LENTES INTELIGENTES CON RECONOCIMIENTO DE IMÁGENES PARA PERSONAS INVIDENTES

Monserrat Cruz-López, Lucía Jiménez-Ruíz, Iridian L. Salinas-Lugo, Joseline Serrano-Abarca, Armida González-Lorence, Govani G. Sánchez-Orduña

Tecnológico Nacional de México/Instituto Tecnológico de San Juan del Rio,
monserrat.cruz@itsanjuan.edu.mx, lucyjimenezruiz1897@hotmail.com,
lizzere3@gmail.com,jos.serrano.aba@gmail.com, armida.lorence@itsanjuan.edu.mx,
govani.sanchez@itsanjuan.edu.mx

RESUMEN

De acuerdo con el Instituto Nacional de Estadística y Geografía (INEGI) se estima que en México cerca del 6% de la población padece algún tipo de discapacidad, de este porcentaje un poco más del 58% presenta alguna de discapacidad visual. En esta investigación se presenta la creación de un primer prototipo de los lentes mexicanos que permiten el reconocimiento de objetos en imágenes para posteriormente emitir la interpretación mediante sonido con el propósito de brindar autonomía a las personas invidentes.

Palabras clave: Reconocimiento de objetos, Persona invidente, Interpretación sonora

ABSTRAC

According to the Statistic and Geography National Institute (Instituto Nacional de Estadística y Geografía, INEGI) is estimated that in México about 6% of the population suffers some kind of disability, from this percentage a little more than 58% have some visual disability. This research presents the creation of the first prototype of Mexican lenses that allows the recognition of objects within images and interprets them using sound, these with the purpose of giving autonomy to the blind people.

Keywords: Object Recognition, Blind people, Sound interpretation

1. INTRODUCCIÓN

Nacer sin el sentido de la vista o quedar invidente por alguna enfermedad o accidente provoca una serie de problemas para las personas, en un mundo que no está preparado para los ciegos. Si bien existen sistemas especiales como el Braille, el cerebro de los invidentes logra adaptarse para que las discapacidades visuales no sean un tremendo obstáculo para la vida. La vida diaria de una persona invidente, aunque pueda parecer increíble, podría ser prácticamente igual a la de cualquier vidente, si no se empeñaran en hacer ciudades poco accesibles y poner obstáculos en aceras y calles [4].

La ciencia y las nuevas tecnologías han hecho mucho por los discapacitados en general. El exdirector de la ONCE Guadalajara, Juan Antonio Saiz, ciego desde hace 38 años por culpa de un glaucoma; navega por Internet o habla por el móvil gracias a unos programas especiales que leen la información [4], "Entre tú y yo no hay ninguna diferencia con un ordenador o un móvil, solo que tú utilizas la vista y yo el oído", explica [4].

Las personas invidentes no suelen tener la misma calidad de vida que una persona vidente y en la actualidad no existe el suficiente interés de las personas en desarrollar soluciones completas para mejorarla, con el uso de los lentes inteligentes que se presentan en este trabajo se pretende que las personas invidentes logren desplazarse de forma más autónoma, permitiéndoles mejorar su calidad de vida en los sentidos físicos y emocionales, sin tantas dificultades y teniendo así una mejor inserción en la sociedad.

Así mismo, al tener un medio de guía factible para la vida diaria, un mayor porcentaje de invidentes lograrán incursionar en el mundo laboral y así propiciará que también sus familias tengan una mejor calidad de vida.

2. TEORÍA

2.1. Las personas invidentes

2.1.1. La corteza visual adquiere funciones auditivas

En Israel, se realizó un estudio [5] en el que, se asoció ciertos sonidos con diferentes objetos, con la idea de que los reconocieran al escucharlos. Una vez que las personas invidentes lograron reconocer los sonidos, se les hizo una resonancia, durante la cual se reprodujeron los audios estudiados. Los científicos vieron cómo las zonas del cerebro normalmente asociadas a la visión trabajan para convertir el sonido en concepto o imagen [5].

2.1.2. El cerebro se estimula con la luz, pese a que no la ven

Ya sea en casos de pérdida parcial de la visión, y también en quienes son completamente invidentes, la luz tiene un efecto por sobre su cerebro. La retina, por más dañada y poco funcional que esté, igual tiene receptores de luz que, al sentir el más mínimo estimulo, alerta y pone el cerebro en acción, activando una serie de funciones cognitivas que se realizan durante el día, siguiendo el ritmo circadiano [5].

2.2. Elección del método para la identificación de objetos

2.2.1. Procesamiento de imágenes

Es un proceso complejo que requiere una serie de pasos que sus reconocimientos sucesivamente transforman los datos icónicos a información que la computadora puede reconocer [6].

El reconocimiento de patrones también llamado lectura de patrones, identificación de figuras y reconocimiento de formas consiste en el reconocimiento de patrones de señales. Los patrones se obtienen a partir de los procesos de segmentación, extracción de características y descripción donde cada objeto queda representado por una colección de descriptores. El sistema de reconocimiento debe asignar a cada objeto su categoría o clase (conjunto de entidades que comparten alguna característica que las diferencias del resto). Para poder reconocer los patrones se siguen los siguientes procesos:

- Adquisición de datos
- Extracción de características
- Toma de decisiones

El punto esencial del reconocimiento de patrones es la clasificación: se quiere clasificar una señal dependiendo de sus características. Señales, características y clases pueden ser de cualquiera forma, por ejemplo, se puede clasificar imágenes digitales de letras en las clases «A» a «Z» dependiendo de sus píxeles o se puede clasificar ruidos de cantos de los pájaros en clases de órdenes aviares dependiendo de las frecuencias [6].

2.2.2. OPENCV

OpenCV (Open Source Computer Vision) [3] es una librería de programación de código abierto dirigida principalmente a la visión por computador en tiempo real, desarrollada por la división rusa de Intel en el centro de Nizhni Nóvgorod. El uso es gratuito bajo la licencia open source BSD. La librería OpenCV es multiplataforma [3].

La librería OpenCV es una API de aproximadamente 300 funciones escritas en lenguaje C que se caracterizan por lo siguiente:

- Su uso es libre tanto para su uso comercial como no comercial No utiliza librerías numéricas externas, aunque puede hacer uso de alguna de ellas, si están disponibles, en tiempo de ejecución.
- Es compatible con The Intel Processing Library (IPL) y utiliza The Intel Integrated Performance Primitives (IPP) para mejorar su rendimiento, si están disponibles en el sistema [3].

2.2.3. PYTHON

Python es un lenguaje de programación interpretado de tipo dinámico cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma y disponible en varias plataformas [1].

Características de Python:

Interpretado: Se ejecuta sin necesidad de ser procesado por el compilador y se detectan los errores en tiempo de ejecución.

Multiplataforma: disponible para plataformas de Windows, Linux o MAC.

Gratuito: No dispone de licencia para programar.

Python contiene una gran cantidad de librerías, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas comunes sin necesidad de tener que programarlas desde cero [1].

2.3. Dispositivos hardware

2.3.1. Raspberry PI 3 B

Raspberry es una placa computadora de bajo costo desarrollada en el Reino Unido por la Fundación Raspberry pi, con el objetivo de estimular la enseñanza de la informática en las escuelas. La placa Raspberry Pi 3 modelo B es la tercera generación de Raspberry Pi y viene con una presentación impecable, dentro de su caja y envuelta en un sobre con el logo de Raspberry Pi [7].

Puede utilizarse como servidor web de bajo consumo y construirte tu propia Smart tv, como servidor de impresión y hacerte tu propia nube, como localizador GPS. Existen miles de proyectos en internet para sacarle el máximo [7].

Para que funcione, se necesita de un medio de almacenamiento (Raspberry Pi utiliza tarjetas de memoria SD o microSD dependiendo del modelo), conectarlo a la corriente utilizando cualquier cargador microUSB de al menos 1000mah para las placas antiguas y de al menos 2500mah para las modernas, y si se desea, guardarlo todo utilizando una carcasa para que todo quede a buen recaudo y su apariencia sea más estética [2].

Muchos sistemas operativos Linux han sido optimizados para la Raspberry Pi y además para darle cualquier uso a esta placa se tiene un sistema operativo muy versátil es el RASPBIAN OS que se puede descargar de la página oficial. El archivo que se descarga tiene extensión ".img" y se tiene que grabar en la tarjeta SD de la Raspberry Pi [7].

2.3.2. Cámara raspberry PI

El módulo de la cámara Raspberry Pi es un diseño personalizado para este dispositivo. Se conecta mediante un bus de datos directamente a la placa base a través de la interface dedicada CSi; diseñada especialmente para la conexión de esta cámara. La placa del módulo de la cámara consta de unas dimensiones de 25x20x9mm. Una de sus características es su poco peso (3 g), característica que la hace idónea para aplicaciones móviles donde el peso y el tamaño es algo importante. Este sería nuestro caso, por lo que la cámara se ajusta

perfectamente a nuestras necesidades. El sensor tiene una resolución de 5 megapíxeles compuesto por una lente de foco fijo. (2.592 x 1.944 píxeles imágenes estáticas y 1080p30, 720p60 y 640x480p60/90 vídeo). Por último, la cámara es compatible con la última versión de Raspbian (sistema operativo preferido de Raspberry Pi) [2].

3. DESARROLLO DEL PROTOTIPO

3.1. Algoritmo de procesamiento de imágenes

En la figura 1 y figura 2 se muestra el código del algoritmo del procesamiento de las imágenes para lograr detectar los objetos.

```
import requests
import cv2
import argparse
import numpy as np

while (True):
    ap = argparse.ArgumentParser()
    ap.add_argument('-i', '--image', required=True, help = 'path to input image')
    ap.add_argument('-c', '--config', required=True, help = 'path to yolo config file')
    ap.add_argument('-w', '--weights', required=True, help = 'path to yolo pre-trained weights')
    ap.add_argument('-cl', '--classes', required=True, help = 'path to text file containing class names')
    args = ap.parse_args()

    def get_output_layers(net):
        layer_names = net.getLayerNames()
        output_layers = [layer_names[i[0] - 1] for i in net.getUnconnectedOutLayers()]
        return output_layers

    def draw_prediction(img, class_id, confidence, x, y, x_plus_w, y_plus_h):
        label = str(classes[class_id])
        url = "http://192.168.137.153:81/objeto/5ceb74996d38c3de4e83682c/"+label
        data = {'objeto': label}
        requests.post(url = url, data = data)
        color = COLORS[class_id]
        cv2.rectangle(img, (x,y), (x_plus_w,y_plus_h), color, 2)
        cv2.putText(img, label, (x-10,y-10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)

    image = cv2.imread(args.image)
    Width = image.shape[1]
    Height = image.shape[0]
    scale = 0.00392
    classes = None
    with open(args.classes, 'r') as f:
        classes = [line.strip() for line in f.readlines()]
    COLORS = np.random.uniform(0, 255, size=(len(classes), 3))
    net = cv2.dnn.readNet(args.weights, args.config)
    blob = cv2.dnn.blobFromImage(image, scale, (416,416), (0,0,0), True, crop=False)
    net.setInput(blob)
    outs = net.forward(get_output_layers(net))
```

Figura 1. Código del algoritmo del procesamiento de imágenes, Parte I

```
    class_ids = []
    confidences = []
    boxes = []
    conf_threshold = 0.5
    nms_threshold = 0.4
    for out in outs:
        for detection in out:
            scores = detection[5:]
            class_id = np.argmax(scores)
            confidence = scores[class_id]
            if confidence > 0.5:
                center_x = int(detection[0] * Width)
                center_y = int(detection[1] * Height)
                w = int(detection[2] * Width)
                h = int(detection[3] * Height)
                x = center_x - w / 2
                y = center_y - h / 2
                class_ids.append(class_id)
                confidences.append(float(confidence))
                boxes.append([x, y, w, h])
    indices = cv2.dnn.NMSBoxes(boxes, confidences, conf_threshold, nms_threshold)
    for i in indices:
        i = i[0]
        box = boxes[i]
        x = box[0]
        y = box[1]
        w = box[2]
        h = box[3]
        draw_prediction(image, class_ids[i], confidences[i], round(x), round(y), round(x+w), round(y+h))
    cv2.imshow("object detection", image)
    cv2.imwrite("object-detection.jpg", image)
    cv2.destroyAllWindows()
```

Figura 2. Código del algoritmo del procesamiento de imágenes, Parte II

3.2. Diseño del funcionamiento del prototipo físico

Primero se captura la imagen con la cámara Raspberry Pi utilizando JavaScript, la cual es enviada a un servidor local que funciona con MongoDB y Ruby on Rails; posteriormente la imagen es procesada en el lenguaje OpenCV en conjunto con Python, para guardar el dato del objeto reconocido en una pequeña base de datos; finalmente la base de datos es consultada por la Raspberry Pi para poder emitir la interpretación del reconocimiento del objeto mediante el sonido que identifica por nombre al objeto que se encuentra en la imagen, en la figura 1. se puede apreciar en diagrama del funcionamiento.

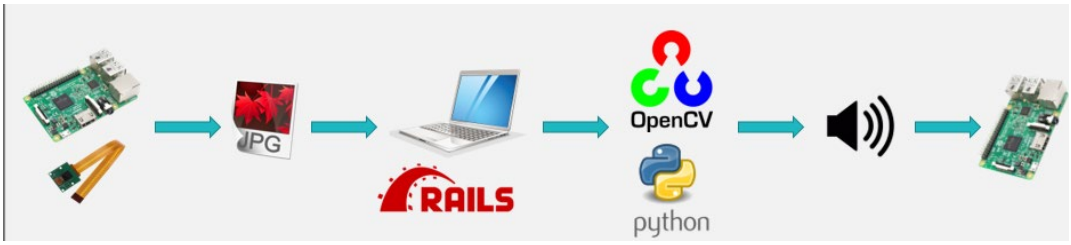


Figura 3. Diagrama del funcionamiento del prototipo físico

3.3. Funcionamiento del prototipo

De la figura 4 a la figura 8 se muestra como es el funcionamiento del prototipo por módulos.

Figura 4. Visualización del circuito para el prototipo

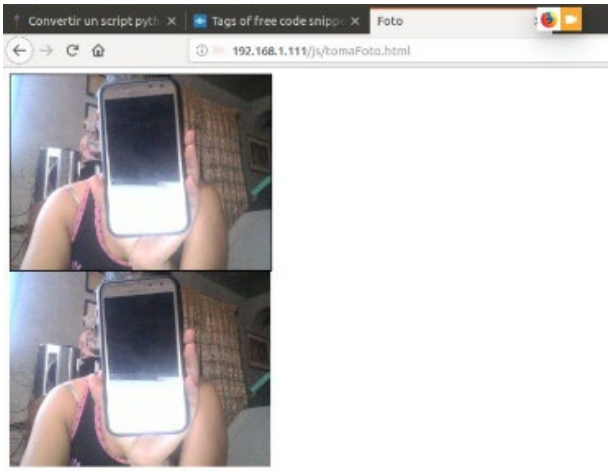


Figura 5.Captura de imagen con la cámara de la Raspberry Pi

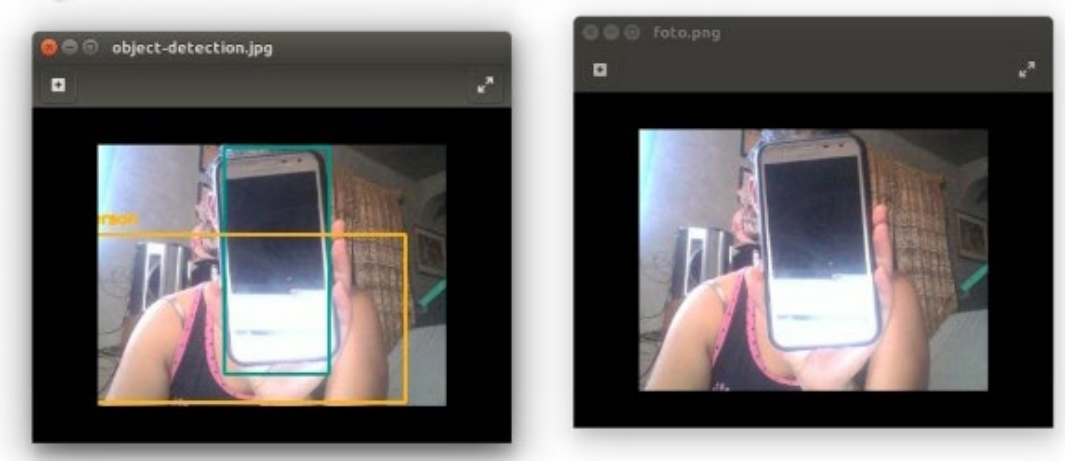


Figura 6. Comparación de las imágenes antes y después del reconocimiento con OpenCV

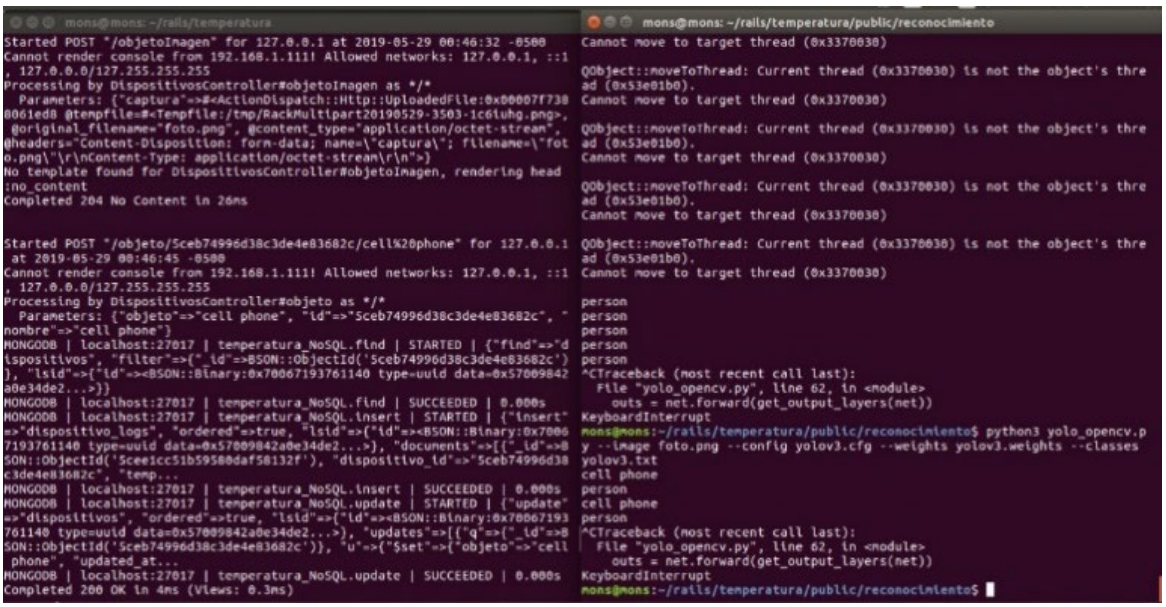


Figura 7. Envío de la imagen al servidor local (Izquierda) y detección del objeto (Derecha)

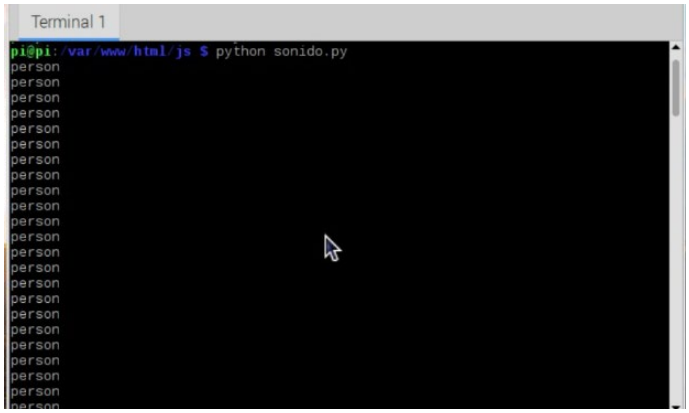


Figura 8. Visualización de la lectura para emitir el sonido en la Raspberry Pi

3.4. Visualización final del prototipo inicial

En la figura 9 y figura 10 se observa cómo se visualiza de manera física el prototipo, todo se encuentra resguardado en el sombrero, esto tomando en cuenta cuestiones de estética y para evitar burlas cuando la persona invidente los utilice.

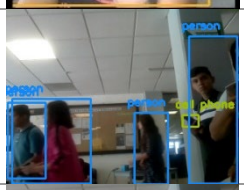
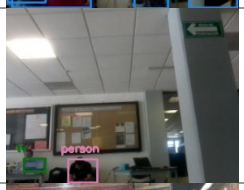
Figura 9. Visualización final del prototipo inicial

Figura 10. Visualización final del prototipo inicial, segundo ángulo

4. EXPERIMENTACIÓN

Como puede observarse en la Tabla 1, se presenta la interpretación de las imágenes logradas por el algoritmo y correspondiente interpretación en sonido a forma de traducción para las personas invidentes.

Tabla 1. Experimentación del prototipo

Experimento No.	Imagen	Sonido
1		Persona Celular
2		Persona Celular Persona Persona Persona
3		Persona Televisión
4		Silla Cama
5		Cama Celular Oso de peluche

5. CONCLUSIONES

Aun cuando se debieron hacer, por cuestiones técnicas, diversas adaptaciones, tales como: alternativas de manipulación de la Raspberry Pi, operación de recursos necesarios, realización de un puenteo a un servidor local(utilizando MongoDB y Ruby on Rails) para la captura de imágenes, envío posterior para procesamiento y regreso de a Rasberry Pi, los resultados de la presentación de esta investigación y de las pruebas del prototipo desarrollado fueron satisfactorios, cumpliendo de manera correcta la identificación e interpretación de las imágenes, lo cual representa una autonomía para la persona invidente, ya que al ser en tiempo real todo el manejo de la información, representa un gran beneficio estas personas.

El proyecto se pudo concluir en tiempo y forma de acuerdo con los estándares y factores establecidos, pero aunado a ello el prototipo puede ser enriquecido de manera significativa en el aspecto teórico y tecnológico, lo cual beneficiará de mejor manera a los invidentes y sus familias.

5. REFERENCIAS

[1] Abellán, M. Á. (25 de 05 de 2004). ¿Qué es Python? Recuperado el 27 de 02 de 2019, de https://www.programoergosum.com/cursos-online/raspberry-pi/244-iniciacion-a-python-en-raspberry-pi/que-es-python

[2] Gallego, R. S. (2015). PROYECTO FIN DE CARRERA, SISTEMA DE BAJO COSTE PARA EL SEGUIMIENTO DE PERSONAS. Leganés, Madrid: Escuela Politécnica Superior de la Universidad Carlos III de Madrid.

[3] Jonathan, G. O. (2017). Reconocimiento de los objetos utilizando OpenCV y Python. En Tesis. Recuperado el 26 de febrero de 2019

[4] Longitud. (1 de marzo de 2012). El día a día de un invidente. Obtenido de LDO: https://longitudeonda.com/index.php/el-dia-a-dia-de-un-invidente/

[5] PANORAMA.com.ve. (07 de febrero de 2015). Conoce cómo trabaja el cerebro de las personas ciegas. Obtenido de Conocer cómo trabaja el cerebro de las personas ciegas: https://www.panorama.com.ve/contenidos/2014/04/25/noticia_0026.html

[6] Reconocimiento de patrones. (28 de agosto de 2018). Recuperado el 26 de febrero de 2019, de https://es.wikipedia.org/wiki/Reconocimiento_de_patrones

[7] Ro-botica. (2017). Ro-botica Global S.L, Robótica Educativa & Personal. Obtenido de Raspberry Pi 3 modelo B: https://www.ro-botica.com/Producto/RASPBERRY-PI-3-MODELO-B