

PROTOTIPO DE DETECCIÓN DE APARCAMIENTOS LIBRES MEDIANTE VISION ARTIFICIAL EN UN PARQUEADERO DE LA UNIVERSIDAD TECNICA DEL NORTE

Cristian Erazo, Sandra Narváez

Universidad Técnica del Norte-Ibarra, Ecuador
caerazoe@utn.edu.ec, sknarveaz@utn.edu.ec

Resumen- El Sistema de Monitoreo de Disponibilidad de plazas de estacionamiento en la Universidad Técnica del Norte, está basado en el uso de técnicas de visión artificial para lo cual se usarán cámaras para recolectar información sobre plazas libres de estacionamiento. Para esto se realizó un estudio de la situación actual en cuanto al uso y distribución de plazas de estacionar en la Universidad Técnica del Norte, así como también analizar las ventajas y falencias que este presenta. El sistema cuenta con una cámara ubicada estratégicamente, la cual envía la información hacia una computadora de placa simple que permita el procesamiento de imágenes mediante visión artificial, estos datos serán subidos a la nube para que puedan ser visualizados por los usuarios mediante una página web y también en dispositivos móviles con sistema operativo Android.

I. INTRODUCCIÓN

La congestión del tráfico causada por el vehículo es un problema alarmante a escala global y ha estado creciendo de manera exponencial. El problema del estacionamiento de automóviles es un contribuyente importante y ha sido, aún un problema importante con el aumento del tamaño del vehículo en el segmento de lujo y los espacios de estacionamiento confinados en ciudades urbanas. Buscar una plaza de aparcamiento es una actividad de rutina para muchas personas en ciudades de todo el mundo. Esta búsqueda quema alrededor de un millón de barriles de petróleo del mundo todos los días. A medida que la población mundial continúa urbanizando, sin un retiro del automóvil bien planificado y orientado a la conveniencia, estos problemas empeorarán [1].

El problema es que los estacionamientos simplemente se han ignorado ya que no se han implementado tecnologías que permitan mejorar la eficiencia de estos, a partir de ello se genera la pérdida de tiempo en los conductores cuando buscan un lugar libre. También se generan situaciones peligrosas ya que mientras el conductor se enfoca en la búsqueda de un lugar libre se distrae de lo que pasa a su alrededor lo que aumenta la probabilidad de que cause un accidente.

Con el despliegue de dispositivos de Internet de las cosas (IoT) y de máquina a máquina (M2M) como sensores y monitores, los datos de estacionamiento se pueden enviar a través de redes celulares en ciudades para que la información se entregue de manera confiable y en tiempo real. Se están desarrollando aplicaciones para permitir la gestión de datos y

aprovechar el análisis para nuevos usos. Con las nuevas aplicaciones de estacionamiento inteligente, los conductores pueden recibir información para encontrar espacios de estacionamiento vacíos, los trabajadores pueden administrar las instalaciones de estacionamiento y se pueden mejorar las colecciones, para nombrar algunos usos nuevos [2].

II. REVISIÓN BIBLIOGRÁFICA

A. Parqueaderos Inteligentes

Los sistemas de estacionamiento inteligente generalmente obtienen información sobre los espacios de estacionamiento disponibles en un área geográfica particular y el proceso se realiza en tiempo real para colocar los vehículos en las posiciones disponibles. Implica el uso de sensores de bajo costo, la recopilación de datos en tiempo real y el pago automático habilitado para teléfonos móviles los sistemas que permiten a las personas reservar el estacionamiento por adelantado o predecir con gran precisión dónde encontrarán un lugar. Cuando se implementa como un sistema, el estacionamiento inteligente reduce las emisiones de automóviles en los centros urbanos al reducir la necesidad de que las personas circulen innecesariamente las cuadras de la ciudad en busca de estacionamiento. También permite a las ciudades administrar cuidadosamente su suministro de estacionamiento. El estacionamiento inteligente ayuda a uno de los mayores problemas de conducción en áreas urbanas; Encontrar espacios de estacionamiento vacíos y controlar el estacionamiento ilegal [3].

B. Internet of Things IoT

El IoT revoluciona Internet, ya que no solo las computadoras se están conectando, sino también las cosas físicas. El IoT puede así proporcionarnos datos sobre nuestro entorno físico, con un nivel de detalle nunca antes conocido en la historia humana. La comprensión de los datos generados puede generar una mejor comprensión de nosotros mismos y del mundo en que vivimos, creando oportunidades para mejorar nuestra forma de vivir, aprender, trabajar y entretener. Especialmente la combinación de datos de muchas fuentes diferentes y su análisis automatizado puede proporcionar nuevos conocimientos sobre las relaciones e interacciones

existentes entre las entidades físicas, su entorno y los usuarios. Esto facilita optimizar su comportamiento. La automatización de la interacción entre el análisis y el control de datos puede llevar a nuevos tipos de aplicaciones que utilizan bucles de optimización completamente autónomos [4].

C. Visión Artificial

La visión artificial se puede definir como un campo científico que extrae información de imágenes digitales. El tipo de información obtenida de una imagen puede variar según la identificación, las mediciones de espacio para la navegación o las aplicaciones de realidad [5]. Otra forma de definir la visión por ordenador es a través de sus aplicaciones. La visión artificial está creando algoritmos que pueden comprender el contenido de las imágenes y utilizarlo para otras aplicaciones. Los orígenes de la visión por computadora se remontan a un proyecto de verano de licenciatura del MIT en 1966. En ese momento se creía que la visión por computadora podía resolverse en un verano, pero ahora tenemos un campo científico de 50 años que aún está lejos de resolverse [5].

D. Procesamiento de Imágenes

Las imágenes matriciales también llamadas imágenes de área se representan mediante una matriz donde cada uno de sus elementos tiene la información de intensidad si la imagen es en escala de grises y el contenido de colores rojo R, verde G y azul B si la imagen es a color (RGB). El procesamiento de la imagen consiste en aplicar a la matriz una técnica o algoritmo que realice cambios que permita segmentar las partes de la imagen que son relevantes para el reconocimiento de patrones propios de la aplicación [6].

E. Raspberry Pi 3

Raspberry Pi, es una tarjeta electrónica de computador del tamaño de tarjeta de crédito que se le conecta una pantalla o monitor y un teclado. Soporta diversos componentes propios del computador personal (PC). Tiene la capacidad de utilizar software que se utilizan en un PC de escritorio o en una laptop, tales como: como hojas de cálculo, procesadores de texto y juegos. También reproduce vídeo de alta definición. Desde el año 2012 hasta la fecha actual han introducido al mercado nuevas versiones cada una con mejoras importantes, sin incrementos de coste significativos [7]. A continuación, se muestra la placa Raspberry Pi en la Figura 1.

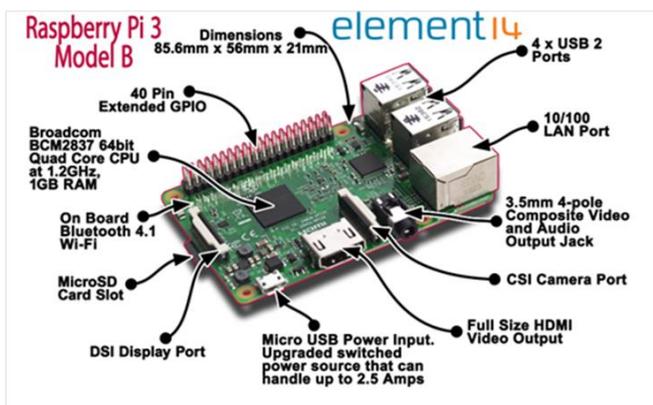


Figura 1. Raspberry Pi Model B

F. OpenCV

OpenCV (Open Source Computer Vision Library) se publica bajo una licencia BSD y, por lo tanto, es gratis para uso académico y comercial. Tiene interfaces C++, Python y Java y es compatible con Windows, Linux, Mac OS, iOS y Android. OpenCV fue diseñado para la eficiencia computacional y con un fuerte enfoque en las aplicaciones en tiempo real. Escrita en C/C++ optimizado, la biblioteca puede aprovechar el procesamiento multi-core. Habilitado con OpenCL, puede aprovechar la aceleración de hardware de la plataforma de cómputo heterogénea subyacente [8].

Adoptado en todo el mundo, OpenCV tiene más de 47 mil personas de la comunidad de usuarios y el número estimado de descargas supera los 14 millones. Los rangos de uso van desde arte interactivo, hasta inspección de minas, mapas de costura en la web o mediante robótica avanzada [8].

III. DESARROLLO EXPERIMENTAL

A. Diagrama de Bloques

El diagrama que se presenta a continuación es el que comprende las fases de diseño del sistema, está formado por 4 bloques los cuales a su vez están contienen varios subprocesos, se han planteado los bloques dependiendo de las funciones que cada uno debe desarrollar.

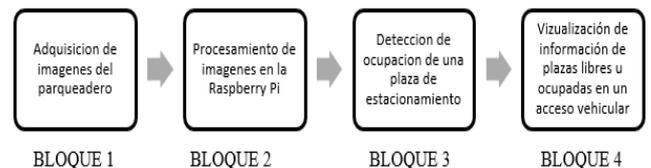


Figura 2. Diagrama de bloques.

Para dar inicio con la adquisición de datos se inicia con el bloque 1, este es el encargado de que la cámara se inicie y comience a adquirir imágenes. En el bloque 2 se inicia el trabajo de OpenCV, aquí se realiza un preprocesamiento a las imágenes adquiridas en el bloque anterior, además en este bloque se añaden los ROI. Las regiones ROI, de sus siglas en inglés "Region of Interest", son funciones que permiten enfocarnos específicamente en un lugar de la imagen para sacar de éste sus características más importantes. La ROI determinada para nuestra imagen permite enfocar un lugar de estacionamiento y determinar si existe algún objeto ocupándolo. Con los datos obtenidos en el bloque 2 y bloque 3 se realiza el proceso de análisis que permite determinar si hay plazas de estacionamiento libres. Finalmente, los datos del bloque 3 deben ser visualizados en un acceso vehicular.

B. Adquisición de Imágenes

Gracias a OpenCV se puede capturar la transmisión de video en vivo mediante el uso de una interfaz muy simple. Para realizar la captura es necesario crear una función denominada VideoCapture, su parámetro será el nombre de un archivo de video o el número de las cámaras que están conectadas a la placa que generalmente empieza desde 0.

A continuación, se muestra la función completa utilizada para la captura de video:

cap = cv2.VideoCapture(número de la cámara o nombre del archivo de video)

C. Regiones de Interés (ROI)

Para asignar los ROI a la captura de video se utilizó un formato de serialización de datos conocido como YAML (YAML Ain't Markup Language), este es un lenguaje más comprensible para personas, se usa particularmente para archivos de configuración, traducciones y representación de información. La importancia de este formato reside en que es amigable, fácil y permite el mapeo de estructura de datos.

Se usó YAML como archivo de configuración para los cuadriláteros, en el archivo de configuración se incluirán las coordenadas tanto en X como en Y que representarán los vértices del ROI. En la Figura 3 se observa el proceso de asignación de ROI.

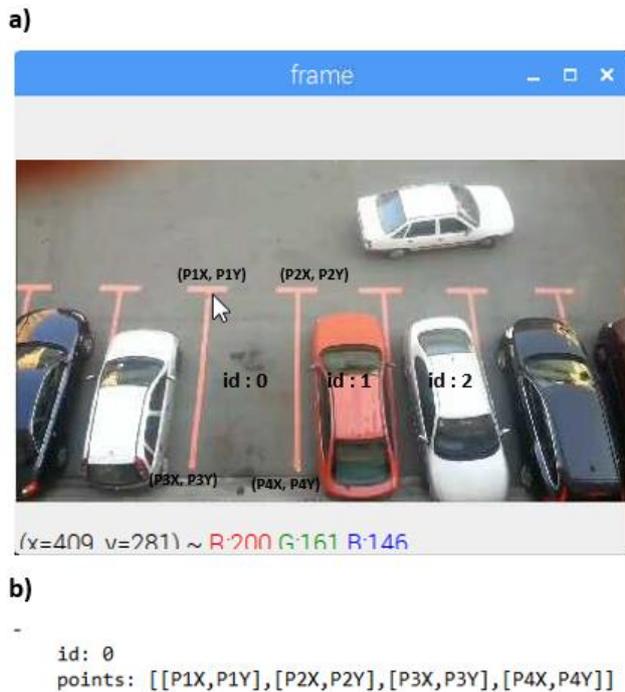


Figura 3. Asignación de ROI

D. Conversión a escala de grises

Una de las razones para usar imágenes en escala de grises en lugar de imágenes a todo color es simplificar el espacio de color tridimensional (R, G y B) a una representación dimensional única, es decir, monocromática. Este hecho es importante porque reduce el costo computacional y simplifica los algoritmos [9]. En la Figura 4 se puede observar la conversión a escala de grises del parqueadero.

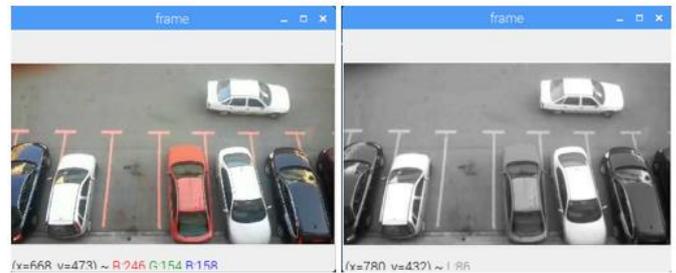


Figura 3. Conversión a escala de grises.

E. Suavizado de imagen

El objetivo principal del suavizado de imágenes es eliminar el ruido en las imágenes digitales. La calidad de la imagen es un factor importante para el punto de vista de la visión humana. El suavizado de imágenes es un método para mejorar la calidad de las imágenes, la imagen generalmente tiene un ruido que no se elimina fácilmente en el procesamiento de la imagen [10]. La calidad de la imagen se ve afectada por la presencia de ruido. En la Figura 4 se puede observar el suavizado de imagen aplicado en el parqueadero.

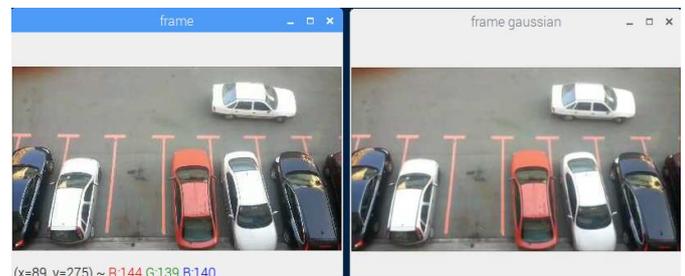


Figura 4. Suavizado de imagen.

F. Resta de Fondo

La sustracción de fondo es un método ampliamente utilizado para detectar objetos en movimiento en videos de cámaras estáticas. El fundamento del enfoque es el de detectar los objetos en movimiento a partir de la diferencia entre el marco actual y el marco de referencia, a menudo llamado "imagen de fondo" o "modelo de fondo" [11]. En la Figura 5 se puede observar la resta de fondo realizada en el parqueadero.

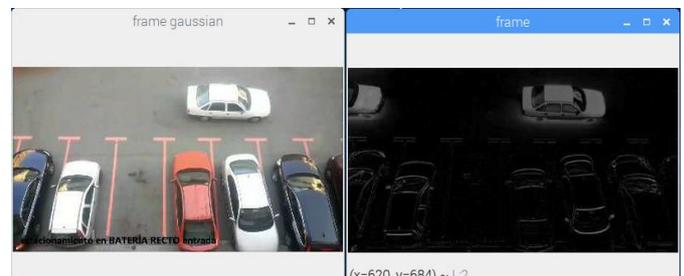


Figura 4. Resta de fondo.

G. Detección de Bordes

El proceso de detección de bordes sirve para simplificar el análisis de imágenes al reducir drásticamente la cantidad de datos a procesar y al mismo tiempo preservar información estructural útil sobre los límites de los objetos. La detección de bordes apunta a identificar puntos en una imagen digital en los

que el brillo de la imagen cambia bruscamente. Esta técnica trata principalmente de extraer bordes en una imagen identificando píxeles donde la variación de intensidad es muy alta [12].

Para este caso utilizaremos el operador Laplaciano que utiliza la segunda derivada para la detección de bordes, puesto que las imágenes que se están tratando son en 2D se debe considerar la derivada en ambas direcciones, esto hace que el operador Laplaciano sea útil en estos casos. En la Figura 5 se puede observar la detección de bordes mediante el operador Laplaciano.

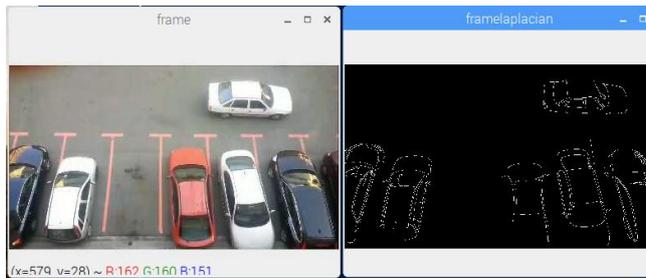


Figura 5. Detección de bordes.

H. Algoritmo de detección.

El siguiente punto es detectar si una plaza de estacionamiento está libre u ocupada, básicamente el algoritmo se basa en detectar la presencia de bordes, es decir que se obtendrá ciertos valores cuando la plaza de estacionamiento este vacío y se obtendrá otros cuando este ocupada por un auto.

Primeramente, debemos obtener valores numéricos para poder establecer un umbral, hay que tener en cuenta que una imagen es una matriz con varios valores, dicho esto se debe realizar un proceso que permita obtener un único valor numérico.

Mediante la librería matemática Numpy se puede obtener la media aritmética de una matriz utilizando la función `mean()`. Además se utilizó la función `abs()` para obtener valores absolutos en el cálculo de la media aritmética. A, continuación se muestra un ejemplo del uso de estas funciones.

La media aritmética a obtener será la de la variable en donde se ha almacenado el cálculo del operador Laplaciano que para este caso es la variable `laplacian`. Esta media aritmética será almacenada en la variable `delta`, hay que destacar que este cálculo se realiza para cada ROI que se ha creado.

A continuación, en la Figura 6 se muestran los valores de las medias aritméticas que cada ROI arroja.



Figura 6. Media aritmética de cada ROI.

Se puede establecer un umbral de valor 1, esto quiere decir que cuando exista valores que sobrepasen el umbral existirá presencia de vehículos y si este valor es menor al umbral indicara que el puesto está vacío.

Para realizar este proceso se utilizará una comparación básica que se muestra en la siguiente línea de código:

```
status = delta < 1
```

Se almacenó el estado en la variable `status`, esta comparación arrojará un valor lógico `TRUE` si `delta` es menor que 1 y un valor `FALSE` si `delta` es mayor que 1, en la Figura 7 se puede observar un ejemplo de este proceso



Figura 7. Detección de plazas.

A manera de demostración se puede dibujar los ROI y asignarles color dependiendo de su estado, en la Figura 8 se puede observar los ROI asignados a cada plaza de estacionar, se ha asignado el color verde para las plazas libre y el color rojo para las que están ocupadas.

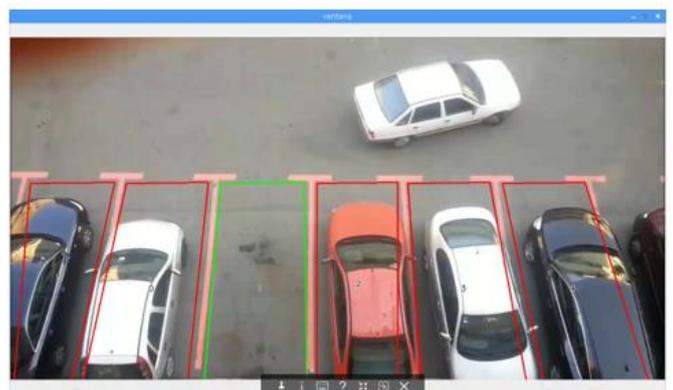


Figura 8. Detección de plazas.

I. Plataforma de visualización

La siguiente fase de nuestro prototipo es la visualización de los datos obtenidos en el proceso de detección, dicho esto se ha utilizado la plataforma Ubidots que es muy conocida dentro de la industria de hardware, software e ingeniería integrada dentro del ecosistema de IoT.

La ventaja de esta plataforma es su fácil uso y manejo en lo que respecta al almacenamiento y visualización de datos en la nube, además proporciona versiones de prueba gratuita, por lo que lo hace ideal en la implementación de este prototipo como plataforma de visualización [13].

Se diseñó una página WEB para mejorar la representación de información en la Figura 9,10 y 11 se puede observar la página web.



Figura 9. Página Inicial



Figura 10. Página informativa sobre lugares disponibles.



Figura 11. Página informativa estado de cada puesto.

J. Diagrama circuital.

Básicamente el prototipo cuenta con la placa Raspberry Pi, la cámara web que estará conectada al puerto USB de la placa, la fuente de energía que es un adaptador de 5V 2A que se conecta a la placa a través de una conexión micro USB tipo B y finalmente tendrá conexión WIFI para tener acceso a Internet. En la figura 12 se puede observar los componentes del prototipo.

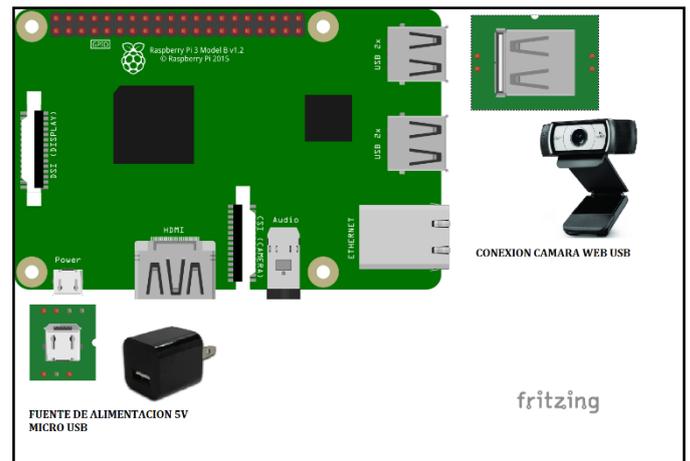


Figura 12. Diagrama circuital.

K. Pruebas de funcionamiento

En esta fase se realiza el análisis del desempeño y rendimiento del prototipo, las pruebas fueron realizadas en el parqueadero ubicado frente al edificio de la FECYT de la UTN y en varios horarios del periodo diurno.

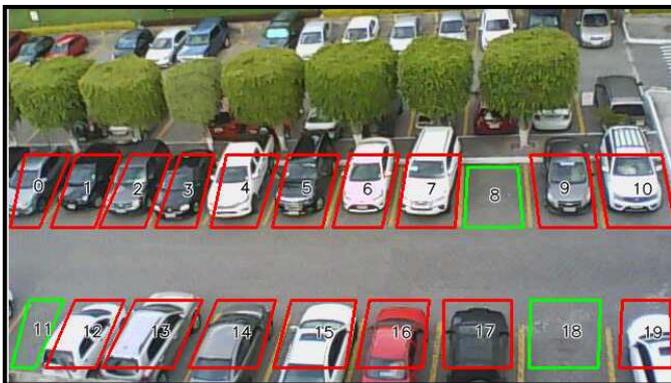
Una vez que se han asignado los ROI se procede a calibrar el umbral, para esto se ejecuta el script y se observa los valores arrojados por cada ROI, en la Figura 13 se observa los valores de las media aritméticas de cada ROI.



Figura 12. Diagrama circuital.

Se puede observar que en los ROI 8, 11 y 18 tenemos valores de 1.35, 0.89, 1.45 respectivamente, se usa estos valores como referencia para establecer el umbral, dicho esto se establecerá el umbral en un valor de 1.50.

Se ejecuta de nuevo el script y en la Figura 13 se puede observar el resultado obtenido.



IV. CONCLUSIONES

Este trabajo presenta un enfoque alternativo para tratar el problema expuesto en los parqueaderos, este está basado en el uso de técnicas de visión artificial, la base fundamental es la detección de bordes en cada lugar de estacionamiento mediante esta técnica podemos discretar el estado ocupado o libre de este sitio.

El uso de visión artificial en este tipo de aplicaciones permite reducir el número de nodos sensores lo cual se ve reflejado en la reducción de costos y también en la reducción de consumo energético, dicho esto este tipo de tecnologías es muy viable para este tipo de aplicaciones.

V. Referencias

- [1] A. Basu, «SMART PARKING,» 2014. [En línea]. Available: <https://www.happiestminds.com/whitepapers/smart-parking.pdf>.
- [2] Aeris Communications, Inc., «Introduction to Smart Parking,» *AERIS*, 2015.
- [3] M. Fraifer y M. Fernström, «Investigation of Smart Parking Systems and their technologies,» *Thirty Seventh International Conference on Information Systems*, 2016.
- [4] A. Sayed y A. Mohamed, «Internet of Things Applications, Challenges and Related Future Technologies. world scientific news.,» *World Scientific News*, pp. 127-135, 2017.
- [5] R. Krishna, *COMPUTER VISION FOUNDATIONS AND APPLICATIONS*, Stanford: Stanford University, 2017.
- [6] M. Martin, *PROCESAMIENTO DIGITAL DE IMAGENES*, Mexico: Universidad Autonoma de Puebla, 2013.
- [7] Raspberry, «Raspberry,» 2018. [En línea]. Available: <https://www.raspberrypi.org/>.
- [8] OpenCV , «OpenCV,» 2018. [En línea]. Available: <https://opencv.org/>.
- [9] S. Macedo, G. Melo y J. Kelner, «A comparative study of grayscale conversion techniques applied to SIFT descriptors,» *SBC Journal on Interactive Systems*, pp. 30-35, 2015.
- [10] A. Goyal, A. Bijalwan y K. Chowdhury, «A Comprehensive Review of Image Smoothing Techniques,» *International Journal of Advanced Research in Computer Engineering & Technology*, pp. 315-320, 2012.
- [11] J. & K. J. & S. V. & J. P.-M. Mchugh, «Foreground-Adaptive Background Subtraction.,» *Signal Processing Letters*, vol. 16, pp. 390-393, 2009.
- [12] S. Savant, «A Review on Edge Detection Techniques for Image Segmentation,» *International Journal of Computer Science and Information Technologies*, pp. 5898-5910, 2014.
- [13] Ubidots, «Ubidots,» 2018. [En línea]. Available: <https://ubidots.com>.