

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería



Diseño e implementación de un algoritmo detector de billetes de
Quetzal para los no videntes

Trabajo de graduación en modalidad de tesis presentado por
Rodrigo Alberto Mejia Guzman
para optar al grado académico de Licenciado en Ingeniería
Electrónica

2016


Diseño e implementación de un algoritmo detector de billetes de
Quetzal para los no videntes



Diseño e implementación de un algoritmo detector de billetes de
Quetzal para los no videntes

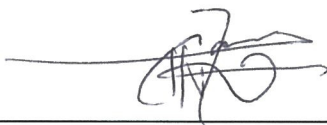
Trabajo de graduación en modalidad de tesis presentado por
Rodrigo Alberto Mejia Guzman
para optar al grado académico de Licenciado en Ingeniería
Electrónica

Vo. Bo.:

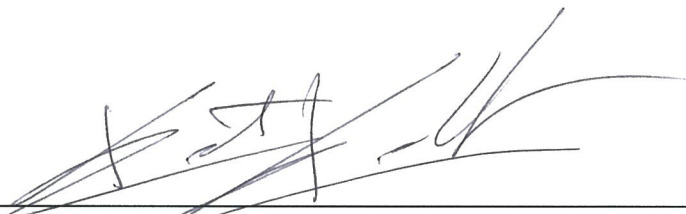
(f)  _____

Ingeniero Carlos Alberto Esquit Hernández


Tribunal Examinador:

(f)  _____

Ingeniero Carlos Alberto Esquit Hernández

(f)  _____

Ingeniero Kurt Kellner

(f)  _____

Ingeniero Pablo Oliva

Fecha de Aprobación: Guatemala 5 de Diciembre del 2016

PREFACIO

Cuando BBDO llegó a proponer este proyecto a la Universidad yo de inmediato me ofrecí como voluntario para desarrollarlo sin saber exactamente qué era lo que buscaban. Además, quería la experiencia de trabajar con personas ajenas a la Universidad mientras desarrollaba un proyecto.

Un año después puedo decir que este proyecto me ha enseñado mucho. No me refiero a los fundamentos teóricos o las prácticas que implementé mientras lo desarrollaba. Pienso que lo que este proyecto me ha hecho ver es el entorno y la presión que se maneja en el mundo laboral en el ámbito de desarrollos de proyectos.

Cuando se nos exige lo suficiente podemos ver con claridad lo que deseamos y lo que consideramos valioso para nuestra vida. Soy capaz de decir que ahora tengo una mejor visión de mi vida y fue gracias a este proyecto. Gracias a este proyecto me he logrado conocer mejor a mí mismo y me ha hecho reevaluar los objetivos en mi vida.

ÍNDICE

Prefacio.....	vi
Índice.....	vii
Listado de figuras.....	ix
Listado de cuadros.....	xi
Resumen.....	xii
I. Introducción.....	1
II. Objetivos	3
A. Generales.....	3
B. Específicos	3
III. Justificación.....	4
IV. Marco teórico	5
A. DISCAPACIDAD VISUAL	5
1. Errores refractivos	5
2. Cataratas.....	7
3. Glaucoma.....	9
B. Tratamiento digital de imágenes.....	10
1. Imagen digital	10
2. Imagen vectorial	10
3. Imágenes en mapa de bits.....	11
a. Modos de color.....	12
4. Procesamiento de imágenes.....	14
a. Captura.....	14
b. Segmentación	14
c. Extracción y comparación de características.....	15

V.	Diseño experimental	17
A.	Desarrollo del algoritmo	17
B.	Selección y configuración de la plataforma	20
C.	Implementación del algoritmo	22
D.	Diseño y construcción del estuche	24
E.	Pruebas de la implementación.....	24
F.	Pruebas con no videntes.....	25
VI.	RESULTADOS	26
A.	Algoritmo resultante	26
B.	Resultados de análisis	27
C.	Estuche fabricado	31
D.	Resultados de pruebas en ambientes diferentes.....	32
1.	Capturas en distintos ambientes.....	32
2.	Porcentaje de acierto en un ambiente controlado	34
3.	Porcentaje de acierto al aire libre.....	34
4.	Porcentaje de acierto bajo el Sol	34
E.	Retroalimentación por parte de no videntes	35
VII.	Análisis de resultados	36
VIII.	Conclusiones.....	40
IX.	Recomendaciones	41
X.	Bibliografía	42

LISTADO DE FIGURAS

Figura 1. Comparación de enfoque de luz en un ojo normal con uno con miopía.	6
Figura 2. Comparación de enfoque de luz en un ojo normal con uno con hipermetropía.....	6
Figura 3. Comparación de enfoque de luz en un ojo normal con uno con astigmatismo.....	7
Figura 4. Ilustración del cristalino del ojo.....	8
Figura 5. Comparación de vista en un ojo normal y otro con catarata.	8
Figura 6. Ilustración del nervio óptico.	9
Figura 7. Comparación de vista de un ojo normal con un con glaucoma.....	10
Figura 8. Ejemplo de imagen vectorial.....	11
Figura 9. Representación explícita de una imagen en mapa de bits.....	11
Figura 10. Comparación entre profundidades de color; original, 5, 4, 3, 2, y un bit respectivamente.....	12
Figura 11. Representación gráfica del modo RGB.	13
Figura 12. Representación gráfica del modo HSB.	13
Figura 13. Representación gráfica del modo CMYK.	14
Figura 14. Comparación de características de una imagen.	15
Figura 15. Diagrama de flujo del algoritmo resultante.	26
Figura 16. Ejemplo de imagen de referencia.....	27
Figura 17. Imagen de referencia analizada, cada punto verde es un punto de referencia.	27
Figura 18. Ejemplo de imagen capturada por medio de la cámara.....	28
Figura 19. Análisis completo de la imagen capturada.....	28
Figura 20. Comparación de puntos de referencia coincidentes con un billete completo.	28
Figura 21. Comparación de puntos de referencia con un billete completo luego de aplicar el umbral de aceptación.....	29
Figura 22. Extracción de rostro de la imagen capturada.....	29
Figura 23. Rostro extraído analizado por puntos de referencia.....	29
Figura 24. Puntos de referencia coincidentes con el rostro extraído.	30
Figura 25. Puntos de referencia coincidentes con el rostro extraído luego de aplicar el umbral de aceptación.....	30
Figura 26. Estuche fabricado a base de madera.....	31
Figura 27. Detalle de relieve para denotar las funciones.	31
Figura 28. Captura en ambiente controlado.	32

Figura 29. Captura al aire libre, bajo la sombra.....	32
Figura 30. Capturas al aire libre bajo el Sol.....	33
Figura 31. Lic. Laida e Ing. Carrilo.....	35

LISTADO DE CUADROS

Cuadro 1. Resultados utilizando ORB en un ambiente controlado.	34
Cuadro 2. Resultados utilizando SIFT en un ambiente controlado.	34
Cuadro 3. Porcentaje de acierto al aire libre con iluminación neutra.	34
Cuadro 4. Porcentaje de acierto bajo el Sol.	34

RESUMEN

El presente trabajo de graduación consiste en el diseño e implementación de un algoritmo detector de billetes de Quetzal para los no videntes. El objetivo principal consistió en proporcionar un sistema que facilitara la recepción de dinero a los no videntes por medio de la detección de la denominación de los billetes.

Como primer objetivo se tenía el diseño del algoritmo de reconocimiento de billetes de Quetzal usando como principal herramienta el procesamiento de imágenes. Finalmente utilizando las librerías de OpenCV y el lenguaje de programación Python se logró desarrollar un algoritmo que reconociera la denominación de los billetes de Quetzal.

Como segundo objetivo se tenía implementar un sistema integrado portátil para el reconocimiento de billetes. Finalmente se utilizó la plataforma Raspberry Pi que ofrecía flexibilidad en cuanto a la cámara a utilizar y respecto a la comunicación con el usuario.

Como tercer objetivo se tenía el diseño y la construcción de un estuche o envoltura que contuviera todos los componentes necesarios para el funcionamiento del sistema. Para el diseño en tres dimensiones se utilizó la plataforma en línea Tinkercad, luego se trasladó el diseño a Inventor para que las piezas fueran fabricadas por medio de una cortadora laser.

Como cuarto objetivo se tenía realizar pruebas de campo para evaluar el desempeño del algoritmo en diferentes ambientes. Finalmente se obtuvo, en promedio, un 90.67% de acierto.

Como último objetivo se tenía obtener retroalimentación por parte de usuarios no videntes para comprobar la funcionalidad en el público objetivo. Se obtuvo la retroalimentación deseada teniendo así puntos de mejora importantes para los no videntes.

I. INTRODUCCIÓN

El presente trabajo de graduación consiste en el diseño e implementación de un algoritmo procesador de imagen que reconoce las distintas denominaciones de billetes de la moneda oficial de Guatemala, el Quetzal, para ser usado por los no videntes. El diseño representa el componente de investigación y el diseño del componente físico, mientras que la implementación representa las pruebas y modificaciones realizadas al algoritmo en el transcurso del avance del proyecto, así como la construcción del sistema donde se integrará.

El proyecto implementado está dividido en dos secciones: el módulo de captura de imagen y el módulo de procesamiento de la misma. El módulo de captura de imagen tiene la responsabilidad de, tal y como lo expresa su nombre, capturar la imagen a procesar por el siguiente módulo. El segundo es el encargado de procesar la imagen para que el resultado sea el del correcto reconocimiento de la denominación del billete.

Se investigaron diferentes plataformas diseñadas para montar sistemas integrados que podrían utilizarse para realizar el procesamiento de imágenes capturadas. Con base en los requerimientos del proyecto se optó por utilizar la Raspberry Pi, por su alto desempeño y compatibilidad con el lenguaje de programación Python.

Se investigaron distintas cámaras que podrían utilizarse para un proyecto de este tipo y se contempló la posibilidad de una cámara web. Sin embargo, se aprovechó que la plataforma Raspberry Pi posee una cámara que puede ser utilizada nativamente, por lo tanto, se optó por adquirir dicha cámara.

Posteriormente se escogió el lenguaje de programación a utilizar, siendo este Python. La elección se basó principalmente en la facilidad de uso y la flexibilidad del mismo para ser utilizado en distintas plataformas de procesamiento. Asimismo, se optó por utilizar las librerías de OpenCV, por ser estas de licencia gratuita y por su vasta documentación.

Se procedió a la implementación del algoritmo de procesamiento de imágenes. En un inicio este se evaluó y ajustó en una computadora personal, cuando los resultados llegaron a ser satisfactorios se migró la implementación a la Raspberry Pi 3. Seguidamente se realizaron pruebas con la cámara elegida y se realizaron los ajustes necesarios para garantizar el buen funcionamiento del algoritmo.

Por último, se procedió al diseño y construcción del componente físico del proyecto. El cual consistía en la elaboración de un estuche el cual tendría como función alojar y proteger la cámara y la plataforma de procesamiento.

Con el sistema completo construido y funcionando correctamente se procedió a realizar una reunión con personas no videntes para que pudieran expresar su opinión acerca del sistema y pudieran proveer retroalimentación para saber cómo sería posible mejorar el sistema.

II. OBJETIVOS

A. GENERALES

Proporcionar un sistema que facilite la recepción de dinero a los no videntes por medio de la detección de la denominación de los billetes.

B. ESPECÍFICOS

- Diseñar y evaluar un algoritmo de procesamiento de imagen para el reconocimiento de billetes.
- Implementar un sistema integrado portátil para el reconocimiento de billetes.
- Diseñar y construir un estuche o envoltura en dónde todos los componentes del sistema estén contenidos.
- Realizar pruebas de campo para evaluar el desempeño del algoritmo en diferentes ambientes.
- Obtener retroalimentación por parte de usuarios no videntes para comprobar la funcionalidad del sistema en el público objetivo.

III. JUSTIFICACIÓN

Actualmente 0.52% de la población mundial sufre de alguna discapacidad visual (WHO, 2014), y aproximadamente un 0.71% de la población de Guatemala (INE, 2005). La ceguera impide al afectado el conllevar una vida completa gracias a los obstáculos presentados debido a su discapacidad.

Uno de los obstáculos más comunes para las personas con alguna discapacidad visual es el manejo del dinero ya que no les es posible diferenciar con seguridad la denominación del billete que poseen sin la ayuda de una segunda persona. Este problema sale a relucir cuando dichas personas deben recibir dinero, ya que no es posible hacerlo sin cierta incertidumbre debido a que necesitan confiar que la persona entregando el dinero está siendo honesta con ellos. Generalmente los no videntes prefieren preguntarle a una persona de su confianza la denominación de los billetes para luego ellos organizarlos como más les convenga. Sin embargo, eso quiere decir que de una forma u otra siempre necesitan a alguien más aparte de su propia persona para manejar dinero.

Una herramienta para el manejo de dinero representa una mayor libertad para los no videntes dado que les da la independencia necesaria al momento de recibirlo, ya no necesitan de otra persona con ellos. Esto quiere decir que la herramienta es fácil de usar, y al mismo tiempo no invasiva para los usuarios.

IV. MARCO TEÓRICO

A. DISCAPACIDAD VISUAL

La función visual esta subdividida en cuatro niveles, en congruencia con la Clasificación Internacional de Enfermedades (CIE-10) (WHO, 2016):

- Visión normal;
- Discapacidad visual moderada;
- Discapacidad visual grave;
- Ceguera.

La discapacidad visual moderada y la discapacidad grave se reagrupa bajo el término “baja visión”, la baja visión y la ceguera representan conjuntamente el total de casos de discapacidad visual (WHO, 2014). Actualmente hay aproximadamente 285 millones de personas con algún tipo de discapacidad visual, de las cuales 39 son ciegas y 246 millones presentan baja visión (WHO, 2014). Cabe resaltar que una persona con baja visión es una que tenga alteración de la función visual incluso después de tratamiento o corrección de refracción estándar (WHO, 2014). Las causas más comunes de discapacidad visual son los casos de ametropía no tratados, cataratas y el glaucoma (WHO, 2012).

Aproximadamente el 90% de las personas que sufren de alguna discapacidad visual son de países en desarrollo. Aproximadamente el 65% de las personas con discapacidad visual son mayores de 50 años. Un estimado de 19 millones de niños sufren de algún tipo de discapacidad visual, sin embargo, 12 millones de estos son causados por casos de ametropía o errores refractivos que no son tratados (WHO, 2014)

1. Errores refractivos. Los errores refractivos, o ametropicos, ocurren cuando la forma del ojo previene que la luz se concentre directamente en la retina. El largo del ojo, ya sea más corto o largo, cambia la forma de la córnea, lo que causa este tipo de defectos. La refracción es la curvatura de la luz al pasar de un medio a otro. La vista ocurre cuando los rayos de luz se refractan cuando pasan a través de la córnea y el cristalino, y luego la luz se concentra en la retina. Luego la retina convierte los rayos de luz a señales eléctricas que viajan por el nervio óptico hacia el cerebro. Las imágenes que vemos son la interpretación del cerebro de dichos mensajes. Los errores refractivos más comunes son la miopía, la hipermetropía, el astigmatismo y la presbicia (NEI, 2016).

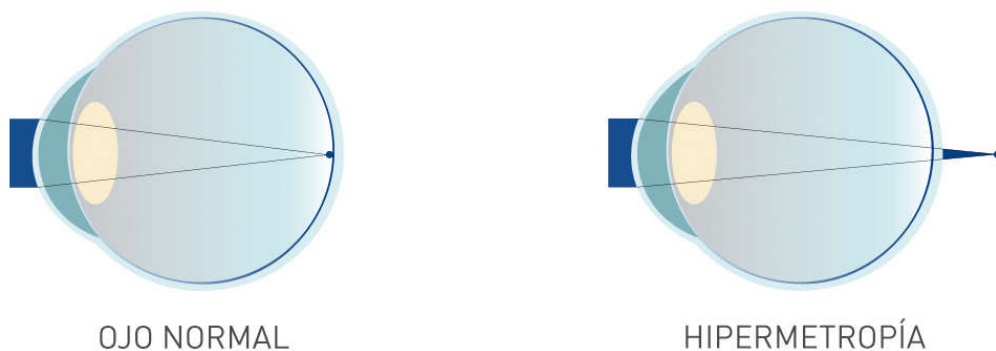
La miopía es la condición en donde los objetos se ven nítidos cuando es tan cerca, mientras que los objetos lejanos parecen borrosos. Con la miopía, el punto de enfoque de la luz dentro del ojo es frente a la retina (NEI, 2016).

Figura 1. Comparación de enfoque de luz en un ojo normal con uno con miopía.



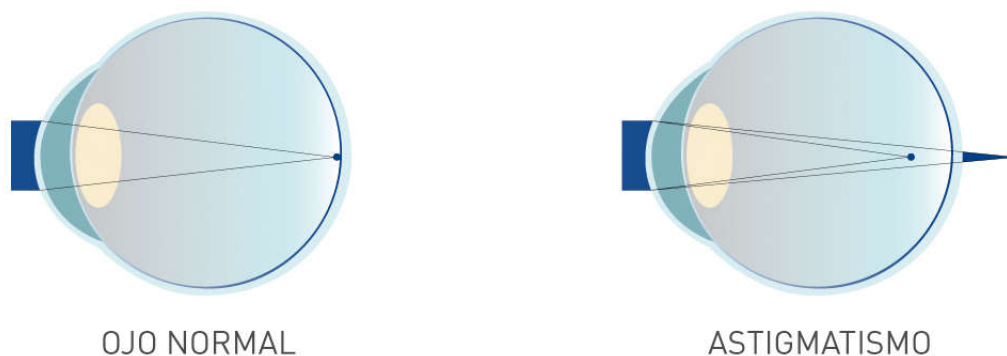
La hipermetropía consiste en que los objetos distantes son los que se ven más claros que los objetos que están cerca. Sin embargo, todos experimentan la hipertrofia de forma distinta, ya que en algunos casos extremos la mirada puede ser borrosa a cualquier distancia de enfoque (NEI, 2016).

Figura 2. Comparación de enfoque de luz en un ojo normal con uno con hipermetropía.



El astigmatismo sucede cuando la luz no se enfoca de igual forma en la retina, otra forma de verlo es que el punto de enfoque estaría detrás del ojo, esto causa que las imágenes se vean borrosas y estiradas.

Figura 3. Comparación de enfoque de luz en un ojo normal con uno con astigmatismo.



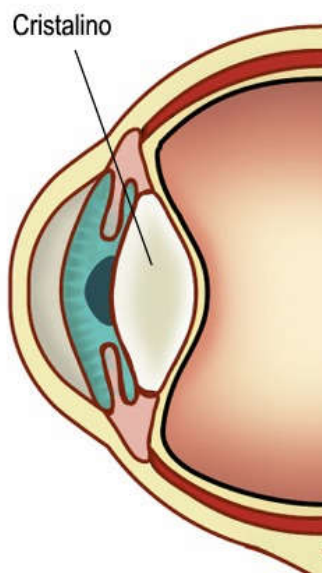
La presbicia es una condición relacionada con la edad en la que el individuo poco a poco pierde la habilidad de enfocar de cerca. Mientras el sujeto envejece el cristalino del ojo ya no es capaz de cambiar de forma para enfocar los objetos de forma correcta (NEI, 2016).

La presbicia afecta principalmente a los adultos mayores a 35 años mientras que los demás errores pueden afectar a individuos de todas las edades. El síntoma más común es una visión borrosa pero también se incluyen la doble visión, ojos cansados, mareos y dolores de cabeza. Entre los tratamientos se incluyen usar anteojos para corregir el problema, lentes de contacto, o bien es posible realizar una cirugía refractiva para cambiar la forma del ojo de forma permanente y corregir el problema (NEI, 2016).

2. Cataratas. Una catarata es una opacidad del cristalino del ojo que afecta la visión. Dicha opacidad provoca que la luz se disperse dentro del ojo y que no se pueda enfocar en la retina. Puede afectar uno o ambos ojos. La mayor causa de cataratas mundialmente es la edad; sin embargo también puede ser causado por el cigarro y la diabetes (NEI, 2016).

El cristalino es la parte clara del ojo que ayuda a enfocar la luz a la retina. La retina es el tejido sensible a la luz detrás del ojo. En un ojo normal, la luz pasa por el cristalino hacia la retina, y cuando lo hace esta luz es interpretada como señales eléctricas por el cerebro. El cristalino debe estar transparente para que la retina sea capaz de recibir una imagen nítida, si existe una catarata la imagen recibida será borrosa (NEI, 2016).

Figura 4. Ilustración del cristalino del ojo.



El cristalino está detrás del iris y la pupila. Está constituido principalmente por agua y proteína, la cual está organizada de una forma específica que mantiene el cristalino despejado y deja pasar la luz a través de él. Sin embargo, mientras vamos envejeciendo la proteína se agrupa y poco a poco se comienza a nublar una pequeña área del cristalino, esto es una catarata (NEI, 2016).

El agrupamiento de la proteína reduce la nitidez en la que la imagen es proyectada a través del cristalino, lo que resulta en que la imagen interpretada por el cerebro sea borrosa. Cuando una catarata es pequeña lo borroso solo afectará una parte del cristalino, lo cual no es inmediatamente evidente. A través del tiempo las cataratas suelen crecer, causando que la vista se torne más borrosa y más opaca. Asimismo, la catarata también causa que el cristalino cambie de color, a un tono amarillento o café. Esto causa que las imágenes percibidas también posean este tono (NEI, 2016).

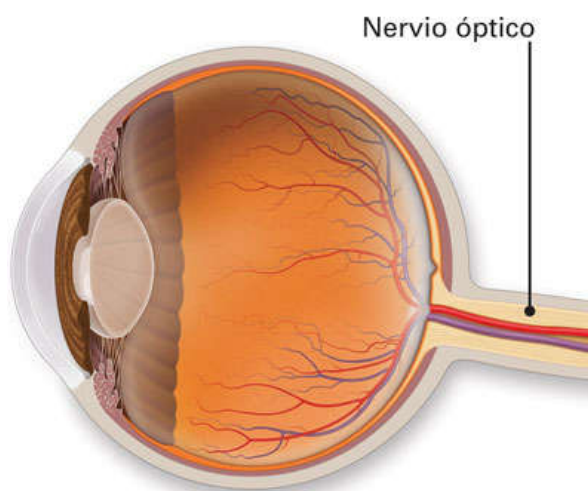
Figura 5. Comparación de vista en un ojo normal y otro con catarata.



Las cataratas pueden ser tratadas tempranamente por medio de anteojos, anteojos oscuros anti reflejo o lentes de aumento. Si estas medidas no ayudan al paciente la única alternativa es la cirugía. Esta cirugía consiste en remover el cristalino nublado y defectuoso y reemplazarlo con un cristalino artificial (NEI, 2016).

3. Glaucoma. Glaucoma es un grupo de enfermedades que dañan el nervio óptico y pueden resultar en la pérdida de la vista y en ceguera. El nervio óptico es un grupo de más de un millón de fibras ópticas nerviosas. Conecta la retina con el cerebro es por eso que es necesario que funcione de forma correcta. El tipo de glaucoma más común es el de ángulo abierto (NEI, 2016).

Figura 6. Ilustración del nervio óptico.



La presión intraocular es un factor de riesgo importante cuando se habla de dañar el nervio óptico. En la parte frontal del ojo esta la cámara anterior, un fluido claro fluye continuamente hacia dentro y hacia afuera de la cámara y nutre los tejidos cercanos. El fluido sale de la cámara en el ángulo abierto en el que la córnea y el iris se unen, cuando el fluido llega al ángulo, fluye por una malla esponjosa, como un drenaje, y sale del ojo. En el glaucoma de ángulo abierto el fluido pasa demasiado lento a través de la malla esponjosa, lo que causa que este se acumule, que a su vez resulta en que la presión dentro del ojo crece hasta dañar el nervio óptico. Cuando se daña el nervio óptico como resultado de alta presión intraocular, el glaucoma de ángulo abierto y la pérdida de visión, pueden resultar (NEI, 2016).

Figura 7. Comparación de vista de un ojo normal con un con glaucoma.



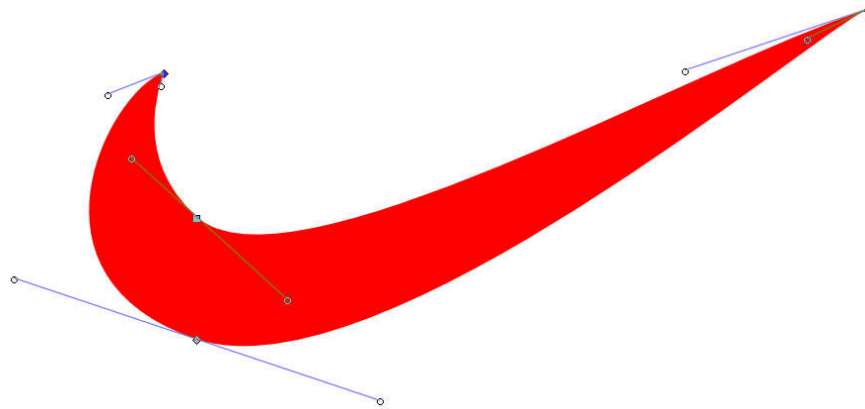
No todas las personas con presión intraocular desarrollan glaucoma de ángulo abierto. Esto varía debido a que algunas personas pueden tolerar más presión que otras. La probabilidad de que alguien desarrolle glaucoma depende de la presión que el nervio óptico soporta, este umbral es diferente para cada persona. Para determinar dicho umbral es necesario realizar un examen de los ojos dilatados (NEI, 2016).

B. TRATAMIENTO DIGITAL DE IMÁGENES

1. **Imagen digital.** Una imagen, en su definición más simple, es una representación de un objeto real, en otras palabras, es la representación plana de un objeto en tres dimensiones (UNICEN, 2016). Las imágenes digitales son aquellas que se obtienen a través de dispositivos de conversión analógica-digital, como las cámaras fotográficas digitales. Toda información digital es almacenada en unidades de información binaria, bits (ITE, 2016). Sin embargo, dependiendo de la técnica de codificación que se utilice las imágenes pueden ser de dos tipos, vectoriales o mapa de bits (Grimaldos, 2005).

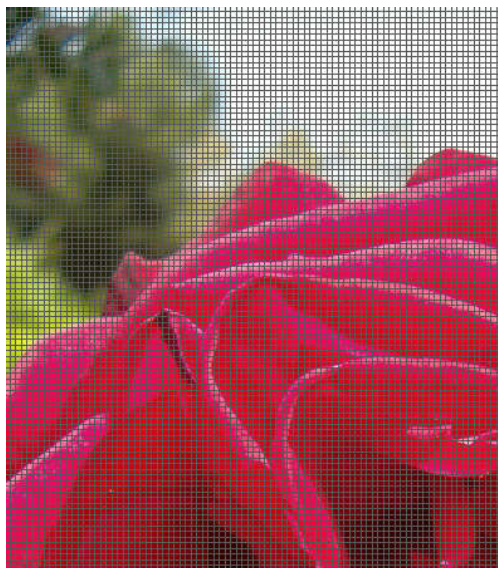
2. **Imagen vectorial.** Las imágenes vectoriales están compuestas de contornos y rellenos que se definen de forma matemática por medio de ecuaciones que describen a la perfección cada ilustración. Gracias a esto dichas imágenes pueden ser escaladas sin que su calidad se vea afectada, dado que al momento de aumentar o disminuir su tamaño solamente se tiene que recalcular sus parámetros (Grimaldos, 2005).

Figura 8. Ejemplo de imagen vectorial.



3. **Imágenes en mapa de bits.** Las imágenes de mapa de bits están descritas por una matriz de pequeños cuadrados, llamados píxeles, que están rellenos de un color específico. Para comprender este tipo de imágenes basta con suponer que deseamos reproducir una fotografía en una hoja en cuadrícula y que cada cuadro solo puede estar relleno de un color (Grimaldos, 2005). Con esto nos damos cuenta que mientras más pequeños sean los cuadros, más nítida será la imagen, la medida de cuantos píxeles se tienen en un espacio determinado es lo que llamamos resolución (ITE, 2016).

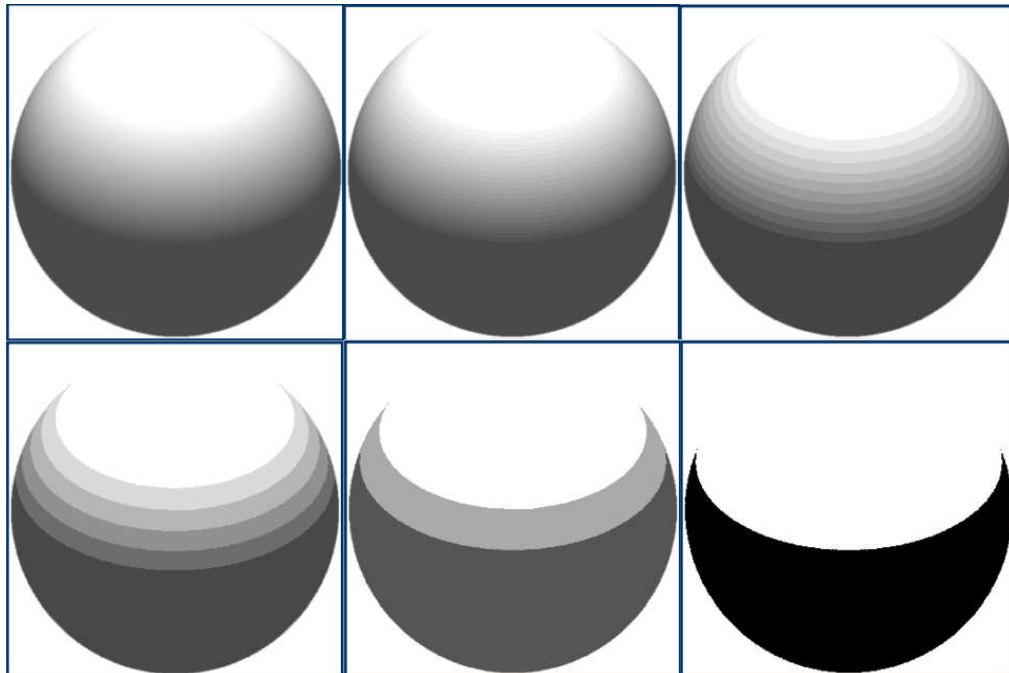
Figura 9. Representación explícita de una imagen en mapa de bits.



La resolución de una imagen es la medida de pixeles que la describen. Esta es medida en pixeles por pulgada, PPP, o dpi por sus siglas en inglés (dots per inch), y de ella depende la calidad de la imagen, así como el tamaño que ocupa en memoria. Por supuesto mientras más alta es la resolución de una imagen esta posee más pixeles que la describan lo que resulta en una mejor representación de la imagen usando un dispositivo de salida que se adecua a nuestros requerimientos (Grimaldos, 2016).

Una característica específica para las imágenes en mapa de bits es la profundidad de color. Esta se refiere al número de bits necesarios para codificar y guardar la información de cada color de cada pixel. Un bit solamente puede poseer el valor de 0 o 1. Cuanto mayor sea la profundidad de color en bits, la imagen tendrá una paleta mayor de colores. Para saber cuántos colores se tienen disponibles basta con utilizar la fórmula 2^n en donde n es la profundidad de color (UNICEN, 2016).

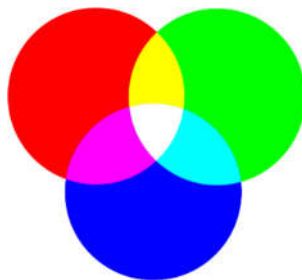
Figura 10. Comparación entre profundidades de color; original, 5, 4, 3, 2, y un bit respectivamente.



a. Modos de color. Se le llama modo de color al sistema de coordenadas que nos permiten describir el color de cada pixel utilizando valores numéricos. Los principales modos de color son RGB (rojo, verde y azul), HLS (tono, luminosidad, saturación) y el CMYK (cian, magenta, amarillo y negro) (Grimaldos, 2005).

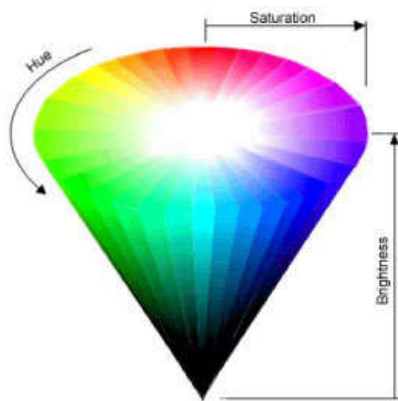
El modo RGB corresponde a cuando un color es el resultado de la mezcla de tres canales distintos. Cada canal corresponde a un color primario: rojo, azul, verde. Cada uno de estos se le asigna una intensidad que oscila entre 0 y 255. De la combinación de estos tres colores pueden surgir 16.7 millones de colores distintos (ITE, 2016).

Figura 11. Representación gráfica del modo RGB.



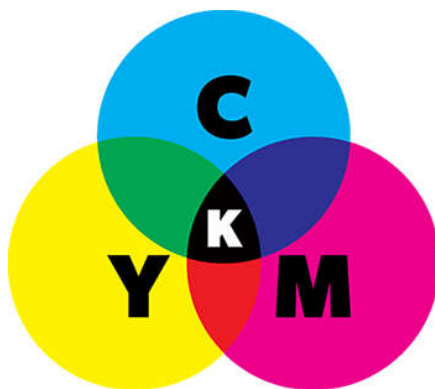
El modo HSB posee tres parámetros: el tono, la saturación, y el brillo. El tono se refiere a la longitud de onda de la luz que un objeto refleja. Para medirlo se utiliza una rueda de color estándar donde los colores primarios secundarios se disponen en una circunferencia de modo que cada color esta opuesto a su complementario. El tono se mide en grados desde 0 a 360. La saturación es la intensidad del color, midiéndose en términos de porcentaje de 0 100. El brillo consiste en la claridad u oscuridad de cada tono de color, en otras palabras, cuanto blanco o negro se desea. Este se mide de 0 a 100 por ciento, siendo 0 el negro y 100 el blanco (Grimaldos, 2005).

Figura 12. Representación gráfica del modo HSB.



El modo CMYK posee cuatro canales para cada color, siendo cada canal un color primario de impresión: cian, magenta, amarillo y negro. Al igual que en el modo RGB cada canal se mide entre 0 y 255 y el color final resulta en la combinación aditiva de los cuatro canales (Grimaldos, 2005).

Figura 13. Representación gráfica del modo CMYK.



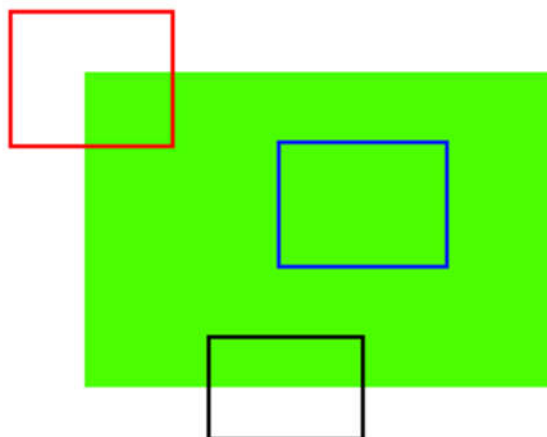
4. **Procesamiento de imágenes.** El procesamiento de imágenes es el conjunto de técnicas matemáticas y computacionales con el objetivo de realzar una o varias clases de patrones espectrales o espaciales de un espacio de datos bidimensionales. Es decir, el procesamiento de imágenes son las transformaciones aplicadas a una imagen para extraer las características deseadas (Lira, 2010).

a. **Captura.** La captura comienza desde la decisión y diseño de las propiedades de la captura (UNICEN, 2016). Luego de eso la adquisición está a cargo de algún transductor o conjunto de transductores que mediante la manipulación de la luz o de alguna otra forma de radiación que es emitida o reflejada por los cuerpos, logra formar una representación fiel del objeto, dando lugar a una imagen. Cabe notar que dependiendo del transductor que se utilice es posible que la imagen obtenida posea ruido y la resolución también estará limitada (Escalante, 2006).

b. **Segmentación.** Se define la segmentación como la partición de una imagen en regiones que pueden o no tener un significado relativo a la escena respectiva (Lira, 2010). En esta etapa se pretende reconocer y extraer cada una de los objetos presentes en la imagen (UNICEN, 2016). Dado que la imagen contiene una variedad de patrones relacionados directamente con objetos de la escena y habrá otros patrones que no tengan ningún significado. Para que el análisis sea realizado con mayor facilidad los patrones de interés son separados del resto de la imagen (Lira, 2010).

c. Extracción y comparación de características. Tal y como su nombre lo indica, esta etapa consiste en extraer la información de interés a partir de la imagen que previamente ha sido pre procesada y segmentada. No existe una definición universal de lo que es una característica y depende del problema que se tiene. Se buscan características que sean únicas y que puedan ser fácilmente rastreadas y comparadas. Para poder entender mejor este concepto se puede observar la Figura 14 (OpenCV, 2016).

Figura 14. Comparación de características de una imagen.



Si decidiéramos mover el recuadro azul vertical u horizontalmente no habría diferencia alguna. Si hiciéramos lo mismo con el recuadro negro obtendríamos el mismo resultado cuando lo moviéramos horizontalmente, pero si lo moviéramos verticalmente notaríamos el cambio. Por último, si moviéramos el recuadro rojo hacia cualquier dirección notaríamos el cambio, lo que quiere decir que el contenido dentro del recuadro es único. Cabe notar que el recuadro rojo contiene una esquina la cual resulta ser un buen punto de referencia al momento de identificar puntos clave en las imágenes. Así que una característica es aquel espacio dentro de la imagen que podemos considerar único. Es decir, aquellas regiones que tienen una alta variación cuando se mueven en cualquier dirección, en pequeñas dimensiones (OpenCV, 2016).

Para poder identificar una esquina por medio de un algoritmo computacional es necesario traducir estos conceptos a una fórmula matemática, lo cual fue realizado por Chris Harris y Mike Stephens en 1988. Harris y Stephens describen una fórmula que encuentra la diferencia de intensidad en una región en todas las direcciones. Con esto desarrollaron un algoritmo llamado *El Detector de Esquinas de Harris*. Si la diferencia de intensidad es suficiente, la región contiene una esquina. Luego se comparan los píxeles dentro de dicha región para determinar el punto exacto de la esquina. Dependiendo de la aplicación se elige un umbral por el que son evaluados los

pixeles dentro de las regiones para determinar si nos interesan como esquinas o no (Harris & Stevens, 1988).

El detector de esquinas descrito anteriormente, aunque muy útil no funciona lo suficientemente rápido en aplicaciones reales por lo que alternativas comenzaron a surgir. Una de estas alternativas fue FAST, *Features from Accelerated Segment Test*, propuesto por Edward Rosten y Tom Drummond. Rosten y Drummond proponen tomar un pixel p como punto de referencia, luego considera una circunferencia de 16 pixeles alrededor de dicho pixel. El algoritmo evalúa cada uno de los pixeles en la circunferencia por medio de un umbral, establecido por el usuario, para determinar si el pixel en la circunferencia es más brillante u oscuro que el pixel de referencia. Para determinar si el pixel de referencia es una esquina al menos n bits deben ser más oscuros o brillantes que p , donde n es propuesto por el usuario (Rosten & Drummond, 2006).

Una vez que se encuentran las esquinas o punto de interés es necesario realizar una descripción acerca del punto encontrado, tal y como el cerebro humano lo hace cuando toma un punto de referencia. Para esto es necesario calcular el gradiente alrededor del punto de interés. Estos gradientes son transformados en una representación que permite la medir el cambio de iluminación y la distorsión de la imagen alrededor del punto de interés. Toda esta información es guardada para cada punto de interés y se le llama descriptores (Lowe, 2004).

Para determinar si una imagen se encuentra dentro de otra o si deseamos detectar un objeto en específico, es necesario comparar los descriptores de ambas imágenes. Esto significa tomar el descriptor de un punto de interés de una imagen y compararlo con todos los descriptores de la segunda imagen. El resultado de esta operación regresa el punto de interés, si es que encuentra uno, más cercano de la segunda imagen. Si se detectan los suficientes puntos de interés en común entre las imágenes, se obtendrá una comparación satisfactoria (OpenCV, 2016).

V. DISEÑO EXPERIMENTAL

A. DESARROLLO DEL ALGORITMO

Primeramente, se tuvo que realizar una ligera investigación acerca de los diferentes enfoques que se podían usar para el problema a la mano. El primer enfoque se basaba en buscar puntos específicos dentro de una imagen, calcular el valor RGB de dicho punto y compararlo con un punto de referencia previamente elegido. Este enfoque se rechazó debido a que se basaba en colores, y al momento de que cambie la iluminación el valor RGB cambia, por lo que este enfoque tendría que tener un umbral de aceptación basado en pruebas en distintos ambientes, y se necesitaba algo más flexible. Segundo se consideró realizar una red neural. Una red neural es básicamente enseñarle a una maquina a pensar basado en una gran cantidad de muestras, aproximadamente tal y como el cerebro humano lo hace. Sin embargo, la cantidad de muestras necesitadas por billete no parecía algo completamente realizable por lo que este enfoque también se rechazó. El tercer enfoque era buscar una imagen dentro de otra, exactamente la misma imagen. Este enfoque parecía más flexible que el primero y no necesitaba una gran recopilación de datos como el segundo, solamente necesitaba una buena muestra. Sin embargo, al querer encontrar exactamente la misma imagen dentro de otra se necesitaba que la imagen de referencia estuviera perfectamente representada para ser encontrada, en caso contrario el reconocimiento fallaría.

Basado en una las necesidades del problema, se llegó a OpenCV. OpenCV es una librería de software de uso libre de visión por computadora. Se diseñó para proveer una infraestructura común para las aplicaciones de visión de computadora. OpenCV posee más de 2500 algoritmos optimizados utilizados para el procesamiento de imagen. Debido a la robustez de dicha librería se creyó conveniente utilizarlas dado que no sería necesario comenzar de cero el diseño del algoritmo y se tendría una documentación extensiva para hacerlo también.

OpenCV es escrito originalmente en C++ sin embargo existen extensiones del mismo para casi todos los lenguajes de programación (OpenCV, 2016). Debido a la familiaridad que ya se tenía con dicho lenguaje se decidió utilizar Python como lenguaje de programación para el desarrollo de este proyecto.

Con OpenCV disponible se llegó a analizar el uso de la técnica conocida como el emparejamiento de características. Esta técnica extrae puntos de referencia de una imagen y la busca dentro de otra. Bajo esta premisa se comenzó a analizar distintas imágenes de billetes para determinar si era posible utilizar dicha técnica para resolver el problema establecido. El punto consistiría en analizar una imagen de referencia para obtener la mayor cantidad de puntos de referencia posible, para luego buscar dichos puntos en una imagen capturada.

El emparejamiento de características tiene dos fases principales: la detección de puntos de referencia y el cálculo de los descriptores de cada uno de los puntos de referencia. Este proceso ya fue descrito anteriormente sin embargo se tiene que notar el hecho que ya existen varios algoritmos que realizan dichas funciones. Es decir que existen algoritmos encargados de encontrar puntos de referencia, otros que se encargan solamente de calcular descriptores, y existen otros capaces de realizar ambas cosas. Para este proyecto se buscó uno que pudiera hacer ambas cosas, por lo que se llegó a usar ORB y SIFT. Ambos algoritmos son capaces de analizar imágenes y encontrar puntos de referencia, así como sus descriptores, además están basado en algoritmos previamente descritos, como FAST (Rublee, 2011).

Para los puntos de referencia ORB utiliza básicamente FAST para encontrar puntos destacados, pero luego aplica la fórmula descrita por Harris para encontrar “esquinas” dentro del set de puntos previamente encontrados para obtener la mayor cantidad de puntos confiables de referencia. Luego para sus descriptores ORB utiliza una variación del algoritmo BRIEF para calcular los descriptores de cada uno de los puntos de referencia.

SIFT, por su parte analiza las imágenes basándose en la posibilidad de encontrar características en varios tamaños. Primero se busca extremos locales dentro de la imagen, es decir que un pixel es analizado contra los ocho pixeles alrededor de él, así como los nueve pixeles en la escala siguiente y la anterior. Si el pixel resulta ser un extremo local se toma como candidato para ser un punto de referencia. Luego de encontrar todos los posibles candidatos se analiza su cambio en intensidad al ser expandidos en el espacio, si la diferencia en intensidad es muy baja se rechaza el punto. De este modo se eliminan los puntos con bajo contraste. Seguido de esto se calcula el descriptor para cada punto tomando de referencia un bloque de 16x16 pixeles alrededor del punto de referencia. Un punto importante que para utilizar SIFT con un motivo de lucro esta patentado por lo que es necesario pagar una cantidad al año para usarlo (Lowe, 2004).

Cabe mencionar que, aunque ORB y SIFT funcionan de maneras distintas el resultado de ambos es el mismo, un set de puntos de referencia y sus descriptores. Es por eso que el algoritmo

completo no sufriría cambios sin importar cuál de estos se usara. Se tuvo en cuenta los dos para analizar los resultados que se obtuvieran de ambos para llegar a una conclusión de cual utilizar.

Utilizando ambos algoritmos ya era posible comenzar a diseñar un algoritmo que los contuviera para lograr reconocer el billete. Al principio se tenía como imagen de referencia todo el frente de un billete para intentar reconocerlo en cualquier imagen. Esto, sin embargo, rápidamente fue descartado. Al momento en que se analiza una imagen no es posible indicarles a los algoritmos donde buscar puntos de referencia lo que resultaba en que este encontraba puntos de referencia que no resultarían convenientes reconocer en otras imágenes.

Se comenzó a buscar alternativas para segmentar la imagen del billete tomando en cuenta los puntos clave de un billete. Se consideró el número de serie, así como el diseño del número de cada billete. Sin embargo, la característica más consistente a través de cualquier línea de billetes es la rendición de un personaje histórico. Se decidió, por la cara en cada billete debido a que esta era constante a través de todas las iteraciones de los billetes en cada una de sus líneas. También por la complejidad de la misma se contempló que sería una característica de mayor peso.

Después de haber elegido la cara como punto de interés era necesario extraer dicha sección de la imagen capturada. Para esto se implementó un reconocedor de rostros basado en un clasificador de Haar. Este tipo de detección se basa en el aprendizaje automático, en donde se tiene que entrenar al clasificador usando muchas imágenes positivas, con rostros, y negativas, sin rostros. OpenCV contiene una función para el reconocimiento de rostros por lo que no resulto difícil implementar este paso.

Llegado este punto era posible comenzar a pensar en un algoritmo completo. Primero se debían analizar las imágenes de referencia, estas son cada una de los rostros en los billetes, para guardar sus puntos de referencia y sus descriptores. Luego de esto se capturaría una imagen, de la cual se extraería, de ser posible, un rostro. Seguidamente se analizaría esta imagen para obtener sus propios puntos de referencia y descriptores. Por último, se compararían los puntos de referencia de la imagen capturada con todos los puntos de referencia almacenados previamente para determinar a cuál billete se apega más la imagen capturada. Con esto se obtiene un set de parejas de puntos de referencia entre dos imágenes.

Se pueden mejorar los resultados al aplicar un umbral de aceptación propuesto por D. Lowe. Lo que se busca es comparar la distancia de aceptación entre las parejas, si la distancia es demasiado grande entre los puntos se rechaza el punto de referencia (Lowe, 2004). De aquí se obtiene un set de puntos que pueden asegurar que están contenidos en la imagen de referencia. Realizando este proceso con todos los sets de parejas se podría determinar que billete se está capturando simplemente comparando la cantidad de puntos aceptados. Obviamente se busca rechazar la mayor cantidad de falsos positivos en los puntos reconocidos para evitar malos reconocimientos.

B. SELECCIÓN Y CONFIGURACIÓN DE LA PLATAFORMA

Luego que el algoritmo fuese diseñado por completo era tiempo de comenzar a pensar en la implementación del mismo. Esto requería considerar una plataforma lo suficientemente portátil para que pudiese ser integrada a un sistema completo. Debido a que el algoritmo fue diseñado en el lenguaje de programación Python se buscó una plataforma que soportara dicho lenguaje. Se decidió usar la plataforma Raspberry Pi (RPI) por su facilidad de uso y su compatibilidad nativa con el lenguaje de programación Python. Luego de esto era necesario determinar qué cámara sería usada para el proyecto, dado que era necesario que el usuario pudiese tomar fotos en cualquier momento. El hecho que se necesitara una cámara para el proyecto influyó en la elección de plataforma ya que la RPi posee un módulo externo correspondiente a un cámara, es decir, existe una cámara nativa para la RPi. Esto le dio una ventaja a la RPi dado que se sabría que esta cámara funcionaría con la plataforma de forma nativa y que no se tendrían errores de compatibilidad.

La RPi es una plataforma de desarrollo de bajo costo creada para estimular la enseñanza de los conceptos básicos de programación en las escuelas (Adafruit, 2016). Sin embargo, también resulta una plataforma versátil para todo tipo de proyectos ya que es usada por entusiastas alrededor del mundo. Existen varias iteraciones de la RPi, cinco hasta el momento. Cada una de ellas ha mejorado los componentes con el objetivo de proveer un mayor desempeño en cada iteración. Asimismo, se ha mejorado su conectividad ya que las iteraciones actuales poseen un puerto micro USB para alimentación, 4 puertos USB, un puerto RJ-45 para Ethernet y comunicación SSH, un puerto HDMI y dos hileras con 40 pines de salida de propósito general (General Purpose Input/Output o GPIO por sus siglas en inglés).

Para utilizar la RPi en conjunto con Python es necesario instalar el sistema operativo Raspbian. Raspbian es un sistema operativo de uso libre basado en Debian optimizado para el hardware de la RPi. La primera versión de Raspbian fue completada en el 2012, pero este siempre se encuentra

en constante desarrollo con el énfasis en mejorar su estabilidad y desempeño (Raspbian, 2016). Para utilizar Raspbian es necesario cargar el sistema operativo en una tarjeta micro SD. El proceso para cargar el sistema en la memoria micro SD se detalla en la página oficial de RaspberryPi.org (Raspberry Pi, 2016).

Una vez cargada la imagen en la memoria micro SD se procedió a comunicarse con la RPi usando el protocolo SSH (Secure Shell) por medio de la red de área local. El usuario y la contraseña predeterminadas para la imagen de Raspbian recién instalada es “pi” y “raspberrypi” respectivamente (sin las comillas). Estando conectado a la RPi primero era necesario actualizar el sistema operativo. Para lograr esto se utilizaron los comandos siguientes en la consola de la RPi:

- sudo apt-get update
- sudo apt-get upgrade
- sudo rpi-update

Ahora bien, para poder implementar los algoritmos en la RPi era necesario instalar las librerías requeridas, en este caso las librerías de OpenSV. Para esto primero se instalaron los paquetes y herramientas utilizadas por las librerías de OpenCV ingresado los siguientes comandos en la consola de la RPi:

- sudo apt-get install build-essential cmake pkg-config
- sudo apt-get install libjpeg8-dev libtiff4-dev libjasper-dev libpng12-dev
- sudo apt-get install libgtk2.0-dev
- sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev lib4l-dev
- sudo apt-get install libatlas-base-dev gfortran
- wget https://bootstrap.pypa.io/get-pip.py
- sudo python get-pip.py
- sudo pip install virtualenv virtualenvwrapper
- sudo rm -rf ~/.cache/pip

Habiendo correctamente instalado todos los requerimientos que OpenCV necesita ya es posible instalar e actualizar Python en la RPi. Para esta implementación se utilizó la versión 2.7 de Python en conjunto con la librería numpy, la cual se usa extensivamente en conjunto con OpenCV. Para la instalación Python se ingresaron los siguientes comandos en la terminal de la RPi:

- sudo apt-get install python2.7-dev
- pip install numpy

Una vez descargados e instalados los paquetes requeridos y Python, se procedió a descargar los archivos de OpenCV del repositorio en SourceForge.net, ingresando los siguientes comandos a la terminal de la RPi:

- `wget -O opencv-2.4.10.zip http://sourceforge.net/projects/opencvlibrary/files/opencv-unix/2.4.10/opencv-2.4.10.zip/download`

Seguidamente ya fue posible descomprimir los archivos, preparándolos para su compilación, compilarlos e instalarlos. Para esto se utilizaron los siguientes comandos en la RPi:

- `unzip opencv-2.4.10.zip`
- `cd opencv-2.4.10`
- `mkdir build`
- `cd build`
- `cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D BUILD_NEW_PYTHON_SUPPORT=ON -D INSTALL_C_EXAMPLES=ON -D INSTALL_PYTHON_EXAMPLES=ON -D BUILD_EXAMPLES=ON ..`
- `make`
- `sudo make install`
- `sudo ldconfig`

Una vez instaladas las librerías de OpenCV se obtiene la posibilidad de trasladar los algoritmos desarrollados en la computadora a la plataforma de desarrollo. Para poder ejecutar un archivo de Python en la RPi es necesario ingresar el siguiente comando a la terminal:

- `python nombre_del_archivo.py`

C. IMPLEMENTACIÓN DEL ALGORITMO

Lo primero que se realizó fue trasladar el programa con el algoritmo a la plataforma RPi ya que por el momento todo el desarrollo se había realizado en una computadora. Para trasladar archivos entre la computadora y la RPi se utilizó el software WinSCP. WinSCP es un cliente FTP de uso libre el cual su principal función es precisamente el traslado entre una computadora local y otra remota, dado que siempre se trabajó sobre una misma red de área local el traslado de archivos no conlleva ninguna complicación extra (WinSCP, 2016).

Una vez trasladado el programa se tuvieron que realizar dos modificaciones al programa. La primera fue el cambio de la cámara que se estaría usando, ya que en la computadora se estaban usando la cámara web integrada mientras que en la RPi se estaría usando la cámara nativa de la plataforma. Luego de este cambio, el programa era virtualmente el mismo. El segundo cambio correspondió al cambio en el canal de comunicación con el usuario. Para esto la comunicación con el usuario se utilizó el conveniente puerto de audio contenido en la RPi. Convenientemente la

RPi posee un puerto de audio de uso general de un tamaño de 3.55 mm, ideal para cualquier tipo de audífonos que el usuario desee utilizar.

Para que el sistema fuese completamente portátil era necesario que este fuese alimentado fuera de la energía convencional, es decir que era necesario implementar un sistema de alimentación basado en baterías para que el sistema fuese portátil. Para esto se utilizó el módulo BattBorg producido por PiBorg. El BattBorg es un regulador de energía para la RPi. Funciona con cualquier batería ente 7 y 36 V, ya que este se encarga de regular el voltaje y la corriente que necesita la RPi. Se utilizaron 8 baterías AA para alimentar a la RPi con lo cual se resolvía el problema de la portabilidad (PiBorg, 2016).

Dado que se deseaba crear un sistema completo y no solo un algoritmo corriendo sobre una plataforma se agregó funcionalidad de control al programa completo por medio de botones físicos y los pines GPIO de la RPi. Estos botones de control corresponden a la captura de la foto a analizar, y dos botones de volumen. Para completar el set de botones se tiene un interruptor para encender la plataforma y un botón para apagarla. Para encender la RPi se tiene un simple interruptor para comenzar a alimentar a la RPi. Esto es suficiente para encenderla ya que el software carga automáticamente al encender la RPi. Para apagar la RPi es necesario primero detener el sistema operativo y luego interrumpir la alimentación. Para hacer esto posible se agregó un botón que activa el proceso para cerrar el sistema operativo, y luego ya es posible abrir el interruptor de alimentación para apagar por completo la plataforma.

Por último solo se necesitaba que el programa se ejecutara al momento de encender la plataforma. Dado que la RPi es una plataforma de desarrollo y no de implementación, no hay manera nativa de ejecutar los programas al encenderla, siempre es necesario ingresar un comando a la terminal. Para lograrlo se utilizó "cron". Cron es un planificador de tareas que se ejecuta en segundo plano que permite realizar tareas en puntos o intervalos de tiempo específicos, como cuando se enciende la plataforma (Raspberry Pi, 2016). Usando cron fue posible ejecutar un archivo sh que se encargaba de ingresar el comando necesario para ejecutar el programa. De esta forma el programa se ejecutaría cada vez que la RPi fuese encendida.

D. DISEÑO Y CONSTRUCCIÓN DEL ESTUCHE

El diseño de la plataforma se basó principalmente en que tenía que contener todos los componentes de forma segura, la plataforma, la cámara, las baterías y el regulador de voltaje debían estar asegurados en el estuche. Se buscaba principalmente que el estuche fuese practico más que ergonomía por lo que el diseño general se asemeja a una caja. Un punto a tomar en cuenta fue que este estuche debía ser amigable con los no videntes.

Para realizar el diseño en 3D se utilizó Tinkercad. Tinkercad es una herramienta para el diseño y modelado 3D basad en línea. Usando formas básicas y combinándolas es posible realizar modelos complejos, además, Tinkercad ofrece exportar los archivos a los formatos correspondientes para realizar impresiones en 3D o para realizar cortes por medio de una cortadora laser (Tinkercad, 2016).

Para poder realizar el diseño se midieron todos los componentes para saber cuánto espacio se necesitaba para albergarlos. Teniendo ya todos estos datos se procedió al diseño respectivo dentro de Tinkercad. Dado que el sistema debe ser amigable con los no videntes se colocaron los símbolos en relieve de cada una de las funciones de los botones, esto para que los no videntes fuesen capaces de percibir las funciones con su sentido del tacto. Para la fabricación de las piezas se usó madera de 3.2 mm el cual fue procesado por una cortadora laser para obtener las piezas a ensamblar.

E. PRUEBAS DE LA IMPLEMENTACIÓN

Estas pruebas serían para determinar en la practica el porcentaje de acierto del algoritmo implementado en varios ambientes. Para estas pruebas se tomó una muestra de 15 billetes de las denominaciones de cinco, diez, veinte, cincuenta y cien quetzales. La razón por la cual se tuvieron más un billete era para realizar las mociones de mover y colocar el billete en la posición correcta.

Primero se realizaron capturas con ORB y cien con SIFT para cada una de las denominaciones del billete en un ambiente controlado con iluminación invariable, de esta forma se determinaría cuál sería mejor usar en los demás ambientes. Se define como ambiente controlado el espacio donde la iluminación se mantiene constante y no se necesita sostener el billete de ninguna forma.

El segundo ambiente fue al aire libre con una iluminación neutra, como lo es al aire libre bajo la sombra. El tercer ambiente fue bajo el sol. Para cada uno de estos ambientes se realizaron cien capturas y se documentó el resultado, de esta forma obtendríamos el porcentaje de acierto para cada ambiente.

F. PRUEBAS CON NO VIDENTES

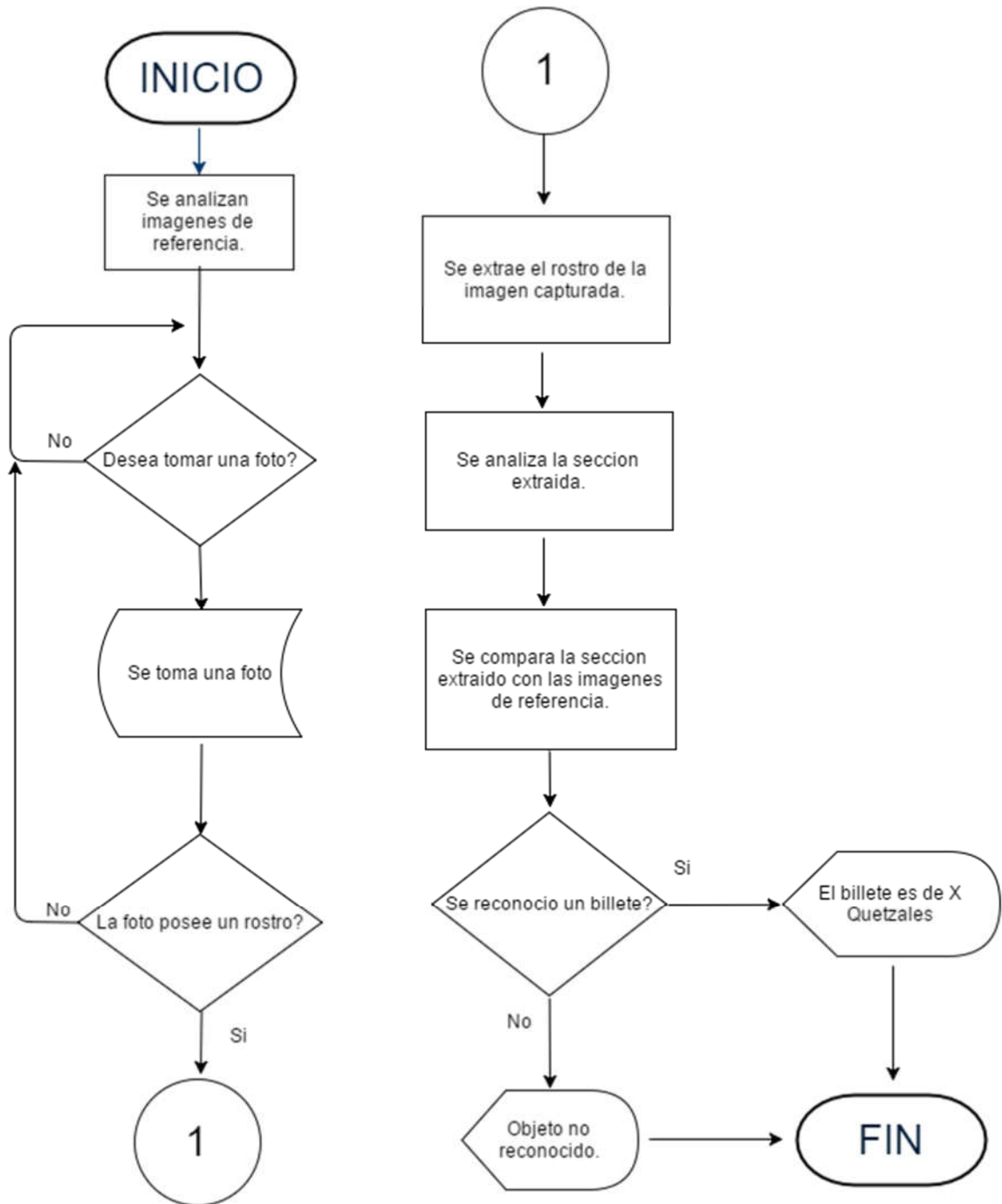
Se organizó una reunión con no videntes por medio del Benemérito Comité Pro Ciegos y Sordos de Guatemala quienes otorgaron un espacio en sus instalaciones para que los no videntes pudieran utilizar el sistema y proveer la retroalimentación deseada.

En esta reunión ya no era necesario obtener porcentajes de acierto sino más bien las opiniones del público objetivo. Para esto se tomó el sistema finalizado y se orientó a los usuarios respecto a su uso para que al finalizar pudieran proveer retroalimentación acerca del mismo.

VI. RESULTADOS

A. ALGORITMO RESULTANTE

Figura 15. Diagrama de flujo del algoritmo resultante.



B. RESULTADOS DE ANÁLISIS

Figura 16. Ejemplo de imagen de referencia.



Figura 17. Imagen de referencia analizada, cada punto verde es un punto de referencia.



Figura 18. Ejemplo de imagen capturada por medio de la cámara.



Figura 19. Análisis completo de la imagen capturada.



Figura 20. Comparación de puntos de referencia coincidentes con un billete completo.



Figura 21. Comparación de puntos de referencia con un billete completo luego de aplicar el umbral de aceptación.

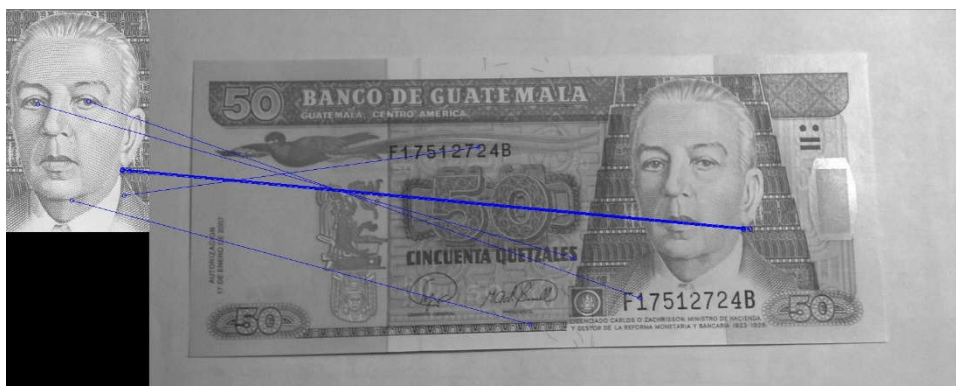


Figura 22. Extracción de rostro de la imagen capturada.



Figura 23. Rostro extraído analizado por puntos de referencia.



Figura 24. Puntos de referencia coincidentes con el rostro extraído.

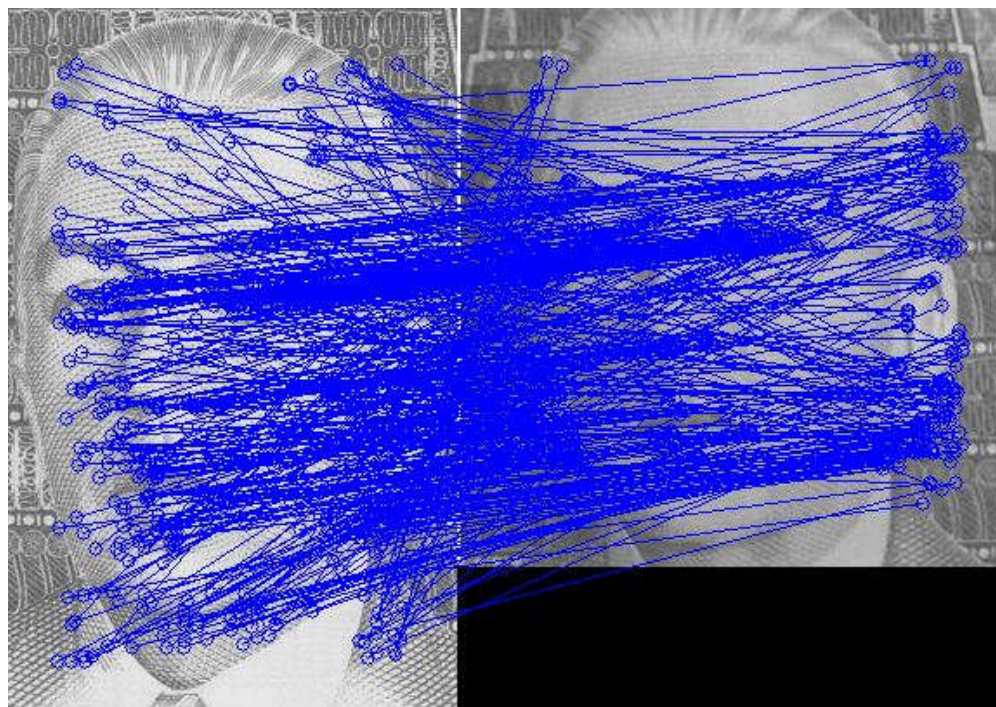
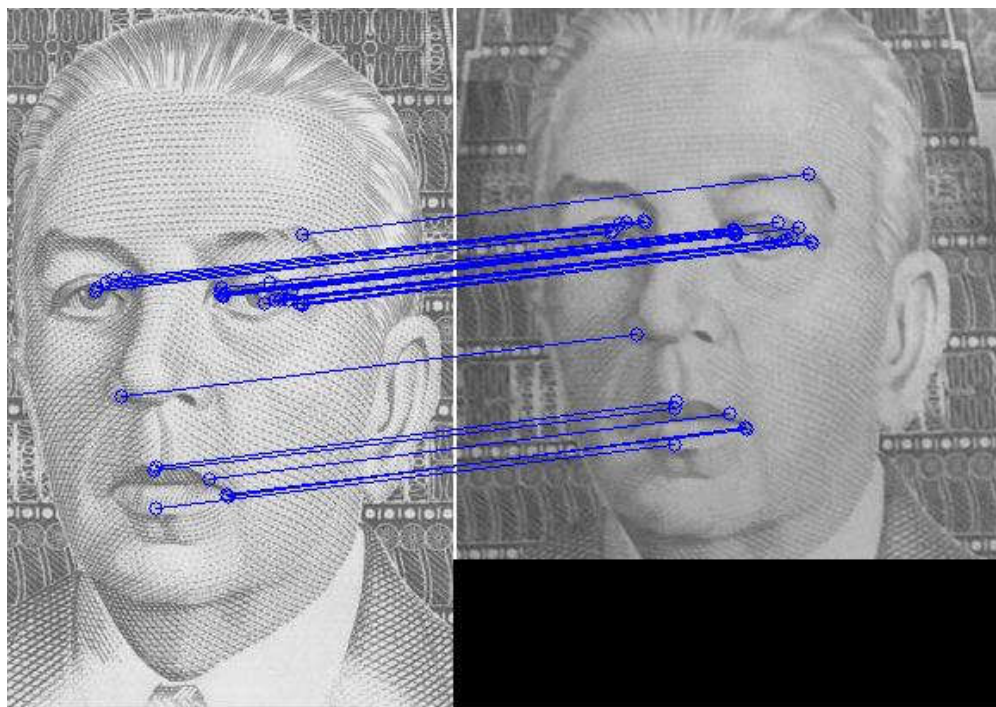


Figura 25. Puntos de referencia coincidentes con el rostro extraído luego de aplicar el umbral de aceptación.



C. ESTUCHE FABRICADO

Figura 26. Estuche fabricado a base de madera.

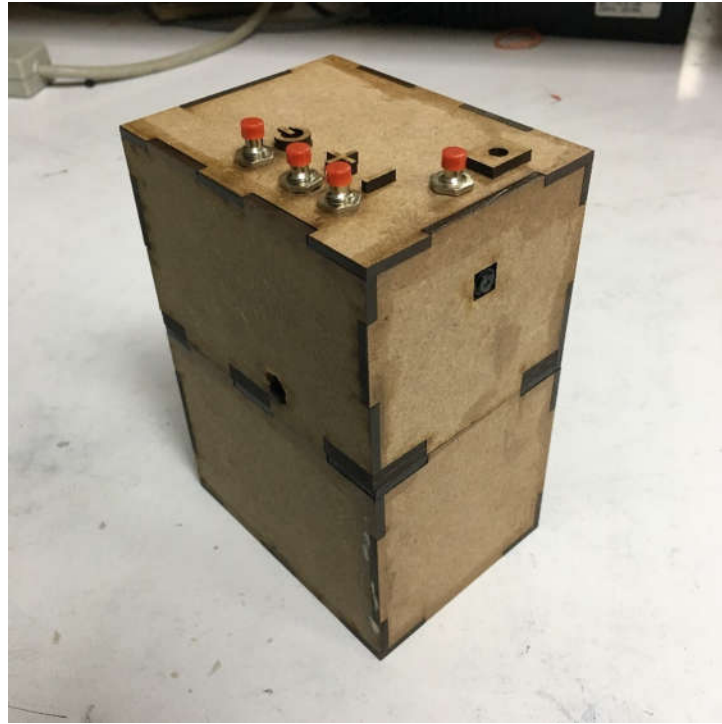


Figura 27. Detalle de relieve para denotar las funciones.



D. RESULTADOS DE PRUEBAS EN AMBIENTES DIFERENTES

Primero se realizaron capturas con ORB y cien con SIFT para cada una de las denominaciones del billete en un ambiente controlado con iluminación invariable, de esta forma se determinaría cual sería mejor usar en los demás ambientes. Se define como ambiente controlado el espacio donde la iluminación se mantiene constante y no se necesita sostener el billete de ninguna forma. El segundo ambiente fue al aire libre con una iluminación neutra, como lo es al aire libre bajo la sombra. El tercer ambiente fue bajo el sol. Para cada uno de estos ambientes se realizaron cien capturas y se documentó el resultado, de esta forma obtendríamos el porcentaje de acierto para cada ambiente.

1. Capturas en distintos ambientes

Figura 28. Captura en ambiente controlado.



Figura 29. Captura al aire libre, bajo la sombra.



Figura 30. Capturas al aire libre bajo el sol.



2. Porcentaje de acierto en un ambiente controlado

Cuadro 1. Resultados utilizando ORB en un ambiente controlado.

ORB	
Billete	% de acierto
Cinco	99.09%
Diez	78.18%
Veinte	89.09%
Cincuenta	87.28%
Cien	71.82%

Cuadro 2. Resultados utilizando SIFT en un ambiente controlado.

SIFT	
Billete	% de acierto
Cinco	99.00%
Diez	99.00%
Veinte	98.00%
Cincuenta	100.00%
Cien	95.00%

3. Porcentaje de acierto al aire libre

Cuadro 3. Porcentaje de acierto al aire libre con iluminación neutra.

Billete	% de acierto
Cinco	95.00%
Diez	92.00%
Veinte	94.00%
Cincuenta	93.00%
Cien	88.00%

4. Porcentaje de acierto bajo el Sol

Cuadro 4. Porcentaje de acierto bajo el Sol.

Billete	% de acierto
Cinco	89.00%
Diez	73.00%
Veinte	85.00%
Cincuenta	81.00%
Cien	79.00%

E. RETROALIMENTACIÓN POR PARTE DE NO VIDENTES

Por parte del Benemérito Comité Pro Ciegos y Sordos de Guatemala se tuvo la oportunidad de trabajar con el ingeniero Edgar Carrillo Reynosa y la licenciada Laida Ninett Alvarado Juarez, ambos no videntes. Se les presentó y orientó en el uso del sistema y se tomó nota de su retroalimentación para saber cómo sería posible mejorar el sistema.

Figura 31. Lic. Laida e Ing. Carrilo.



A continuación, se presentan, sin ningún orden específico las opiniones y sugerencias que los usuarios mencionaron durante la reunión:

- El sistema debe ser físicamente resistente al maltrato involuntario.
- Sería preferible una bocina integrada en lugar de audífonos.
- Es necesaria una referencia espacial para poder colocar el billete correctamente frente la cámara.
- Sería conveniente que se pudieran reconocer diferentes monedas del mundo.
- Más rapidez al reconocer el billete.

VII. ANÁLISIS DE RESULTADOS

El objetivo principal de este trabajo de graduación era proporcionar un sistema que facilitara la recepción de dinero a los no videntes por medio de la detección de la denominación de los billetes. Además del objetivo general establecido se plantearon también una serie de objetivos específicos a alcanzar. El cumplimiento de todos y cada uno de los objetivos específicos suponía a su vez el cumplimiento del objetivo general. Es por esto que se enfocó el trabajo correspondiente a cumplir cada uno de los objetivos para alcanzar el objetivo general.

El primer objetivo específico fue el diseño de un algoritmo de procesamiento de imagen para el reconocimiento de billetes. Como se especifica en la metodología, se utilizaron las librerías de OpenCV para realizar el algoritmo el cual conllevó varias modificaciones mayores. La más importante fue la modificación de extraer el rostro de los billetes. Como se puede observar en las Figuras 20 y 21 cuando no se había implementado esta modificación los resultados que se obtenían no eran muy confiables. Dado que la herramienta que mejor se puede controlar de los algoritmos de búsqueda de características es el número de dichas características que se buscan, no importaba cuántas buscáramos si los resultados iban a ser dispares. Es por esto que fue necesario extraer alguna característica del billete. Se eligió el rostro por dos razones, por su trascendencia y su complejidad. Cuando se menciona trascendencia me refiero al hecho que esta característica está presente en todas las iteraciones de todos los billetes, es decir, todos poseen el rostro de una figura histórica. Con esto nos asegurábamos que siempre podíamos buscar los mismo a través de todo el rango de billetes. La complejidad se refiere a una imagen en la que sus puntos de referencia sean los suficientemente característicos para que cuando sean encontrados en otra imagen podamos estar seguros que si coinciden.

La segunda mayor modificación en el algoritmo correspondió en la aplicación de un umbral de aceptación. Como se ven entre las Figuras 20 y 21, 24 y 25, la aplicación de un umbral de aceptación influenciaba de gran manera el resultado, pero al mismo tiempo proveía resultados más confiables. Como se ve en la Figura 20 y 24 cuando no se tiene un umbral de aceptación existen parejas de características que de hecho no coinciden, pero a nivel lógico sí. Fue hasta que se visualizaron los resultados que se pudo observar como existen características entre dos imágenes casi iguales que no coinciden, por lo tanto, se infiere que esto puede llegar a suceder entre dos imágenes diferentes, que en nuestro caso resultaría en lo que se conoce como un falso positivo. En nuestro caso un falso positivo correspondería a que el algoritmo se equivoque en la denominación del billete, el peor resultado posible. El umbral de aceptación llega a reducir las posibilidades de los falsos positivos calculando la diferencia entre las parejas coincidente, si esta diferencia resulta ser demasiado alta, la pareja se desecha. Dado que la imagen capturada es

comparada contra todas las de referencia, este umbral principalmente reduce las parejas de puntos entre la imagen capturada y las imágenes de referencia erróneas. Un punto a enfatizar es que el umbral con el que se inició no fue el final. Para calibrarlo se realizaron pruebas con billetes aleatorios, para luego por medio de una suposición educada elevar o disminuir el valor de dicho umbral hasta llegar a un punto donde se consideraba que el desempeño era óptimo.

El segundo objetivo específico consistía en la implementación de un sistema integrado portátil. Con esto se refiere a la creación del sistema en sus componentes. Al volver al sistema completamente portátil se están disminuyendo las complicaciones de uso debido a que eliminamos la variable de depender de un lugar físico para poder usar el sistema. Significa que el usuario puede utilizar el dispositivo en cualquier lugar sin depender de una fuente de corriente eléctrica.

Como tercer objetivo específico se tenía el diseño y construcción de un estuche que contuviera todos los componentes. Aquí están contenidos la plataforma de procesamiento RPI, la cámara, las baterías y su regulador de poder. El estuche puede ser visto en la Figura 26. La gran consideración que se tuvo para este objetivo es que la señalización de las funciones estuviera debidamente señalizada para que pudiera ser reconocida por medio del sentido del tacto, considerando el público objetivo. Tal y como se puede ver en la Figura 27 las funciones están señalizadas en relieve para su fácil reconocimiento por parte de los usuarios.

Como cuarto objetivo específico se tenía la realización de pruebas en distintos ambientes para obtener un porcentaje de acierto para el algoritmo. Primero se hicieron dos conjuntos de pruebas en un ambiente controlado, uno con ORB y otro con SIFT, cuyos resultados pueden ser vistos en los cuadros 1 y 2 respectivamente. Un ejemplo de las capturas realizadas puede verse en la Figura 28. Según los resultados podemos concluir la superioridad que SIFT posee sobre ORB. Los resultados de ORB no son un fracaso, pero para el problema actual, SIFT posee un mayor porcentaje de acierto. El mayor factor que impulsa los buenos resultados de SIFT es que este es invariante en la escala. Esto quiere decir que puede reconocer puntos de referencia sin importar la escala a la que este la imagen. En este caso lo que esto significa es que la distancia del billete a la cámara al hacer la captura no afecta el análisis al momento de comparar los puntos de referencia entre imágenes. Dado que no es práctico pedirle a un usuario aleatorio que sostenga el billete a una distancia específica, para nuestra aplicación es más conveniente SIFT, donde se tiene un rango de distancia en el que se puede colocar el billete para ser capturado.

En el Cuadro número 3 se puede observar el resultado de las pruebas al aire libre con iluminación neutra. Un ejemplo de las capturas realizadas en este ambiente puede ser apreciado en la Figura 29. Cuando se menciona iluminación neutra se refiere a cualquier ambiente bajo la

sombra o sin luz solar directa. Tal y como se puede notar en los resultados, se tuvo una leve caída en los porcentajes de acierto sin embargo aún son resultados aceptables.

En el Cuadro número 4 se puede observar el resultado de las pruebas al aire libre, pero estas se realizaron en presencia de luz solar. Lo primero a notar en este cuadro es la falta de uniformidad que se posee en los porcentajes de acierto. Mientras en el resto de sets de pruebas se obtuvieron porcentajes semejantes entre billetes, en este set de pruebas los porcentajes están bastante dispersos. La razón del esparcimiento de los datos es la falta de un ambiente constante, como en las pruebas anteriores. En primer set de pruebas se realizó bajo luz artificial, el segundo bajo la sombra o cuando el sol estaba detrás de las nubes. Estos dos escenarios proveen condiciones de luz lo suficientemente constantes, y los resultados lo reflejan. Bajo el Sol se obtiene más de una condición lumínica ya que este se mueve de posición y, si no colocamos el billete frente a la cámara en la misma posición en todas las pruebas, la captura se verá afectada. La variación de las capturas bajo el sol se puede ver en la Figura 30. En esta figura se puede observar el problema que se posee, en donde la cámara absorbe demasiada luz en algunos casos, lo que conlleva a que el algoritmo no sea capaz de detectar un rostro. Debido a esta variación lumínica es que se obtienen resultados dispersos. Es por esto que la mayor restricción que el sistema posee es la cámara.

El primer punto de retroalimentación, respecto a la resistencia física del sistema, es completamente comprensible debido que los no videntes no perciben los dispositivos de la misma manera, ya que no les es posible determinar la fuerza con la cual tratar las cosas por puntos tan simples como que no ven hasta donde se puede presionar un botón, por ejemplo. Asimismo, una caída del sistema no es algo improbable debido para los no videntes. El segundo punto retroalimentado fue sugerido a causa de la continua necesidad que los no videntes por parte de sus oídos, tanto que existen momento en los que si usan audífonos demasiado tiempo durante el día les puede causar molestias en los oídos.

El punto más importante es el tercero, en donde se menciona que se necesita una referencia espacial para que los no videntes puedan saber dónde colocar el billete siempre. Incluso con el gran sentido de orientación que la mayoría de no videntes posee por necesidad, pedirles que sostengan el billete en el aire sin una referencia fija se convierte en una situación en donde la probabilidad de falla aumenta, por error humano y no por problemas en el algoritmo.

Los últimos dos puntos de retroalimentación pueden considerarse como puntos de mejora ya que ambos sugieren el continuo desarrollo del proyecto. Primeramente la rapidez de reconocimiento está atada a la capacidad el hardware disponible ya que el procesamiento de

imágenes es una tarea de con altos requerimientos de recursos. En cuanto a la conveniencia de que se puedan reconocer varias monedas existen dos enfoques. El primero sugiere que el usuario pueda elegir qué moneda desea reconocer. El segundo, más realista debido al público objetivo, sugiere la posibilidad de que en todo momento se pueda reconocer cualquier moneda. El segundo enfoque es obstaculizado también por la capacidad de procesamiento de la plataforma a utilizar ya que se tendrían que realizar una mayor cantidad de comparaciones para determinar qué denominación de que moneda es el billete presentado. Por lo tanto, existe una compensación entre ambos enfoques, el primero es menos práctico, pero no requiere tantos recursos mientras que el segundo es más práctico pero requiere de más recursos.

VIII. CONCLUSIONES

Se logró diseñar un algoritmo de procesamiento de imagen que reconoce billetes por medio de la extracción de rostros de un billete y usando como base un reconocedor de características y la aplicación de un umbral de aceptación para determinar la denominación de un billete de Quetzal.

Se logró implementar el algoritmo por medio de la configuración y personalización de la plataforma Raspberry Pi. Se incluyó la implementación de un sistema de alimentación y de botones de funcionamiento por el cual el sistema se vuelve completamente portátil.

Se logró diseñar y construir un estuche orientado al uso por parte de los no videntes donde todos los componentes están contenidos para poder obtener un dispositivo completo. El material utilizado fue madera.

Según las pruebas realizadas la mayor restricción que posee el sistema es la cámara. No se obtienen resultados favorables cuando se realiza una captura con un exceso de luz.

El diseño del estuche fue abordado correctamente, pero existen temas ergonómicos en los que se puede mejorar para facilitar el uso del sistema a los no videntes.

IX. RECOMENDACIONES

Una posible mejora puede resultar al comparar el desempeño de todos los algoritmos de extracción de puntos de referencia, SIFT, SURF, BRIEF, ORB, BRISK Y BRIEF. Al ser evaluados se podría determinar el más indicado para esta aplicación.

Para optimizar el algoritmo es posible analizar la parte posterior de cada billete para incluir una imagen de referencia de la misma e incluirla en el análisis.

Para evitar el exceso de luz y para proveer una referencia espacial al usuario al momento de las capturas, se podría construir un estuche en el cual se pueda colocar el billete de tal forma que se obtengan capturas congruentes en todo tipo de ambiente.

X. BIBLIOGRAFÍA

- Adafruit. *Raspberry Pi*. https://www.adafruit.com/Raspberrypi?gclid=Cj0KEQjw1K2_BRC0s6jtgJzB-aMBEiQA-WzDMT9o5vQ2tJjqhRhOZMfAeMyOeElw77jAZ6EvA1AfyV0aAu5T8P8HAQ [24/09/16]
- Escalante, Boris. 2006. *Procesamiento digital de imágenes*. México, D.F.: Departamento de Procesamiento de Señales, UNAM. 12 págs.
- Grimaldos, Jose. 2005. *La cuestión del color*. <http://www.grimaldos.es/cursos/imgdig/color.html> [24/09/16]
- Grimaldos, Jose. 2005. *El problema de la resolución*. <http://www.grimaldos.es/cursos/imgdig/resolucion.html> [24/09/16]
- Grimaldos, Jose. 2005. *Tipos de imágenes*. <http://www.grimaldos.es/cursos/imgdig/tipos.html> [24/09/16]
- Harris, Chris; Stephens, Mike. 1988. *A combined corner and edge detector*. Reino Unido: Plessey Research Roke Manor. 6 págs.
- Instituto Nacional de Estadística. 2005. *Hallazgos de la Primera Encuesta Nacional de Discapacidad*. <https://www.ine.gob.gt/sistema/uploads/2014/01/16/U3l8MepiRtatBurbPSoTVxo56SY9pqs0.pdf> [24/09/16]
- Instituto de Tecnologías Educativas (ITE). *Conceptos básicos de imagen digital*. <http://www.ite.educacion.es/formacion/materiales/107/cd/imagen/pdf/imagen01.pdf> [24/09/16]
- Instituto de Tecnologías Educativas (ITE). *Características de la imagen digital*. http://www.ite.educacion.es/formacion/materiales/86/cd/m2/qu_es_la_imagen_digital.html [24/09/16]
- Lira, Jorge. 2010. *Tratamiento Digital de Imágenes Multiespectrales*. 2ª ed. México, D.F.: instituto de Geofísica, UNAM. 605 págs.
- Lowe, David. 2004. *Distinctive Image Features from Scale-Invariant Keypoints*. Vancouver: University of British Columbia, Computer Science Department. 28 págs.
- Lowe, David. 2004. *Patent No.: US 6711293 B1: Method And Apparatus For Identifying Scale Invariant Features In An Image And Use Of Same For Locating An Object In An Image*. 20 págs.

- National Eye Institute (NEI). *Facts About Cataract*. https://www.nei.nih.gov/health/cataract/cataract_facts [24/09/16]
- National Eye Institute (NEI). *Facts About Cataract*. https://nei.nih.gov/health/glaucoma/glaucoma_facts [24/09/16]
- National Eye Institute (NEI). *Facts About Refractive Errors*. <https://nei.nih.gov/health/errors/errors> [24/09/16]
- OpenCV. *Introduction to OpenCV-Python Tutorials*. http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_setup/py_intro/py_intro.html#intro [24/09/16]
- OpenCV. *Understanding Features*. http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_features_meaning/py_features_meaning.html#features-meaning [24/09/16]
- OpenCV. *Feature Matching*. http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_matcher/py_matcher.html#matcher [24/09/16]
- Raspbian. *Welcome to Raspbian*. <https://www.raspbian.org/> [24/09/16]
- PiBorg. *BattBorg – Power your Raspberry Pi from AA batteries*. <https://www.piborg.org/battborg> [24/09/16]
- Raspberry Pi. *Installing Operating Systems Images*. <https://www.raspberrypi.org/documentation/installation/installing-images/README.md> [24/09/16]
- Raspberry Pi. *Scheduling Tasks with Cron*. <https://www.raspberrypi.org/documentation/linux/usage/cron.md> [24/09/16]
- Rosten, Edward; Drummond, Tom. 2006. *Machine learning for high-speed corner detection*. Reino Unido: Universidad de Cambridge. 14 págs.
- Rublee, Ethan; Rabaud, Vincent; Konolige, Kurt; Bradski, Gary. 2011. *ORB: an efficient alternative to SIFT or SURF*. Menlo Park: Willow Garage. 8 págs.
- Tinkercad. *Tinkercad Features*. <https://www.tinkercad.com/about/features> [24/09/16]
- Universidad Nacional del Centro (UNICEN). *Procesamiento digital de imágenes*. <http://www.exa.unicen.edu.ar/catedras/pdi/FILES/TE/CP1.pdf> [24/09/16]
- WinSCP. *Introducing WinSCP*. <https://winscp.net/eng/docs/introduction> [24/09/16]
- World Health Organization (WHO). *Change the Definition of Blindness*. <http://www.who.int/blindness/Change%20the%20Definition%20of%20Blindness.pdf> [24/09/16]

World Health Organization (WHO). 2012. *Global Data on Visual Impairments 2010*. Geneva. 17 págs.

World Health Organization (WHO). 2014. *Ceguera y discapacidad visual*. <http://www.who.int/mediacentre/factsheets/fs282/es/> [24/09/16]

World Health Organization (WHO). 2016. *International Classification of Diseases and Related Health Problems*. 10^a rev. <http://apps.who.int/classifications/icd10/browse/2015/en> [24/09/16]