior. A continuación se muestra la medición de tiempo en segundos:

Tabla 4-3: Tiempo invertido para la identificación de un billete

Cantidad	Billetes	Tiempo invertido (segundos)
1	1 Frontal	141,200
2	1 Frontal	151,615
3	1 Frontal	151,486
4	1 Frontal	140,051
5	1 Posterior	138,871
6	1 Posterior	150,865
7	1 Posterior	150,328
8	1 Posterior	134,276
9	5 Frontal	140,168
10	5 Frontal	151,978
11	5 Frontal	139,647
12	5 Frontal	151,808
13	5 Posterior	140,264
14	5 Posterior	151,810
15	5 Posterior	138,711
16	5 Posterior	148,800
17	10 Frontal	151,067

Tabla 4-3 (Continua): Tiempo invertido para la identificación de un billete

18	10 Frontal	136,050
19	10 Frontal	151,633
20	10 Frontal	139,857
21	10 Posterior	151,438
22	10 Posterior	151,512
23	10 Posterior	140,187
24	10 Posterior	139,278
25	20 Frontal	139,685
26	20 Frontal	146,957
27	20 Frontal	139,958
28	20 Frontal	136,557
29	20 Posterior	139,847
30	20 Posterior	149,301
31	20 Posterior	140,051
32	20 Posterior	151,804
Promedio		144,596

Realizado por: Solano, G. 2018

Los valores en segundos, obtenidos de las 32 imágenes con billetes de distintas denominaciones dieron como resultado un promedio de 144,596 segundos o 2,41 minutos. Se representan los datos de la tabla 4-3, en el siguiente gráfico:

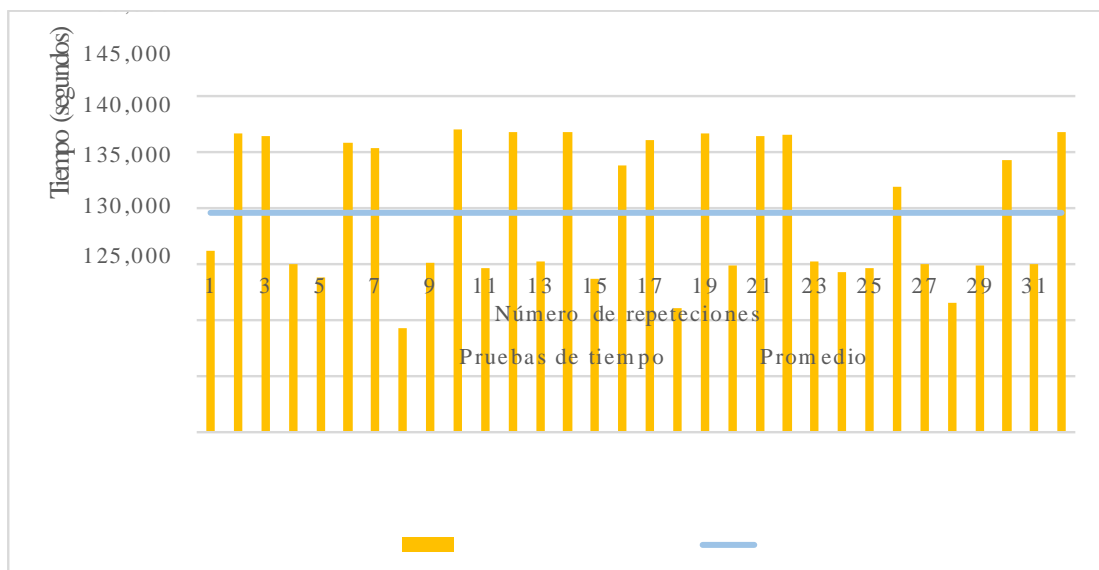


Gráfico 11-3: Pruebas de tiempo de respuesta y promedio en segundos

Realizado por: Solano, G. 2018

En el gráfico 11-3 se pueden apreciar las pequeñas variaciones de tiempo en segundos de cada identificación, así como el promedio de tiempo empleado. Se midió desde el momento en que se captura la imagen a analizar, se emite el resultado, y se da la opción de nueva captura.

3.3 Consumo de energía del dispositivo

La fuente de energía que se escogió como alimentación del dispositivo detector de billetes es una batería externa de litio-ion de 3000mAh, que además cuenta con 2 celdas. La razón por la cual se tomó esta a diferencia de otras es debido al espacio que ocupa, siendo más pequeña en comparación a otras baterías con mayor mili amperios hora.

Este tipo de baterías externas entregan 5V y 1A constantemente, sin embargo la apreciación de 3000mAh es nominal, ya que se refiere a la capacidad de la batería interna, y no toma en cuenta las pérdidas dadas por el circuito que contienen estas baterías o por los materiales con las que están fabricadas, por lo que para determinar el consumo de energía se consideró un rendimiento del 80%. (Top batería externa, 2016)

$$\text{Capacidad real} = \text{Capacidad de power Bank(mAh)} \times \text{Rendimiento} \quad \text{Ecuación 1.3}$$

$$\text{Capacidad real} = 3000\text{mAh} \times 80\% = 2400\text{mAh}$$

3.3.1 Consumo de energía del dispositivo en funcionamiento

. El tiempo que toma el dispositivo en detectar la denominación de un billete es de 2,41 minutos, que en horas son 0,04017 horas. Para determinar la corriente absorbida por el dispositivo se debe sumar la de cada uno sus los elementos.

Tabla 5-3: Consumo de corriente del dispositivo en funcionamiento

Componente	Corriente (mA)
Amplificador de audio	4
Altavoz	50
Cámara	45
Raspberry pi 3 B	360
TOTAL	459

Realizado por: Solano, G. 2018

$$\text{Capacidad (identificación)} = \text{Consumo de corriente} \times \text{tiempo} \quad \text{Ecuación 2.3}$$

$$\text{Capacidad (identificación)} = 459\text{mA} \times 0,04017\text{h}$$

$$\text{Capacidad (identificación)} = 18,438\text{mAh}$$

Para poder calcular la energía consumida por el dispositivo en funcionamiento se aplica:

$$\text{Energía (identificación)} = \text{Voltaje} \times \text{Capacidad (Identificación)} \quad \text{Ecuación 3.3}$$

$$\text{Energía consumida por cada identificación} = 5\text{V} \times 18,438\text{mAh}$$

$$\text{Energía consumida por cada identificación} = 92,19\text{mWh}$$

3.3.2 Consumo de energía del dispositivo en reposo

Para determinar la corriente absorbida por el dispositivo en reposo o sin funcionamiento, se debe tomar en cuenta que únicamente entra en funcionamiento la placa raspberry pi puesto que los otros circuitos no actúan.

Tabla 6-3: Consumo de corriente del dispositivo en reposo

Componente	Corriente (mA)
Raspberry pi 3 B	230
TOTAL	230

Realizado por: Solano, G. 2018

Para calcular la capacidad del dispositivo en reposo, se tomó el mismo tiempo de identificación del billete es decir 0,04017 horas, por lo tanto se aplica la ecuación 2.3:

$$\text{Capacidad (en reposo)} = 230\text{mA} \times 0,04017\text{h}$$

$$\text{Capacidad (en reposo)} = 9,2391\text{mAh}$$

Para poder calcular la energía consumida por el dispositivo en reposo se aplica la ecuación 3.3:

$$\text{Energía (en reposo)} = \text{Voltaje} \times \text{Capacidad (en reposo)}$$

$$\text{Energía consumida en reposo} = 5V \times 9,2391mAh$$

$$\text{Energía consumida en reposo} = 46,1955mWh$$

3.3.3 Tiempo de uso de la batería

El primer paso para determinar el tiempo de desempeño de la batería es calcular la energía de la batería dada por la ecuación 3.3:

$$\text{Energía (batería)} = \text{Voltaje} \times \text{Capacidad (batería)}$$

$$\text{Energía de la batería} = 5V \times 2,400mAh$$

$$\text{Energía de la batería} = 12000mWh$$

Entonces se pueden determinar el número de detecciones que se pueden realizar:

$$\text{Número de detecciones} = \frac{\text{Energía de la batería}}{\text{Energía consumida por cada identificación}} \quad \text{Ecuación 4.3}$$

$$\text{Número de detecciones} = \frac{12000mWh}{92,19mWh}$$

$$\text{Número de detecciones} = 130$$

Ahora se puede calcular el tiempo de desempeño de la batería, multiplicando el número de detecciones por el tiempo que le toma realizar cada una.

$$\text{Tiempo de desempeño de la batería} = \text{Número de detecciones} \times \text{tiempo por cada detección} \quad \text{Ecuación 5.3}$$

$$\text{Tiempo de desempeño de la batería} = 130 \times 0,04017h$$

$$\text{Tiempo de desempeño de la batería} = 5,22h$$

Otro análisis que se puede plantear para determinar el tiempo de uso de la batería es basándose en el consumo de corriente tomando como referencia la corriente consumida por el dispositivo en funcionamiento, y la capacidad de la batería externa esto es:

Tiempo de desempeño de la batería =

$$\frac{\text{Capacidad real de la batería}}{\text{Consumo de corriente por cada identificación}}$$

Ecuación 6.3

$$\text{T tiempo de desempeño de la batería} = \frac{2400mAh}{459mA}$$

T tiempo de desempeño de la batería = 5,22h

Se obtuvo un total de 5 horas con 22 minutos aproximadamente de desempeño de la batería.

3.4 Análisis de costos

A continuación se presentan los costos necesarios para la construcción del dispositivo presentado en este trabajo de titulación.

3.4.1 Costos de materiales y herramientas

Los materiales y herramientas para llevar a cabo la construcción del dispositivo detector de billetes son:

Tabla 7-3: Costos de materiales y herramientas

Cantidad	Detalle	Valor unitario	Total
1	Resistencia	\$0,05	\$0,05
1	Capacitor	\$0,15	\$0,15
1	Circuito integrado LM386	\$0,40	\$0,40
1	Pulsador	\$0,30	\$0,30
1	Altavoz de 0,25W	\$0,75	\$0,75
1	Potenciómetro de precisión	\$0,15	\$0,15
1	Cable de timbre	\$0,25	\$0,25
5	Cables macho-hembra	\$0,10	\$0,50
1	Placa perforada	\$0,30	\$0,30
1	Switch	\$0,30	\$0,30
1	Multímetro	\$10,00	\$10,00

Tabla 7-3 (Continúa): Costos de materiales y herramientas

1	Flex cable para cámara Raspberry	\$11,00	\$11,00
1	Estaño	\$0,25	\$0,25
1	Cautín	\$3,75	\$3,75
1	Impresión 3D	\$42,00	\$42,00
TOTAL			\$69,85

Realizado por: Solano, G. 2018

3.4.2 Costos de equipos

Los equipos usados para la construcción del dispositivo detector de billetes son los siguientes:

Tabla 8-3: Costos de equipos

Cantidad	Detalle	Valor unitario	Total
1	Raspberry Pi 3	\$67,50	\$67,50
1	Pi Camera v2	\$30,00	\$30,00
1	Router	\$20,00	\$20,00
1	Teclado	\$15,00	\$15,00
1	Conversor HDMI-VGA	\$15,00	\$15,00
TOTAL			\$147,5

Realizado por: Solano, G. 2018

3.4.3 Costos de implementación

En la siguiente tabla se describen los costos de implementación del dispositivo:

Tabla 9-3: Costos de implementación

Cantidad	Detalle	Tiempo (Horas)	Valor unitario	Total
1	Desarrollo de software	260	\$6,00	\$1.560,00
1	Ensamblaje del dispositivo	4	\$4,00	\$16,00
TOTAL				\$1.576,00

Realizado por: Solano, G. 2018

3.4.4 Costo total de construcción del dispositivo

El costo total del dispositivo se calcula con la suma de los costos previamente expuestos.

Tabla 10-3: Costo total

Descripción	Costos
Costos de materiales y herramientas	\$69,85
Costos de equipos	\$147,5
Costos de implementación	\$1.576,00
TOTAL	\$1.793,35

Realizado por: Solano, G. 2018

El costo total para la construcción del dispositivo tomando en cuenta los costos de materiales y herramientas, equipos e implementación fue de \$1.793,35 dólares.

CONCLUSIONES

- Se realizó el diseño y construcción de un dispositivo portátil capaz de identificar las denominaciones de los billetes de 1, 5, 10 y 20 dólares, en su forma frontal y posterior.
- Mediante el uso de la variación del descriptor Local Binary Pattern se logró invariabilidad de los billetes ante la rotación, haciendo de la misma una buena herramienta para la detección de los mismos con distintas orientaciones, además disminuye un vector de 256 a 59 elementos que es beneficioso para el tiempo de procesamiento.
- Se logró mediante la utilización de un clasificador lineal de regresión logística separar un billete del resto de una imagen, esto a través de un modelo obtenido luego de un entrenamiento de varias imágenes en donde se escoge el modelo con menor coste.
- El template matching utilizado para diferenciar las denominaciones de los billetes es sensible ante cambios de iluminación y por ello puede que no se diferencie correctamente un billete aunque esté bien segmentado de la imagen.
- Debido al gran parecido que tienen las partes posteriores de los billetes de 10 y 20 dólares puede haber confusiones a la hora de identificarlos dando como resultado errores al emitir sus denominaciones.
- La iluminación es un factor importante en el desempeño del dispositivo, a muy poca o en exceso puede causar el mal funcionamiento, es decir que puede detectar falsos positivos
- Se concluye que el rango óptimo de distancia para un buen desempeño del dispositivo está entre 16 y 29cm entre el billete a analizar y el dispositivo, además el tiempo que le toma realizar el procesamiento desde que se captura la imagen del billete hasta que emite su denominación es de 2,41 minutos.
- La energía consumida por el dispositivo en funcionamiento en 2,41 minutos es de 92,19mWh, mientras que en reposo es de 46,1955mWh. Tomando el consumo de energía en funcionamiento y la energía de la batería, se ha determinado que la batería tiene aproximadamente un tiempo de uso de 5 horas con 22 minutos.

RECOMENDACIONES

- La variación del descriptor utilizado brinda una gran ventaja al disminuir el número de elementos de una imagen de intensidad a Local Binary Pattern invariante a la rotación, sin embargo este puede significar también que pueden detectarse billetes en lugares de la imagen donde no las hay, por lo que se recomienda el uso de una combinación de descriptores Local Binary Pattern y/o aplicarlos por bloques para agregarle robustez a la identificación.
- Se recomienda utilizar otro tipo de clasificadores tales como: Análisis de Componentes Principales o redes neuronales, para determinar la denominación del billete en la clasificación misma con el fondo.
- Se recomienda utilizar varias plantillas para identificar un mismo billete, ya que de este modo, reconociendo todas estas en conjunto se determinará su denominación mejorando la detección, sin embargo puede acarrear más tiempo de procesamiento.
- Se recomienda la implementación de una fuente lumínica en el dispositivo pues de este modo se discriminaría mejor el fondo del billete y se lograría una mejor detección de este y su denominación.
- Se recomienda buscar alternativas software y/o hardware que permitan disminuir el tiempo de procesamiento del algoritmo de identificación de billetes.
- En la construcción del dispositivo debe tomarse muy en cuenta la manipulación de los elementos electrónicos dado que pueden sufrir averías en sus componentes ante electricidad estática.
- Se recomienda que para una mejor detección se trate de aplanar lo que más se pueda al billete y no doblarlos mientras se realiza la captura para obtener mejores resultados de identificación.
- Se recomienda adicionar la autenticación de billetes de dólar a más de su denominación, puesto que este trabajo de titulación solamente se basó en el segundo aspecto.

BIBLIOGRAFÍA

CONSEJO NACIONAL PARA LA IGUALDAD DE DISCAPACIDADES. *Información Estadística de personas con discapacidad* [en línea]. Febrero 2017. [Consulta: 1 junio del 2017]. Disponible en: <http://www.consejodiscapacidades.gob.ec/estadistica/index.html>

DEPARTAMENTO DE ELECTRÓNICA U.A.H. “*Fases de un sistema de V.A*” [en línea]. [Consulta: 10 junio del 2017]. Disponible en: <ftp://ftp.unicauca.edu.co/Facultades/FIET/DEIC/Materias/Vision%20de%20Maquina/Documentos/Etapas%20de%20un%20SVA.pdf>

DIRECCIÓN NACIONAL DE DISCAPACIDADES. *Discapacidad* [en línea]. [Consulta: 7 de junio del 2017]. Disponible en: <http://www.salud.gob.ec/direccion-nacional-de-discapacidades/>

GARCÍA-OLALLA, Oscar & ALEGRE Enrique. “*Descripción de textura en imágenes utilizando Local Binary Pattern (LBP)*” [en línea]. Universidad de León, Departamento de Ingeniería eléctrica y de Sistemas y Automática, Escuela de Ingenierías. España. pp. 115-127. [Consulta: 20 junio del 2017]. Disponible: [http://pitia.unileon.es/varp/sites/default/files/PublicationPDF/Descripcion%20de%20textura%20en%20imagenes%20utilizando%20Local%20Binary%20Pattern%20\(LBP\).pdf](http://pitia.unileon.es/varp/sites/default/files/PublicationPDF/Descripcion%20de%20textura%20en%20imagenes%20utilizando%20Local%20Binary%20Pattern%20(LBP).pdf)

HASSANUZZAMAN, Faíz M. et al. “*Robust and Effective Component-based Banknote Recognition for the Blind*”. IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews) [en línea], 2012, (United State of America) 42(6), pp. 1021-1030. [Consulta: 22 junio del 2017]. Número ISSN 0018-9472. Disponible en: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3363999/>

INSTITUTO NACIONAL DE ESTADÍSTICAS Y CENSOS. *Información Censal Cantonal* [en línea]. [Consulta: 1 de junio del 2017]. Disponible en: http://www.inec.gob.ec/cpv/index.php?option=com_content&view=article&id=232&Itemid=128&lang=es

LANCHEROS RUIZ, Carolina. *Crean detector sonoro de billetes para ciegos en Universidad del Bosque* [en línea]. (2008). (Colombia). [Consulta: 5 de junio del 2017]. Disponible en: <http://www.eltiempo.com/archivo/documento/CMS-4325124>

ORGANIZACIÓN MUNDIAL DE LA SALUD. *Ceguera y discapacidad visual* [en línea]. Agosto 2014. [Consulta: 1 junio 2017]. Disponible en: <http://www.who.int/mediacentre/factsheets/fs282/es/>

PÉREZ PORTO, Julián & MERINO, María. *Definición de discapacidad* [en línea]. 2012. [Consulta: 7 junio 2017]. Disponible en: <http://definicion.de/discapacidad/>

PUTTEMANS, Steven. *Cascade Classifier Training*. [En línea]. [Consulta: 20 noviembre del 2017]. Disponible en: https://docs.opencv.org/3.4.1/dc/d88/tutorial_traincascade.html

SÁNCHEZ LÓPEZ, Laura. *“Local Binary Patterns applied to Face Detection and Recognition”* [en línea]. Universitat Politècnica de Catalunya, Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona, España 2010. pp. 7-11. [Consulta: 10 junio 2017]. Disponible: [https://upcommons.upc.edu/bitstream/handle/2099.1/10772/PFC_LauraSanchez_\(LBP_applied_to_FaceDetection&Recognition\).pdf](https://upcommons.upc.edu/bitstream/handle/2099.1/10772/PFC_LauraSanchez_(LBP_applied_to_FaceDetection&Recognition).pdf)

SECRETARÍA TÉCNICA PLAN TODA UNA VIDA. *Programas y servicios* [en línea]. [Consulta: 12 mayo 2018]. Disponible en: <https://www.todaunavida.gob.ec/category/programas-y-servicios/>

TERVEN, Juan. et al. *“Estado del Arte en Sistemas de Visión Artificial para Personas Invidentes”*. Komputer Sapiens [en línea], 2013, (México) volumen I, pp. 20-30. [Consulta: 20 julio del 2017]. ISSN 2007-0691. Disponible en: <http://refbase.cvc.uab.es/files/TSR2013a.pdf>

TOP BATERÍA EXTERNA. *Conocer capacidad real de una batería externa* [en línea], 2016. [Consulta: 14 junio del 2018]. Disponible en: <https://topbateriaexterna.com/capacidad-real-de-una-bateria-externa/>

VARGAS, Víctor. *Sistema de Visión Artificial para el Control de Calidad en Piezas Cromadas* (Tesis). (Maestría). Instituto Politécnico Nacional, México 2010. pp. 21-27. [Consulta: 7 junio 2017]. Disponible en: <http://tesis.ipn.mx/bitstream/handle/123456789/7250/68.pdf?sequence=1>

VILLALBA, Mateo. *Carta a diario el universo a propósito del artículo "sin billetes de usd 50 y usd 100"*. [En línea]. Quito, 2015 [Consulta: 1 junio del 2017]. Disponible en: [https://www.bce.fin.ec/index.php/boletines-de-prensa-archivo/item/795-carta-a-diario-eluniverso-a-prop%](https://www.bce.fin.ec/index.php/boletines-de-prensa-archivo/item/795-carta-a-diario-eluniverso-a-prop%C3%B3)

sito-del-art%C3%ADculo-sin-billetes-de-usd-50-y-usd-100

FERRARONS, Miquel & VAZQUEZ, David. *Detector de objetos LBP HOG*. [En línea]. España, Barcelona 2015. [Consulta: 25 noviembre del 2017]. Disponible en: <https://www.course-ra.org/learn/deteccion-objetos/suplement/muzCm/codigo-ejemplo-de-detector>

ANEXOS

ANEXO A: Programas desarrollados para la identificación de billetes y sus denominaciones

Programa en Python: Descriptor LBP invariante a la rotación

```
import numpy as np
import math
from math import floor, sqrt
import cv2
import os
import pickle
import Information_ as Inf
import LR
import imutils
import time
#####
##### DESCRIPTOR LBP #####
#####

#Descriptor LBPU invariante a la rotacion
def lbpUInvRotate(im):#Este es invariante ante la rotacion
    lbpu=[0, 1, 2, 3, 4, 6, 7, 8, 12, 14, 15, 16, 24, 28, 30, 31, 32, 48,
          56, 60, 62, 63, 64, 96, 112, 120, 124, 126, 127, 128, 129, 131,
          135, 143, 159, 191, 192, 193, 195, 199, 207, 223, 224, 225, 227,
          231, 239, 240, 241, 243, 247, 248, 249, 251, 252, 253, 254, 255]
    a=np.array([0,0,0,0,0,0,0,0])#Vector binario de acuerdo a los vecinos
    Sum=0 #Sumatorio para obtener el valor total de 0 a 255
    vSum=[]
    T=8
    imLBPU=np.zeros(shape=(im.shape[0],im.shape[1]))
    for i in range(im.shape[0]):
        for j in range(im.shape[1]):
            if (i>0 and j>0)and((i<im.shape[0]-1) and j<(im.shape[1]-1)):
                a[0]=1 if im[i,j-1]-im[i,j]>=T else 0
                a[1]=1 if im[i+1,j-1]-im[i,j]>=T else 0
                a[2]=1 if im[i+1,j]-im[i,j]>=T else 0
                a[3]=1 if im[i+1,j+1]-im[i,j]>=T else 0
                a[4]=1 if im[i,j+1]-im[i,j]>=T else 0
                a[5]=1 if im[i-1,j+1]-im[i,j]>=T else 0
                a[6]=1 if im[i-1,j]-im[i,j]>=T else 0
                a[7]=1 if im[i-1,j-1]-im[i,j]>=T else 0
                aux=a
                for k in range (8):
                    for x in range (0,8): #Se determina el valor total del pixel
                        Sum=Sum+(aux[x])*(2**x)

                    vSum.append(Sum)
                    aux=recorrer(aux)
                    Sum=0
                minimo=min(vSum)
                imLBPU[i,j]=(lbpu.index(minimo)+1) if minimo in lbpu else 59
                Sum=0
                vSum=[]
            else: imLBPU[i,j]=(1)
    return imLBPU##, vimLBPU

def recorrer(vec): #Funcion auxiliar para el LBP invariante ante la rotacion
    vNew=[]
    for i in range (8):
        vNew.append(vec[0]) if (i==7) else vNew.append(vec[i+1])
        #else: vNew.append(vec[i+1])
    return vNew
```

Regresión Logística

```
import numpy as np

def sigmoid_learn(xj, w, c):
    xw=0
    for i in range(len(xj)):
        if c==i: w[i]=1
        xw=xw+xj[i]*w[i]
    return 1/float(1+np.exp(-xw))

def sigmoid(xj, w):
    xw=0
    for i in range(len(xj)):
        xw=xw+xj[i]*w[i]
    return 1/float(1+np.exp(-xw))

def dev_fun_cost(M, h, xjj, yj):
    aux=1/float(M)
    devFun=aux*((h-yj)*xjj)
    return devFun

def learning_model(M, xj, yj): #Se aprenden los modelos
    w=np.ones(len(xj), dtype=float)
    model=np.zeros(len(xj), dtype=float)
    alfa=0.5
    for i in range(len(xj)):
        if i==0:
            h=sigmoid_learn(xj, w, i)
            w[i]=w[0]-alfa*dev_fun_cost(M, h, xj[i], yj)
        else:
            h=sigmoid_learn(xj, w, i)
            w[i]=w[i-1]-alfa*dev_fun_cost(M, h, xj[i], yj)
    return w

def fun_cost(M, Xj, y):
    w=np.zeros(M, dtype=object)
    J=np.zeros(M, dtype=float)
    for i in range(M):
        Xj[i]=np.append([1], Xj[i])
        w[i]=learning_model(M, Xj[i], y[i])
        hw=sigmoid(Xj[i], w[i])
        J[i]=y[i]*np.log(hw)+(1-y[i])*np.log(1-hw)
```

```

        J[i]=-J[i]/M
    J=np.abs(J)
    JW=np.sort(J)
    JWarg=np.argsort(J)
    print 'JW=', JW
    print 'JWarg=', JWarg
    return J.min(),w[J.argmax()]

def classify(xj,w,Tmin,Tmax):
    hw=sigmoid(xj,w)
    xw=0
    if (hw>Tmin)and(hw<=Tmax):Billete=1
    else: Billete=0
    return Billete,hw

def classifyFT(xj,w,Tmin1,Tmin2,Tmax1,Tmax2):
    hw=sigmoid(xj,w)
    if ((hw>Tmin1)and(hw<Tmin2))or((hw>Tmax1)and(hw<Tmax2)): Frontal=1
    else: Frontal=0
    return Frontal,hw

```

Extracción de características

```

from PIL import Image
import numpy as np
import os
import pickle

import Information_ as Inf
import Tools as Tl

def extractFeatures(im):

    winW,winH=Inf.sW #Valor de la ventana deslizante
    imlbpU, vecLBPU=Tl.lbpUInvRotate(im)
    features=[]
    count=0
    for (x, y, window) in Tl.slidingWindow(imlbpU, stepSize=Inf.stepW, windowSize=(winW, winH)):
        lbpu_hist,bin_edges=np.histogram(vecLBPU,bins=59)
        lbpu_hist_norm=sum(np.abs(lbpu_hist))
        lbp_hist_llsqtrnorm = np.sqrt(lbpu_hist/float(lbpu_hist_norm))
        features.append(lbp_hist_llsqtrnorm)
    #Aplanar una lista de listas
    features_flatten = [item for sublist in features for item in sublist]
    return features

def storeFeatures(inputF,outputF):

    fileList=os.listdir(inputF)
    #Solo elegir imagenes .png
    imgList=filter(lambda element: '.png' in element, fileList)

    for fileName in imgList:
        imgPath=inputF+ '/' + fileName
        outPath=outputF+ '/' +fileName+'.feat'
        if os.path.exists(outPath):
            print 'Features for ' + imgPath + '. Delete the file if you want to replace.'
            continue
        print 'Extracting features for ' + imgPath

        img=Image.open(imgPath)
        img=np.array(img.getdata(),np.uint8).reshape((img.size[1],img.size[0],3))
        img=Tl.grayScale(img)

```

```
#Extraccion de características
feats=extractFeatures(img)

#Almacenar las características en un archivo
outputFile = open(outPath, "wb")
pickle.dump(feats, outputFile)
outputFile.close()

def run():

    if not os.path.exists(Inf.positiveFS):
        os.makedirs(Inf.positiveFS)
    if not os.path.exists(Inf.negativeFS):
        os.makedirs(Inf.negativeFS)
    #Extract features for positive samples
    print 'Extracting features from images in '+Inf.positive_f
    storeFeatures(Inf.positive_f, Inf.positiveFS)

    print 'Extracting features from images in '+Inf.negative_f
    #Extract features for negative samples
    storeFeatures(Inf.negative_f, Inf.negativeFS)
```

Entrenamiento del clasificador

```
import numpy as np
import os
import pickle
import Information_ as Inf
import LR

def loadFeatures(features):
    print 'Cargando características de'+str(features)
    file=open(features,'r')
    return pickle.load(file)

def run():
    #Se cargan los archivos positivos anteriormente almacenados
    positiveL=os.listdir(Inf.positiveFS)
    positiveL=filter(lambda element: '.feat' in element, positiveL)

    #Se cargan los archivos negativos o fondo anteriormente almacenados
    negativeL = os.listdir(Inf.negativeFS)
    negativeL = filter(lambda element: '.feat' in element, negativeL)

    #Se cuenta el numero de muestras
    PositiveCount=len(positiveL)
    NegativeCount=len(negativeL)
    CountPN=PositiveCount +NegativeCount

    #Se procede a inicializar la estructura necesaria para entrenar
    # X seran los patrones y fondo a entrenar
    # y seran las etiquetas, 0 o 1
    X = np.zeros(shape=(CountPN), dtype=object)
    y = np.append(np.ones(shape=(1, PositiveCount)),
                  0*np.ones(shape=(1, NegativeCount)))

    #Se cargan todos las caracteristicas postivas en X
    count = 0
    AUX=[]
    for filename in positiveL:
        filepath = Inf.positiveFS+'/'+filename
        AUX = loadFeatures(filepath)
        X[count]=AUX[0]
        count += 1

    #Se cargan todos las caracteristicas negativos en y
    for filename in negativeL:
        filepath = Inf.negativeFS+'/'+filename
        AUX = loadFeatures(filepath)
        X[count]=AUX[0]
        count += 1

    model=np.zeros(60,dtype=object)

    J,model=LR.fun_cost(CountPN,X,y)
    print 'J de entrenamiento=',J
    print 'Modelo obtenido=',model
    print 'len(model)=',len(model)

    #Guardar el modelo
    modelFolder, modelFilename=os.path.split(Inf.modelPath)
    if not os.path.exists(modelFolder):
        os.makedirs(modelFolder)

    outputModelFile = open(Inf.modelPath, 'wb')
    pickle.dump(model, outputModelFile)
```


Clasificación billete vs fondo

```
##### CLASIFICACION BILLETE VS FONDO #####
#####
def searchBill(image):
    filepath = Inf.pathModel
    file = open(filepath, 'r')
    model = pickle.load(file)
    imageG=grayScale(image)
    winW,winH=Inf.sW #Se llama al tamaño de la pantalla deslizando
    deteccion=0
    resized=cv2.resize(imageG, (447,469),interpolation=cv2.INTER_AREA)#447,469
    resIM=lbpUInvRotate(resized)
    box1=[]
    hw_dolar=[]
    xi=[]
    yi=[]
    for (x, y, window) in slidingWindow(resIM, stepSize=Inf.stepW, windowSize=(winW, winH)):
        # if the window does not meet our desired window size, ignore it
        if window.shape[0] != winH or window.shape[1] != winW:
            continue
        vWIN=[]#Se declara el vector para almacenar la imagen
        for i in range(window.shape[0]): #Recorre filas
            for j in range(window.shape[1]): #Recorre columnas
                vWIN.append(window[i,j])
        lbpu_hist,bin_edges=np.histogram(vWIN,bins=59)
        lbpu_hist_norm=sum(np.abs(lbpu_hist))
        lbp_hist_llsqtrnorm = np.sqrt(lbpu_hist/float(lbpu_hist_norm))
        lbp_hist_llsqtrnorm=np.append([1],lbp_hist_llsqtrnorm)
        Tmin=0.9532 #Umbral minimo
        Tmax=0.97#0.968 #Umbral maximo
        Dolar,hw=LR.classify(lbp_hist_llsqtrnorm,model,Tmin,Tmax)
        if Dolar==1:
            hw_dolar.append(hw)
            xi.append(x)
            yi.append(y)
    if hw_dolar!=[]:
        hw_dolar=np.array(hw_dolar)
        arg_hw_dolar=np.argsort(hw_dolar)[::-1]#Ordena de forma descendente
        k=arg_hw_dolar[0]
        ht=resized.shape[0]
        wt=resized.shape[1]
        resBill=cv2.resize(image, (wt,ht),interpolation=cv2.INTER_AREA)
        cropBill=resBill[yi[k]:yi[k]+170,xi[k]:xi[k]+320]
        billete=cropBill
        deteccion=1
    if deteccion==0:
        #print '*****BILLETE NO DETECTADO*****'
        return None,deteccion
    if deteccion==1:
        #print '*****BILLETE DETECTADO*****'
        return billete,deteccion #Imagen obtenida de la clasificacion, deteccion=1
```

Determinar el mejor emparejamiento de un grupo de plantillas

```
##### IDENTIFICACION DE DENOMINACIONES #####
#####
#Devuelve el mejor emparejamiento
def temp(img, globalPath, tempFile, method, nScale, auxW, auxH):
    methodTM=eval(method)
    i=0
    t=[]
    kernel = np.ones((5,5),np.float32)/25

    for fileName in tempFile:
        pathName=globalPath+fileName
        im=cv2.imread(pathName,0)
        im = cv2.filter2D(im,-1,kernel)
        im=cv2.Canny(im, 40, 200)
        t.append(im)
    t=np.array(t)
    temp=t[0]
    temp5=t[1]
    temp10=t[2]
    temp15=t[3]
    temp_5=t[4]
    temp_10=t[5]
    temp_15=t[6]

    maxV=[]
    maxL=[]
    wt=[]
    ht=[]
    w,h=temp.shape[1],temp.shape[0]
    for i in range(nScale):
        temp=cv2.resize(temp, (w,h), interpolation=cv2.INTER_AREA)
        temp5=cv2.resize(temp5, (w,h), interpolation=cv2.INTER_AREA)
        temp10=cv2.resize(temp10, (w,h), interpolation=cv2.INTER_AREA)
        temp15=cv2.resize(temp15, (w,h), interpolation=cv2.INTER_AREA)
        temp_5=cv2.resize(temp_5, (w,h), interpolation=cv2.INTER_AREA)
        temp_10=cv2.resize(temp_10, (w,h), interpolation=cv2.INTER_AREA)
        temp_15=cv2.resize(temp_15, (w,h), interpolation=cv2.INTER_AREA)

        result = cv2.matchTemplate(img, temp, methodTM)
        (_, maxVal, _, maxLoc) = cv2.minMaxLoc(result)
        maxV.append(maxVal/(w*h))
        maxL.append(maxLoc)
        wt.append(w)
        ht.append(h)

        result5 = cv2.matchTemplate(img, temp5, methodTM)
        (_, maxVal5, _, maxLoc5) = cv2.minMaxLoc(result5)
        maxV.append(maxVal5/(w*h))
        maxL.append(maxLoc5)
        wt.append(w)
        ht.append(h)

        result10 = cv2.matchTemplate(img, temp10, methodTM)
        (_, maxVal10, _, maxLoc10) = cv2.minMaxLoc(result10)
        maxV.append(maxVal10/(w*h))
        maxL.append(maxLoc10)
        wt.append(w)
        ht.append(h)
```

```

result15 = cv2.matchTemplate(img, temp15, methodTM)
(_, maxVal15, _, maxLoc15) = cv2.minMaxLoc(result15)
maxV.append(maxVal15/(w*h))
maxL.append(maxLoc15)
wt.append(w)
ht.append(h)

result_5 = cv2.matchTemplate(img, temp_5, methodTM)
(_, maxVal_5, _, maxLoc_5) = cv2.minMaxLoc(result_5)
maxV.append(maxVal_5/(w*h))
maxL.append(maxLoc_5)
wt.append(w)
ht.append(h)

result_10 = cv2.matchTemplate(img, temp_10, methodTM)
(_, maxVal_10, _, maxLoc_10) = cv2.minMaxLoc(result_10)
maxV.append(maxVal_10/(w*h))
maxL.append(maxLoc_10)
wt.append(w)
ht.append(h)

result_15 = cv2.matchTemplate(img, temp_15, methodTM)
(_, maxVal_15, _, maxLoc_15) = cv2.minMaxLoc(result_15)
maxV.append(maxVal_15/(w*h))
maxL.append(maxLoc_15)
wt.append(w)
ht.append(h)

w=w-auxW
h=h-auxH
wt=np.array(wt)
ht=np.array(ht)
maxV=np.array(maxV)
maxL=np.array(maxL)
argV=np.argsort(maxV)[::-1]
amaxV=maxV[argV[0]]
amaxL=maxL[argV[0]]
aw=wt[argV[0]]
ah=ht[argV[0]]
return amaxV,amaxL,aw,ah

```

Determinación de la denominación

```
-  
def Denomin(image):  
    img= cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)  
    img=cv2.equalizeHist(img)  
    kernel = np.ones((5,5),np.float32)/25  
    img = cv2.filter2D(img,-1,kernel)  
    img=cv2.Canny(img, 40, 200)  
    globalPath1F=Inf.path1F  
    tempFile1F=Inf.files1F  
    maxV1F,maxL1F,aw1F,ah1F=temp(img,globalPath1F,tempFile1F,'cv2.TM_CCOEFF',16,5,5)  
  
    globalPath5F=Inf.path5F  
    tempFile5F=Inf.files5F  
    maxV5F,maxL5F,aw5F,ah5F=temp(img,globalPath5F,tempFile5F,'cv2.TM_CCOEFF',14,5,5)  
  
    globalPath10F=Inf.path10F  
    tempFile10F=Inf.files10F  
    maxV10F,maxL10F,aw10F,ah10F=temp(img,globalPath10F,tempFile10F,'cv2.TM_CCOEFF',11,5,5)  
  
    globalPath20F=Inf.path20F  
    tempFile20F=Inf.files20F  
    maxV20F,maxL20F,aw20F,ah20F=temp(img,globalPath20F,tempFile20F,'cv2.TM_CCOEFF',12,5,5)  
  
    globalPath1P=Inf.path1P  
    tempFile1P=Inf.files1P  
    maxV1P,maxL1P,aw1P,ah1P=temp(img,globalPath1P,tempFile1P,'cv2.TM_CCOEFF',12,5,5)  
  
    globalPath5P=Inf.path5P  
    tempFile5P=Inf.files5P  
    maxV5P,maxL5P,aw5P,ah5P=temp(img,globalPath5P,tempFile5P,'cv2.TM_CCOEFF',12,5,5)  
  
    globalPath10P=Inf.path10P  
    tempFile10P=Inf.files10P  
    maxV10P,maxL10P,aw10P,ah10P=temp(img,globalPath10P,tempFile10P,'cv2.TM_CCOEFF',10,5,5)  
  
    globalPath20P=Inf.path20P  
    tempFile20P=Inf.files20P  
    maxV20P,maxL20P,aw20P,ah20P=temp(img,globalPath20P,tempFile20P,'cv2.TM_CCOEFF',11,5,5)  
    den=[]  
    if (maxV1F>455): den.append(maxV1F) #425  
    else: den.append(0)  
  
    if (maxV1P>471): den.append(maxV1P) #430  
    else: den.append(0)  
  
    if (maxV5F>365): den.append(maxV5F) #361  
    else: den.append(0)  
  
    if (maxV5P>510): den.append(maxV5P) #450  
    else: den.append(0)  
  
    if (maxV10F>445): den.append(maxV10F) #415  
    else: den.append(0)  
  
    if (maxV10P>505): den.append(maxV10P) #350  
    else: den.append(0)  
  
    if (maxV20F>360): den.append(maxV20F) #315  
    else: den.append(0)  
  
    if (maxV20P>406): den.append(maxV20P) #340  
    else: den.append(0)  
  
    den=np.array(den)  
    argDen=np.argsort(den)[::-1]  
    arg=argDen[0]  
    return den[arg],arg
```

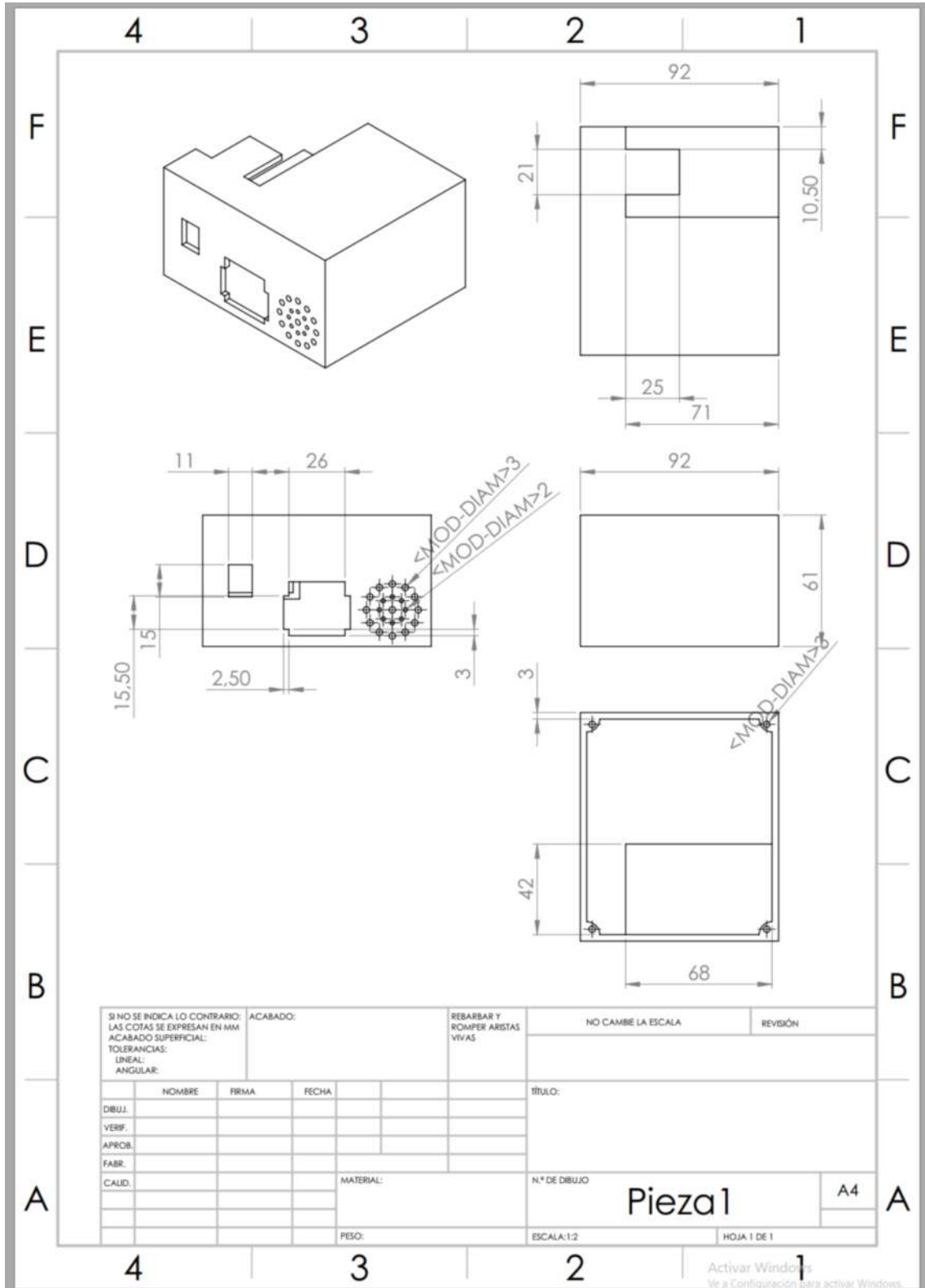
Difusión de resultados

```
def infDollar(arg):
    import pyaudio
    import wave
    path=Inf.pathAudio
    chunk = 1024
    p = pyaudio.PyAudio()
    if (arg==0) or (arg==1):
        infD='ONE DOLLAR'
        wf = wave.open(path+'Un-dolar.wav', 'rb')
    if (arg==2) or (arg==3):
        infD='FIVE DOLLARS'
        wf = wave.open(path+'Cinco-dolares.wav', 'rb')
    if (arg==4) or (arg==5):
        infD='TEN DOLLARS'
        wf = wave.open(path+'Diez-dolares.wav', 'rb')
    if (arg==6) or (arg==7):
        infD='TWENTY DOLLARS'
        wf = wave.open(path+'Veinte-dolares.wav', 'rb')
    stream = p.open(
        format = p.get_format_from_width(wf.getsampwidth()),
        channels = wf.getnchannels(),
        rate = wf.getframerate(),
        output = True)
    data = wf.readframes(chunk)
    while data != '':
        stream.write(data)
        data = wf.readframes(chunk)
    stream.close()
    p.terminate()

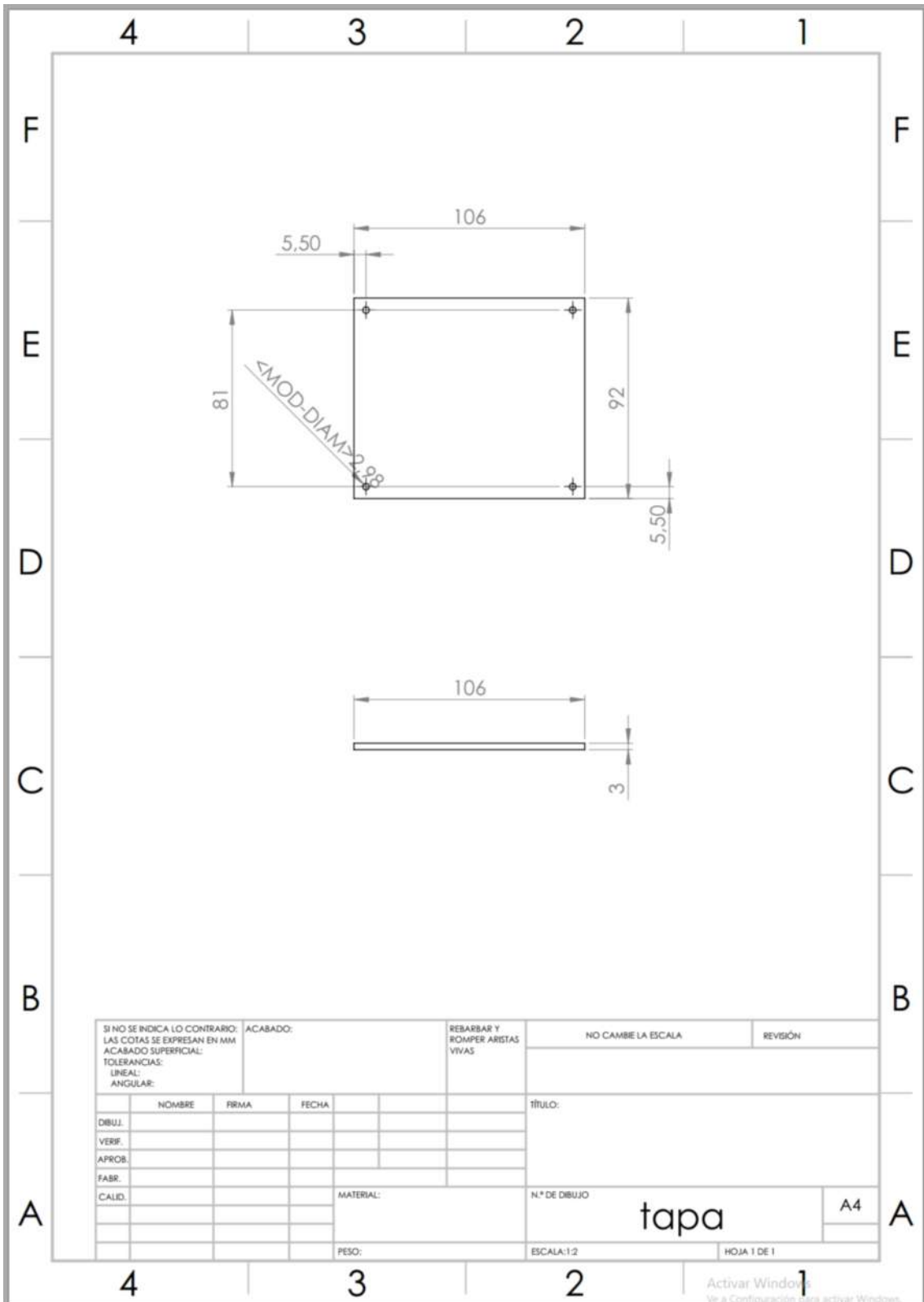
    time.sleep(1)
    chunk = 1024
    wf = wave.open(path+'Captura-nueva-imagen.wav', 'rb')
    p = pyaudio.PyAudio()
    stream = p.open(
        format = p.get_format_from_width(wf.getsampwidth()),
        channels = wf.getnchannels(),
        rate = wf.getframerate(),
        output = True)
    data = wf.readframes(chunk)
    while data != '':
        stream.write(data)
        data = wf.readframes(chunk)
    stream.close()
    p.terminate()
    return infD
```

ANEXO B: Planos estructurales del dispositivo

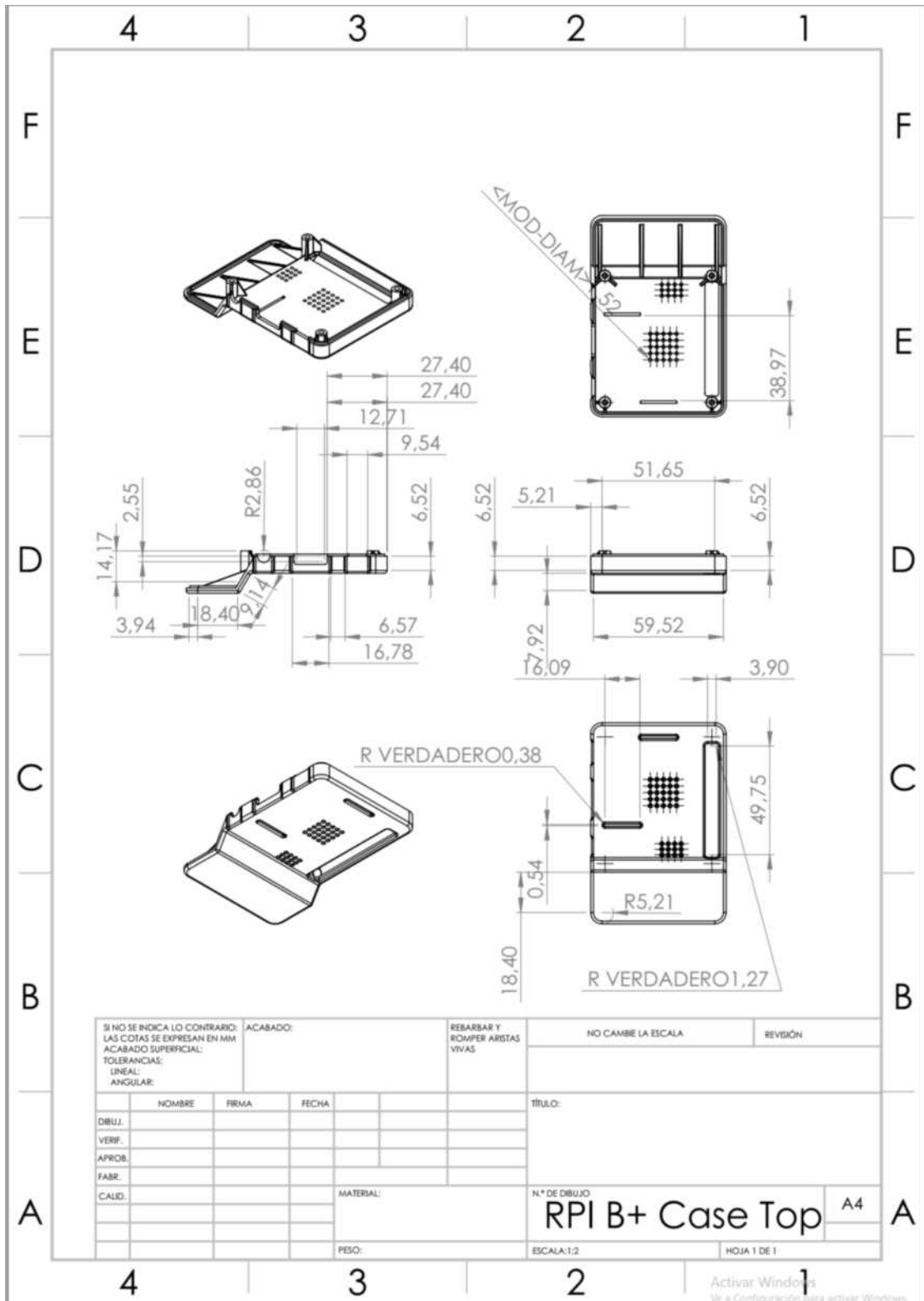
Tapa del dispositivo



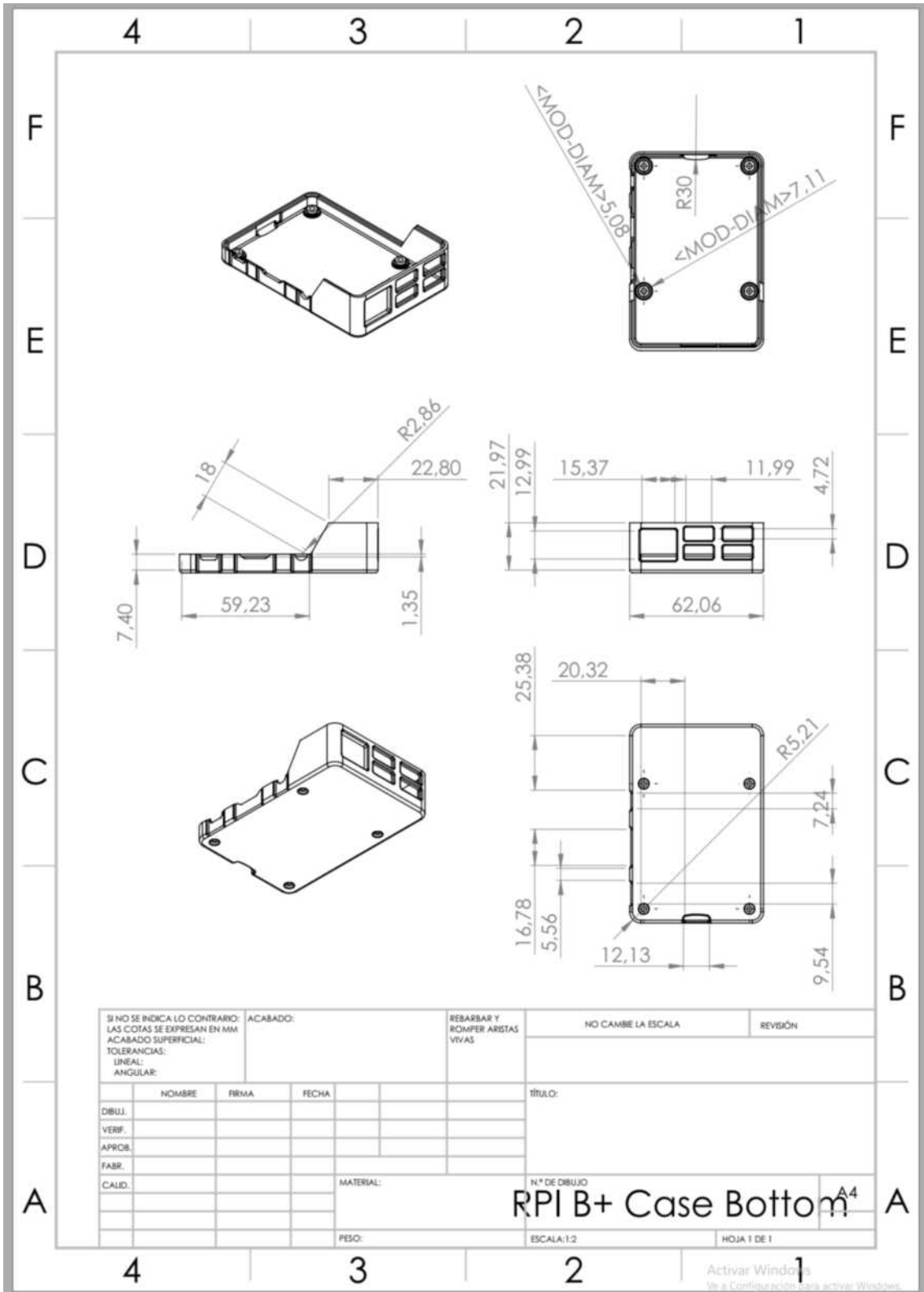
Base del dispositivo



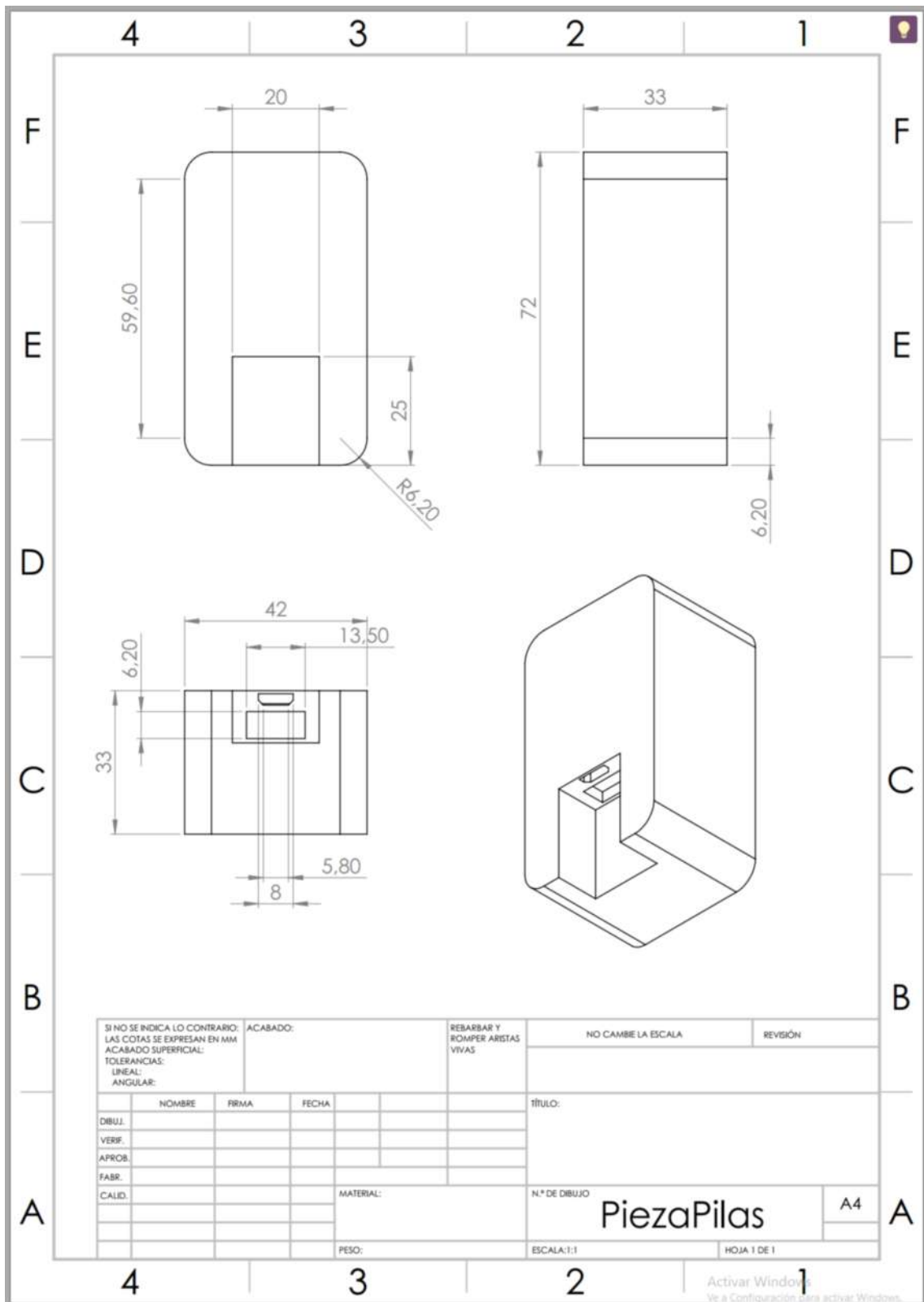
Tapa para Raspberry pi



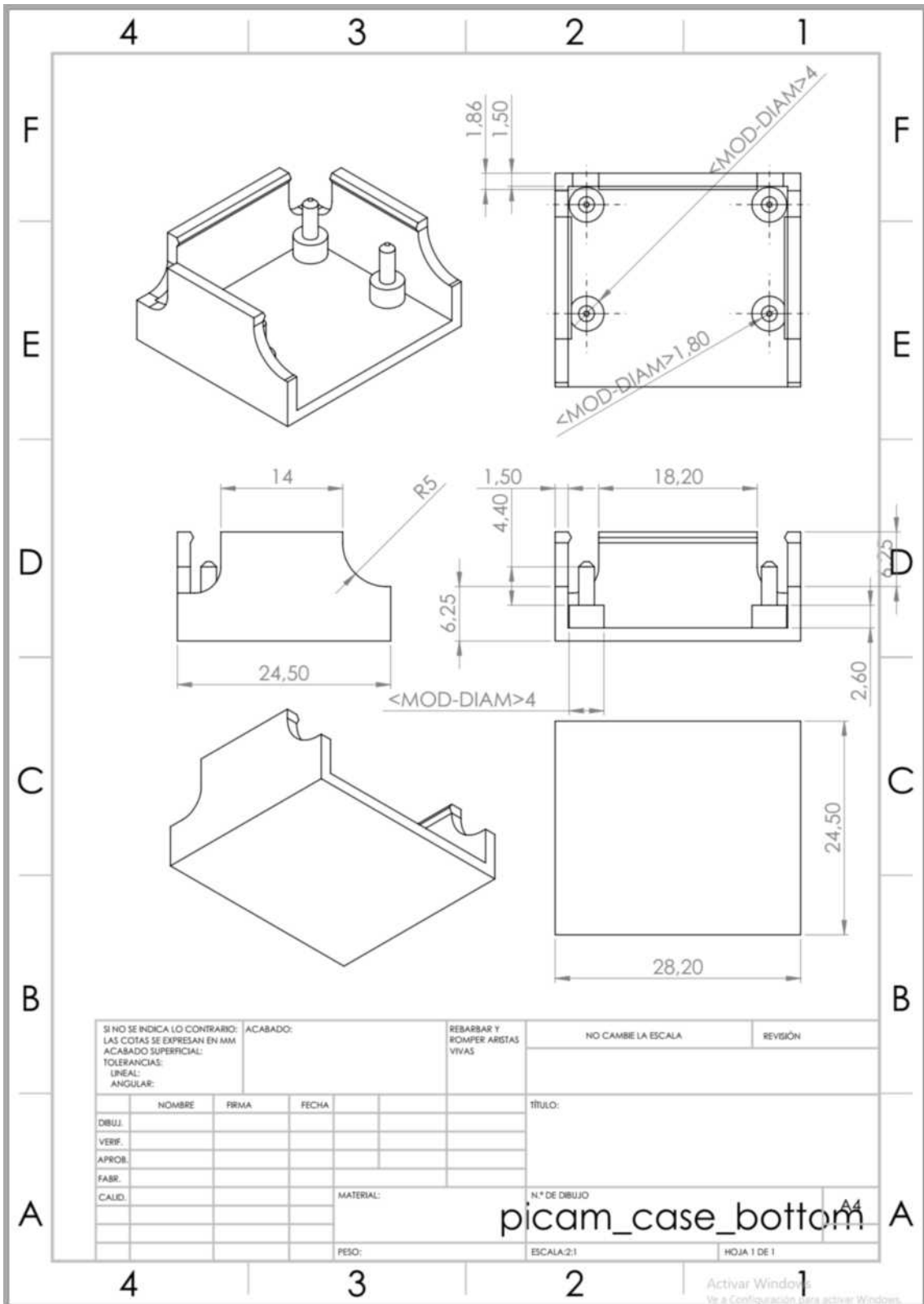
Base para la raspberry pi



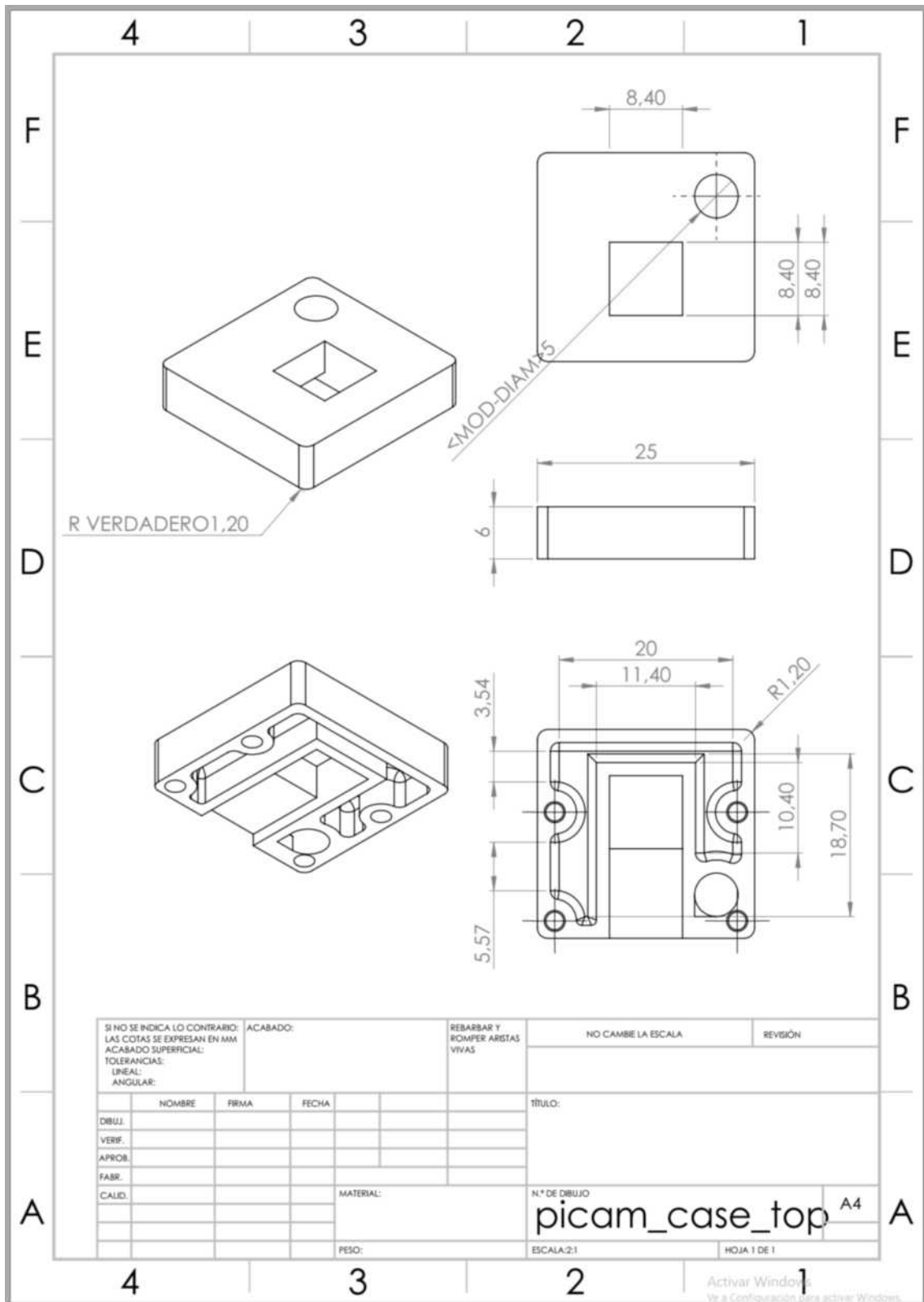
Case para la batería



Base para la cámara Raspberry



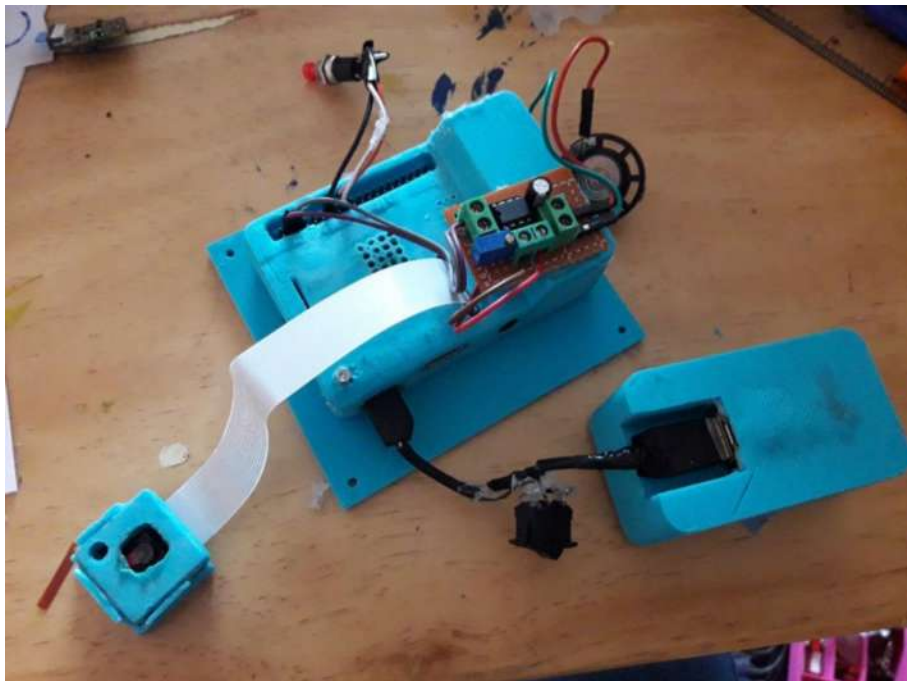
Tapa para la cámara



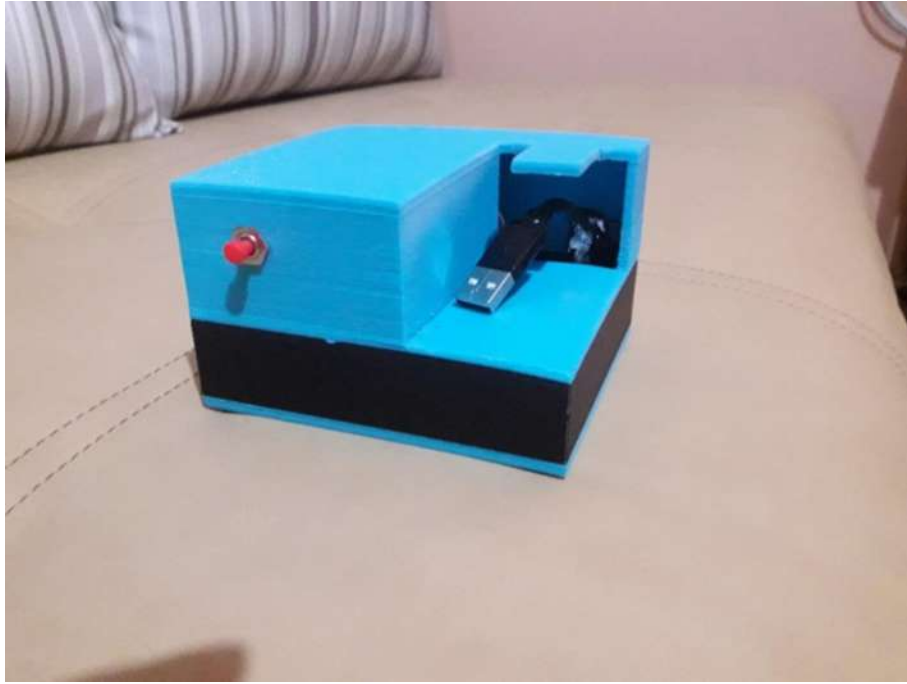
ANEXO C: Montaje del dispositivo



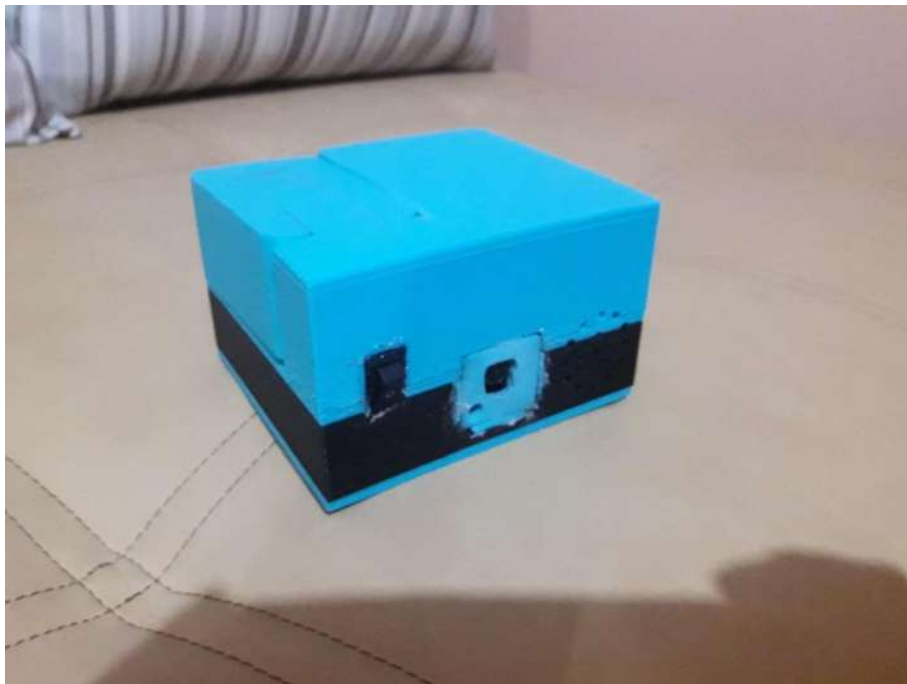
Realizado por: Solano, G. 2018



Realizado por: Solano, G. 2018



Realizado por: Solano, G. 2018



Realizado por: Solano, G. 2018