Diseño e implementación de un sistema de seguridad y control de aparcamiento mediante el uso de cámaras para la Universidad de las Fuerzas Armadas ESPE-L aplicando realidad aumentada.

Anasi Nasimba, Danny Eduardo

Departamento de Eléctrica y Electrónica

Carrera de Ingeniería en Electrónica e Instrumentación

Trabajo de titulación, previo a la obtención del título de Ingeniero en Electrónica e Instrumentación

Ing. Galarza Zambrano, Eddie Egberto

8 de noviembre del 2021

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERIA EN ELECTRÓNICA E INSTRUMENTACIÓN

CERTIFICACIÓN

Certifico que el trabajo de titulación, ***"Diseño e implementación de un sistema de seguridad y control de aparcamiento mediante el uso de cámaras para la Universidad de las Fuerzas Armadas ESPE-L aplicando realidad aumentada"*** realizado por el señor **Anasi Nasimba, Danny Eduardo**, el mismo que ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga, 8 de noviembre del 2021

Firma:

Firmado digitalmente por EDDIE EGBERTO GALARZA ZAMBRANO

Ing. Galarza Zambrano, Eddie Egberto

C. C: 1303128514

REPORTE DEL URKUND

Curiginal

Document Information

Analyzed document	Tesis-Sistema de Seguridad y gestionamiento de Plazas de Estacionamiento.docx (D112592463)
Submitted	9/14/2021 10:43:00 PM
Submitted by	Juan Carlos Altamirano
Submitter email	jc.altamiranoc@uta.edu.ec
Similarity	2%
Analysis address	jc.altamiranoc.uta@analysis.urkund.com

Sources included in the report

W	URL: https://repositorio.espe.edu.ec/bitstream/21000/11976/1/T-ESPEL-ENI-0377.pdf Fetched: 1/5/2021 4:02:47 AM	1
W	URL: https://docplayer.es/amp/39685831-Departamento-de-electrica-y-electronica.html Fetched: 3/18/2020 7:50:18 PM	1
W	URL: https://docplayer.es/45129311-Departamento-de-electrica-y-electronica-carrera-de-ingenieria-en-electronica-e-instrumentacion.html Fetched: 7/23/2021 8:53:17 PM	1
W	URL: https://core.ac.uk/download/pdf/154795857.pdf Fetched: 7/7/2020 7:30:55 AM	1
W	URL: https://dspace.unl.edu.ec/jspui/bitstream/123456789/11588/1/Aponte%20Rueda%2C%20Pedro%20Fernando%2C%20Armijos%20Armijos%2C%20Andr%C3%A9s%20Armanado.pdf Fetched: 7/23/2021 7:54:25 PM	1
SA	**Tesis_C_Manosalvas.docx** Document Tesis_C_Manosalvas.docx (D29386767)	1
SA	**Tesis Danilo Pantoja.docx** Document Tesis Danilo Pantoja.docx (D77340457)	2
W	URL: http://repositorio.uisrael.edu.ec/bitstream/47000/2441/1/UISRAEL-EC-ELDT-378.242-2020-026.pdf Fetched: 3/29/2021 8:18:21 AM	1
W	URL: https://www.redatel.net/html/camara-ip-conceptos-basicos.html#:~:text=Una%20c%C3%A1mara%20IP%20consiste%20principalmente,red%20para%20transmisi%C3%B3n%20de%20datos.Redecker, Fetched: 9/14/2021 10:44:00 PM	1
W	URL: https://www.researchgate.net/publication/283861537_Utilizacion_de_metodos_de_vision_artificial_para_PC_como_apoyo_en_la_automocion Fetched: 6/6/2020 7:21:15 AM	1
SA	**TesisPrototipoParqueosUG.docx** Document TesisPrototipoParqueosUG.docx (D64960587)	1

Curiginal

URL: https://1library.co/document/y8grgn2z-diseno-prototipo-sistema-temprana-inundacion-desbordamiento-artificial-celulares.html
Fetched: 5/26/2021 2:01:16 AM

1

Latacunga, 15 de septiembre de 2021

Firmado digitalmente por EDDIE EGBERTO GALARZA ZAMBRANO

Ing. Eddie Galarza Zambrano
Director del Proyecto

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN

RESPONSABILIDAD DE AUTORÍA

Yo, **Anasi Nasimba, Danny Eduardo**, con cédula de ciudadanía N°1720535382, declaro que el contenido, ideas y criterios del trabajo de titulación: **Diseño e implementación de un sistema de seguridad y control de aparcamiento mediante el uso de cámaras para la Universidad de las Fuerzas Armadas ESPE-L aplicando realidad aumentada** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 08 de noviembre del 2021

Firma:

Danny Eduardo, Anasi Nasimba
C.C.: 1720535382

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN

AUTORIZACIÓN DE PUBLICACIÓN

Yo, **Anasi Nasimba, Danny Eduardo**, con cédula de ciudadanía N° 1720535382, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: Diseño e implementación de un sistema de seguridad y control de aparcamiento mediante el uso de cámaras para la Universidad de las Fuerzas Armadas ESPE-L aplicando realidad aumentada en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Latacunga, 08 de noviembre del 2021

Firma:

Danny Eduardo, Anasi Nasimba
C.C.: 1720535382

DEDICATORIA

Dedico este trabajo a Dios, por darme la fuerza, el conocimiento y guiarme en el camino para cumplir esta meta.

A mis padres quienes han sido el pilar fundamental quienes, con su dedicación, esfuerzo y amor han sido parte de este logro, a mis hermanos quien me han apoyado incondicionalmente en todo momento y a las personas que me brindaron su mano en momentos difíciles que me llenaron de fortaleza para seguir a lo largo de la carrera y en el desarrollo del proyecto para concluirlo de manera satisfactoria y alcanzar esta meta.

Agradezco de igual manera al ingeniero Eddie Galarza por su apoyo y guía brindada a lo largo de este proyecto de investigación desde su iniciación hasta su culminación, además de ayudarme a formar académicamente, gracias también a los maestros que supieron brindar su conocimiento técnico por medio de sus consejos y experiencias.

Danny Eduardo Anasi Nasimba

AGRADECIMIENTO

Agradezco a Dios por estar siempre a mi lado cuidándome, bendiciéndome y brindándome la fortaleza para afrontar las dificultades que me han permitido cumplir mis metas.

Agradezco a mis padres Leonardo y Mercedes, que han sido un constante ejemplo de trabajo, esfuerzo y lucha, gracias por nunca perder la fe en mí y brindarme su apoyo en todo momento, a pesar de las dificultades mantener la familia unida buscando siempre la superación profesional para sus hijos.

Agradezco a mi director de tesis Ing. Eddy Galarza por confiar en mí, por guiarme y brindarme sus conocimientos para cumplir con esta meta.

Agradezco a todas las personas, que de una u otra forma, han estado presentes en mi vida, y han deseado el bien para mí y mi familia, gracias por sus palabras y consejos.

Danny Eduardo Anasi Nasimba

Tabla de contenidos

Índice de tablas

Índice de figuras

Resumen

El presente proyecto de titulación se desarrolló un sistema de seguridad y de gestionamiento de lugares de aparcamiento, mediante una secuencia de imágenes en movimiento (video), que identifica la detección de movimiento y cuantos lugares de estacionamiento se encuentran disponibles u ocupados en el estacionamiento. Para la recolección de la secuencia de imágenes del estacionamiento se utilizó una cámara IP conectada inalámbricamente, su tratamiento en tiempo real se lo realizó aplicando técnicas de visión artificial OPENCV, mediante una computadora de placa simple (Raspberry) para realizar la ejecución del procesamiento de imágenes mediante visión artificial la cual permite identificar si el lugar de estacionamiento se encuentra disponible u ocupado, además de la detección de movimientos dentro del estacionamiento en tiempo real, los datos e imágenes obtenidos se envían a la base de datos de acceso remoto, para la posterior visualización del estacionamiento completo o visualización de cada plaza de estacionamiento por individual, mediante una petición por el usuario desde la aplicación desarrollada mediante un software de realidad aumentada (UNITY) para monitorear, detectar moviento y establecer el sitio exacto de plaza de aparcamiento que se encuentra disponible mediante la presentación de un entorno 3D del estacionamiento que se encuentra ubicado frente al club de robótica de la Universidad de las Fuerzas Armadas ESPE-L.

Palabras clave:

- **GESTIONAMIENTO DE LUGARES DE APARCAMIENTO**
- **PROCESAMIENTO DE IMÁGENES**
- **VISIÓN ARTIFICIAL**
- **REALIDAD AUMENTADA**

Abstract

The present degree project presents a security system to manage parking places, using a sequence of moving images (video), that identifies many parking places that are available within the parking lot. For the collection of the sequence of the images in the parking lot it was used a camera IP, your treatment in real time it was done by applying artificial vision techniques OpenCV, using a single board computer (Raspberry) programmed that allows us to process images using artificial vision which allows to identify if the parking places are available or occupied, In addition to motion detection inside the parking lot in real time, the data and images obtained are sent to the remote access database, for later visualization of the entire parking lot or display of each parking space individually, through a request by the user from the application developed using augmented reality software (UNITY),to monitor, detect movement and set the exact parking space site that is available by presenting a 3D environment from the parking lot located in front of the robotics club of the University of the Armed Forces ESPE-L.

KEYWORDS

- **PARKING LOTS MANAGEMENT**
- **IMAGE PROCESSING**
- **ARTIFICIAL VISION**
- **AUGMENTED REALITY**

Capítulo I

1 Introducción

1.1 Planteamiento del problema

La congestión del tráfico ocasionada por los vehículos 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 serán peores a medida que incrementa la población. (Cristian Erazo, 2019)

La movilidad en ciudades que cuentan con gran cantidad de habitantes y no cuentan con una infraestructura vial para una gran afluencia de vehículos ha venido presentando problemas en los últimos años ya que la ciudad ha crecido significativamente y de la misma manera el porcentaje de vehículos en la ciudad, ocasionando problemas en las calles las cuales no presentan modificaciones o cambios que permitan facilitar la calidad de vida en conceptos de movilidad. El factor principal en la movilidad es el buscar un lugar de estacionamiento que brinde seguridad para guardar el vehículo. En zonas de alto congestionamiento, el estacionar un vehículo puede convertirse en un problema, llegando a provocar retrasos innecesarios por la búsqueda de un espaciamiento disponible.

Debido a estos factores descritos, se genera la necesidad de diseñar e implementar un sistema de seguridad y control de parqueadero a distintos usuarios la

posibilidad eliminar estos inconvenientes, ayudando a mejorar la movilidad, viabilidad y administración de tiempo.

1.2 Antecedentes

Hace algunos años el gestionamiento de parqueaderos de vehículos y las innovaciones tecnológicas han evolucionado de una manera significativa. Los sistemas de aparcamiento existen desde la creación de los primeros automóviles fabricados para brindar confort y movilidad en menor tiempo al ser humano, por esto en cualquier área donde hay una alta demanda de tráfico, hay sistemas de aparcamiento vehicular implementados y adecuados en diferentes partes de la cuidad. (Juan Chiza, 2016)

Los sistemas para el monitoreo y gestionamiento de aparcamientos son diseñados con el propósito de brindar seguridad, eficiencia y confort al usuario, aumentando la seguridad y disminuyendo de manera eficiente el tiempo de búsqueda de un espacio disponible de aparcamiento.

El principal problema para la administración de los estacionamientos es determinar los lugares de estacionamiento disponibles principalmente en las horas de alta demanda, en el centro de la ciudad o en centros de educación superior, por ejemplo.

Brindar un sistema para determinar las plazas de estacionamiento disponibles es necesario, por consiguiente, este proyecto pretende desarrollar un sistema de seguridad y gestionamiento de lugares de aparcamiento que puede informar a los administradores de la aplicación las plazas de estacionamientos que se encuentren disponibles u ocupadas.

1.3 Justificación e Importancia

El presente proyecto busca nuevas herramientas y tecnologías para implementar mejoras en los problemas relacionados con la seguridad y gestionamiento de parqueaderos, brindando una solución al problema de la localización de lugares disponibles en estacionamientos, debido a la gran afluencia de vehículos que transitan

por la ciudad, los cuales requieren una plaza de estacionamiento para estacionar el vehículo que se encuentre cerca de su lugar de destino.

El sistema a emplearse en el proyecto está orientado a brindar seguridad y gestionar los lugares de estacionamiento mediante en procesamiento de imágenes obtenidas de un parqueadero en tiempo real, brindando seguridad y optimizando al máximo el tiempo de búsqueda, brindando información óptima sobre la disponibilidad de lugares de aparcamiento dentro de un estacionamiento, determinando la ubicación correspondiente para cada vehículo mediante una aplicación basada en realidad aumentada para dispositivos móviles, que represente una ayuda a los usuarios menorando los retrasos al recorrer varios parqueaderos sin encontrar una plaza de estacionamiento disponible.

1.4 Objetivos

1.4.1 Objetivo General

Diseñar e implementar un sistema de seguridad, gestionamiento y detección de lugares de estacionamiento mediante secuencia de imágenes, dentro de la Universidad de las Fuerzas Armadas ESPE-L, utilizando procesamiento digital de imágenes, visión artificial y realidad aumentada.

1.4.2 Objetivos Específicos

- Obtener información necesaria sobre el procesamiento digital imágenes en tiempo real por medio del software de visión artificial OPENCV.
- Adquirir los conocimientos necesarios sobre las distintas librerías adecuadas para el procesamiento de imágenes requeridas para la aplicación.
- Realizar procesamiento de imágenes para la extracción de características esenciales de las imágenes del estacionamiento, mediante una computadora de placa simple.

- Determinar los dispositivos y elementos necesarios que formarán parte del sistema de seguridad y gestionamiento de plazas de estacionamiento.
- Posicionar correcta y adecuadamente la cámara Ip, para monitorear todos los segmentos del parqueadero establecidos.
- Relacionar la secuencia de imágenes adquiridas del estacionamiento en tiempo real, por medio de una cámara y OpenCV, para ser procesadas a través de la aplicación de realidad aumentada con el fin de proporcionar un sistema eficiente de seguridad y gestionamiento de parqueaderos.
- Entrenar el sistema de gestionamiento para determinar la disponibilidad de las plazas de aparcamiento disponibles.
- Realizar la verificación del funcionamiento correcto del sistema con el fin de corregir cualquier error que se presente en éste.

1.5 Variables de investigación

1.5.1 Variable independiente

Procesamiento digital de imágenes y realidad aumentada

1.5.2 Variable Dependiente

Seguridad y gestionamiento de lugares de estacionamiento

1.6 Hipótesis

Diseño e implementación de un sistema basado en procesamiento digital de imágenes y realidad aumentada de bajo costo, el cual permitirá el monitoreo y gestionamiento de los lugares de estacionamiento dentro de la Universidad de las fuerzas Armadas ESPE-L.

1.7 Trabajos relacionados

Javier Pacheco realizo un estudio en el año 2011 basado en el análisis automático de secuencias de video, implementada para ayudar al personal de vigilancia

en sus tareas en situaciones complejas en lugares de alta demanda y afluencia de personas y vehículos tales como aeropuertos, estacionamientos, etc.

Debido a este motivo los sistemas de monitoreo con cámaras de video son ampliamente usados, y de igual manera las tecnologías que permitan analizar el contenido del video de estas cámaras permitiendo identificar o diferenciar objetos, que está siendo monitoreados.

Para este trabajo de grado en concreto, se implementaron técnicas de visión artificial por computadora que dan facilidad para el reconocimiento de imágenes mediante algoritmos, los cuales permiten la identificación de los objetos que son removidos del área de visión establecido en la cámara. (SÁNCHEZ, 2011)

Para el reconocimiento de imágenes, mediante sistemas de visión artificial, extraer información para realizar el reconocimiento de objetos, ajustes de cámara entre otras cosas, Cristian Erazo y Sandra Narváez diseñaron un prototipo para la detección de aparcamientos libres mediante visión artificial en un parqueadero. A través de una cámara ubicada estratégicamente en un lugar de altura moderada, la cual transmite video hacia una computadora que permite el procesamiento digital de imágenes aplicando técnicas de visión artificial, almacenando los resultados en una base de datos para poder ser visualizados por los usuarios a través de una página web, al igual que en dispositivos móviles con sistemas operativos Android. (Cristian Erazo, 2019)

Kumar Amit implemento un sistema para el procesamiento de imágenes, utilizado para la clasificación de vehículos con el fin de mejorar la administración de estacionamientos en el año 2015. En concreto el trabajo está basado en el gestionamiento de lugares de aparcamiento, basándose en las dimensiones de los vehículos, para determinar el lugar correcto de parqueo bajo esas condiciones. Para determinar las dimensiones de los vehículos se realizó una comparación de la altura máxima para un grupo significativo de vehículos. Este sistema se lo implementó a

través del uso del software MATLAB y aplicando varios ensayos con vehículos de diferente tamaño para determinar con precisión las clases de vehículos. Con el fin de facilitar la implementación del sistema se utilizó una cámara montada sobre una superficie llana con configuraciones de píxeles predeterminado. (Kumar Amit, 2015)

1.8 Trabajos realizados en la Universidad de las Fuerzas Armadas

La Universidad de las Fuerzas Armadas ESPE cuenta con repositorio de trabajos de investigación relacionados al presente proyecto a desarrollarse, nombrándolos a continuación.

- Diseño e implementación de un sistema de gestionamiento de parqueaderos en la universidad de las fuerzas armadas ESPE-L utilizando algoritmo surf programado en software libre. (Juan Chiza, 2016)

- Diseño y construcción de un sistema de control para monitorear e incrementar la seguridad en el acceso vehicular al parqueadero de la ESPE-L, utilizando procesamiento digital de imágenes. (Mendoza Chipantasi, 2012)

- Desarrollo de software para el reconocimiento de texto manuscrito aplicando redes neuronales caso práctico formulario de inscripción de la Escuela Politécnica del Ejército Sede Latacunga. (Villacis, 2008)

Capítulo II

2 Marco Teórico

2.1 Procesamiento digital de imágenes

Las imágenes matriciales también llamadas imágenes de área se representan mediante una matriz y para cada uno los elementos se tiene su información de intensidad y de igual manera si la imagen está en escala de grises o el contenido de colores rojo R, verde, G y azul B la imagen es a color (RGB). El procesamiento digital de la imagen es aplicar a la matriz una técnica de procesamiento para realizar cambios en ella y que permita la segmentación de partes de la imagen que son del interes para el reconocimiento de características propias de la aplicación desarrollada. (Cristian Erazo, 2019)

Procesamiento digital de imágenes se define como el proceso para la adquisición de imágenes por diferentes medios como: fotografías, video, aparatos electrónicos, etc. Estas imágenes son procesadas con el fin de mejorarlas, extraer características definidas, reconocer un objetivo en una imagen o vídeo. (J, 2015)

Los métodos para el procesamiento digital de imágenes se dividen en dos áreas principales de aplicación: Para el mejoramiento de la información gráfica y mejor interpretación humana, al igual que para el tratamiento de datos de una imagen por medio de un sistema computacional o maquina autónoma incluyendo etapas para la transmisión o almacenamiento de datos en sitios específicos.

El procesamiento digital de las imágenes incluye las siguientes etapas:

- Adquisición o captura: Importar imágenes por diferentes herramientas.
- Realce y mejora: Es la etapa en donde mediante técnicas se resalta el aspecto visual de regiones importantes de las imágenes.
- Segmentación: Se encarga de delimitar las imágenes en regiones o áreas significativas.

- Extracción de características: Se encarga del reconocimiento y localización de objetos geométricos simples y complejos. Estas pueden ser líneas, puntos, curvas y cuádricas. (Gómez & Guerrero, 2016)

2.2 Etapas del procesamiento de imágenes

La función principal de la visión artificial es el tomar imágenes bidimensionales, para realizar un análisis sistemático mediante técnicas y algoritmos de procesamiento de imágenes para generar un modelo aproximado de lo que se está representando en una determinada imagen. Se debe tener en cuenta que la visión artificial no solo se limita a analizar imágenes estáticas sino también a tomas de diferentes ángulos de una misma escena e incluso videos de las cuales se pretende identificar o localizar objetos en movimiento. (Minardi, 2014)

Figura 1

Etapas del procesamiento de una imagen

2.2.1 Adquisición

La adquisición de imágenes se basa principalmente en la toma de fotografías o videos por medio de una cámara o un dispositivo electrónico para su posterior procesamiento digital.

- **Imagen en Escala de Grises.** Cada celda de la matriz de una imagen en escala de grises está representada por 1 píxel, cuyo valor dependerá del número de niveles de grises que posea la imagen. El número de niveles posibles al aumentar el número de bits será 2^n, siendo n el número de bits, es decir obtendremos 256 niveles de grises con ocho bits. Por lo general las imágenes en escalas de grises se crean con ocho bits, con los valores de (0) negro, (255) blanco, los 254 tonos grises restantes son suficientes para que el ojo humano no detecte transiciones agresivas, una buena imagen depende del número de píxeles y de niveles de grises que posea. (Costa Campos & Fernandez Bozal, 2019)

2.2.2 Pre-procesamiento

Está basado principalmente en la disminución del ruido de una imagen o secuencia de imágenes proporcionada por medio de la cámara u otro dispositivo electrónico, para el realce de detalles tales como brillo, color, etc.

Teniendo una imagen digital, se aplica esta técnica para mejorar el aspecto visual y la calidad de la imagen, y de esta manera facilitar su procesamiento para una aplicación específica. (Gómez & Guerrero, 2016)

2.2.2.1 Región de interes (Roi)

Al momento de obtener la imagen en escala de grises, se requiere establecer el are de interes (Roi) con el propósito de delimitar el área y reducir el ruido en la etapa procesamiento de la imagen reduciendo el área de pixeles en la imagen o secuencia de imágenes para interpretar los objetos y tratar de disminuir los errores.

La región de interes (Roi) será utilizada para identificación de vehículos en las , plazas de estacionamiento, por esta razón la mayor cantidad de información se encuentra en el centro de la imagen, discriminando objetos que no forman parte de nuestro primer análisis, que es la identificación de vehículos.

2.2.3 Segmentación

La segmentación es el proceso para dividir o separar los objetos que sean de nuestro interés de una determinada imagen, con el din de aislar la región de interes. (Gómez & Guerrero, 2016)

2.2.3.1 El algoritmo de Canny

Es usado para detectar todos los bordes existentes en una imagen determinada. Este método es considerado como uno de los más utilizados para el reconocimiento de contornos al emplear máscaras de convolución basándose en la primera derivada. Los contornos son zonas de píxels donde existe un cambio del nivel de gris. El procesamiento digital de imágenes está basado tratamiento de píxels en un ambiente reservado, por tal motivo el algoritmo de Canny utiliza máscaras, que muestran aproximaciones en diferencias finitas en los puntos de píxels. (Rebaza, 2007)

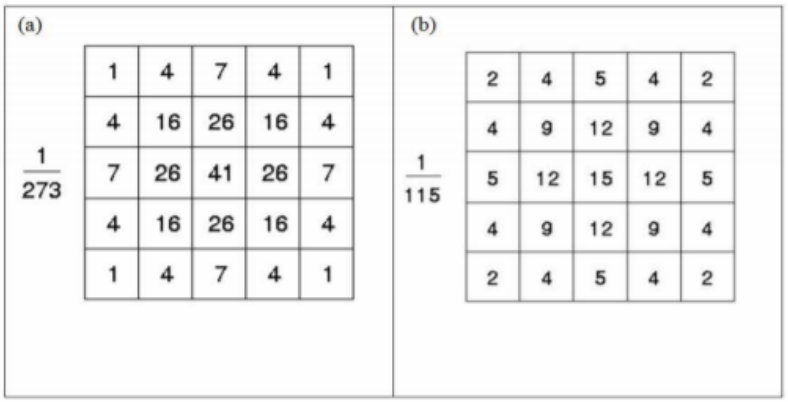
El algoritmo de Canny consiste en tres grandes pasos:

- **Obtención del gradiente.** El primer paso que se realiza es en la imagen original aplicando un filtro gaussiano con el propósito de suavizar la imagen y disminuir o eliminar el posible ruido existente en está.

  Una vez que se suaviza la imagen, para el ancho de bordes cada píxel se obtiene la magnitud y módulo (orientación) del gradiente, obteniendo así dos imágenes. El algoritmo para este primer paso se describe a continuación. (Sarcos, 2009)

Figura 2

Mascara de convolución recomendadas para el filtro gaussiano



Nota. Tomado de Desarrollo de un software que permita mostrar la presencia de características propias de un melanoma (p.23), por Burgos S. Gladys, 2009, Escuela Superior Politécnica del Litoral.

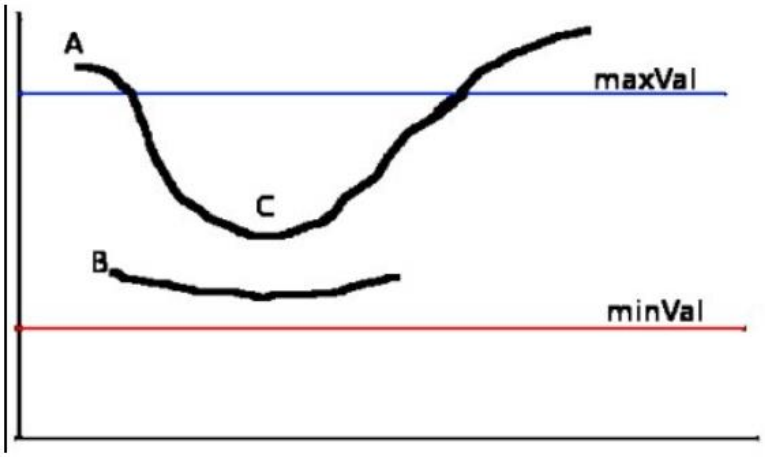
- **Supresión no máxima.** en este paso se lo realiza para el adelgazamiento de la anchura de los bordes de los objetos en la imagen, obtenidos mediante la aplicación de la gradiente en la imagen, hasta conseguir un píxel de ancho. Luego de aplicar la máscara de convolución recomendadas para el filtro gaussiano la imagen resultante nos sirve de entrada con los bordes adelgazados.

  El procedimiento se basa en la toma de cuatro direcciones las cuales son (0, 45, 90 y 135), respecto al eje horizontal. A cada pixel se le asigna una dirección más próxima al ángulo de la gradiente. Obteniendo una imagen con los bordes adelgazados de los objetos en la imagen. (Sarcos, 2009)
- **Histéresis de umbral.** Esta etapa decide cuáles son todos los bordes que son realmente bordes y cuáles no. Para esto, necesitamos dos valores de umbral, minVal y maxVal. Los bordes con un gradiente de intensidad superior a maxVal seguramente serán bordes y aquellos por debajo de minVal seguramente no

serán bordes, por lo que se descartan. Aquellos que se encuentran entre estos dos umbrales se clasifican como bordes o no bordes en función de su conectividad. Si están conectados a píxeles de "borde seguro", se consideran parte de los bordes.

Figura 3

Umbral estimado de histéresis para la detección de bordes de una imagen



Nota. Tomado de Detección de Bordes con Canny, por (OpenCV, s.f.)

El borde A está por encima de maxVal, por lo que se considera "borde seguro". Aunque el borde C está por debajo de maxVal, está conectado al borde A, por lo que también se considera un borde válido y obtenemos esa curva completa. Pero el borde B, aunque está por encima de minVal y está en la misma región que el borde C, no está conectado a ningún "borde seguro", por lo que se descarta. Por lo tanto, es muy importante que tengamos que seleccionar minVal y maxVal en consecuencia para obtener el resultado correcto. (OpenCV, s.f.)

2.2.4 Extracción de características

Esta técnica consiste en extraer las características de interés de la imagen tales como tamaño, colores, área, forma, perímetro; entre otras. También permite reconocer objetos geográficos y el tamaño de la superficie. (Gómez & Guerrero, 2016)

En proceso de reconocimiento de objetos, la aplicación de esta técnica basada en regiones de interes es ampliamente empleada.

Por ejemplo, las características de un automóvil consisten en formas y tamaños diferentes tomando estas consideraciones se procede a la utilización de clasificadores en cascada en funciones de Haar para la detección de objetos, basado en el entrenamiento a partir de varias imágenes de vehículos estacionados en los lugares de estacionamientos determinados para el proyecto tanto positivas y negativas.

2.2.5 Reconocimiento e Interpretación

Proceso realizado para la identificación de un objeto asignándole un determinado valor.

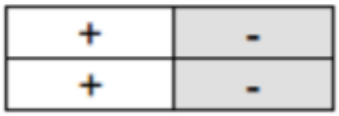
2.3 Clasificador Haar Cascade

Se trata de un clasificador en cascada para el reconocimiento o detección de objetos, mediante el entrenamiento con cientos de imágenes de los objetos, para ello es necesario tanto imágenes positivas (con el objeto a ser reconocido) y negativas (sin el objeto), para reconocer la forma u objeto que se desea por medio del clasificador. Es necesario que el clasificador en cascada pase por diferentes etapas las cuales que son aplicadas una a una mediante un proceso de optimización para la rápida configuración de este, el clasificador haar es una herramienta muy útil desarrollada para ayudar en el procesamiento de imágenes mediante visión artificial.

Haar Cascade se detalla mejor como una ventana de píxeles con tamaño y orientación variable que se encuentra dividida en diferentes regiones (rectangulares). Estas regiones son de dos tipos: positiva y negativa, mostrada a continuación. (Chicaiza, 2017)

Figura 4

Característica Haar



Nota. Tomado de (Chicaiza, 2017)

La ventana constituida por píxeles recorre la imagen completa, sumando los píxeles positivos y negativos respectivamente de cada imagen, como segundo paso se realiza la resta de estos pixeles para la obtención del denomino valor de la característica. Este valor se representa a través de la Ecuación 12:

$$H(x,y) = \sum_p I(x,y) - \sum_n I(x,y) \quad (12)$$

Donde:

$I(x,y) = Imagen\ a\ ser\ evaluada$

$H(x,y) = valor\ de\ la\ caracteristica\ haar\ cascade\ en\ el\ punto\ x\ y\ y$

$p\ y\ n = región\ positiva\ y\ negativa$

La característica principal del clasificador Haar está definida por los parámetros de tamaño, orientación y por la repartición de las regiones positivas y negativas, construyendo infinidad de tipos a la misma vez, estos parámetros dependen de los atributos del objeto que se desea detectar. (Guamán, 2015)

Al momento de construir una característica Haar se busca una estructura parecida a los atributos de una imagen como bordes, líneas y contornos similares al objeto dentro de la misma. La cascada haar está compuesta por un número de regiones determinado para la aplicación que se desee utilizar.

La cascada utilizada en el proyecto es la de Lienhart y Maydt la cual cuenta con Haar inclinadas para mejorar la detección de objetos en las imágenes detectando bordes y líneas de 45 grados. (Guamán, 2015)

Figura 5

Características para la clasificación de imágenes con Haar Cascade utilizadas por Lienhart y Maydt

Numero de regiones	Descripción	Imágenes.
Dos	Detectar bordes según la orientación: Borde diagonal: 45°	
Tres	Detectar lineas según su orientación: Linea diagonal: 45° y 135°	
Dos	Detectar centros-contornos	

Nota. Tomado de (Guamán, 2015)

Los atributos de una imagen son fáciles de calcular, pero el coste computacional es alto. Puesto que en una detección de un objeto se debe evaluar toda la imagen píxel por píxel y a escalas diferentes de una secuencia de imágenes en tiempo real. (Guamán, 2015)

2.4 Python

Python, es un lenguaje de programación orientado a objetos, interpretado e interactivo. Conocido por ser software libre y además open-source. (Holden, 2018)

Figura 6

Logotipo de Software Python

Nota. Tomado de Python, por (Holden, 2018)

Python tiene una base de librerías y funciones incorporadas en el propio lenguaje, facilitando la realización de programas sin necesidad de tener que empezar desde cero. Lo cual ha llevado a este lenguaje a ampliar su entorno, dándole robustez a sus funciones y lo convierte en un lenguaje de más fácil de aprendizaje. (Berrones Reyes, 2019)

2.4.1 Anaconda navigator

Para gestionar y compilar los paquetes de Python se utiliza Anaconda Navigator la cual es una interfaz gráfica de usuario de escritorio incluida en la distribución de Anaconda® que le permite:

- Gestionar
- Compilar
- Iniciar aplicaciones
- Administrar paquetes (Anaconda Documentation, 2020)

Para la edición y ejecución de código se realiza a través de la siguiente aplicación.

2.4.2 Spyder

Spyder es un entorno científico gratuito y de código abierto escrito en Python, para Python, y diseñado por y para científicos, ingenieros y analistas de datos. Cuenta con una combinación única de la funcionalidad avanzada de edición, análisis, depuración y creación de perfiles de una herramienta de desarrollo integral con la exploración de datos, ejecución interactiva, inspección profunda y hermosas capacidades de visualización de un paquete científico. (SPYDER, 2020)

Características de spyder

Algunas de las características notables de Spyder son:

- Resaltado de sintaxis personalizable

- Disponibilidad de puntos de interrupción (depuración y puntos de interrupción condicionales)
- Ejecución interactiva que le permite ejecutar línea, archivo, celda, etc.
- Ejecuta configuraciones para selecciones de directorios de trabajo, opciones de línea de comandos, consola actual / dedicada / externa, etc.
- Puede borrar variables automáticamente (o ingresar a la depuración)
- La navegación a través de celdas, funciones, bloques, etc. se puede lograr a través del Explorador de Esquemas
- Proporciona introspección de código en tiempo real (la capacidad de examinar qué funciones, palabras clave y clases son, qué están haciendo y qué información contienen)
- También proporciona funciones como ayuda, explorador de archivos, búsqueda de archivos, etc.

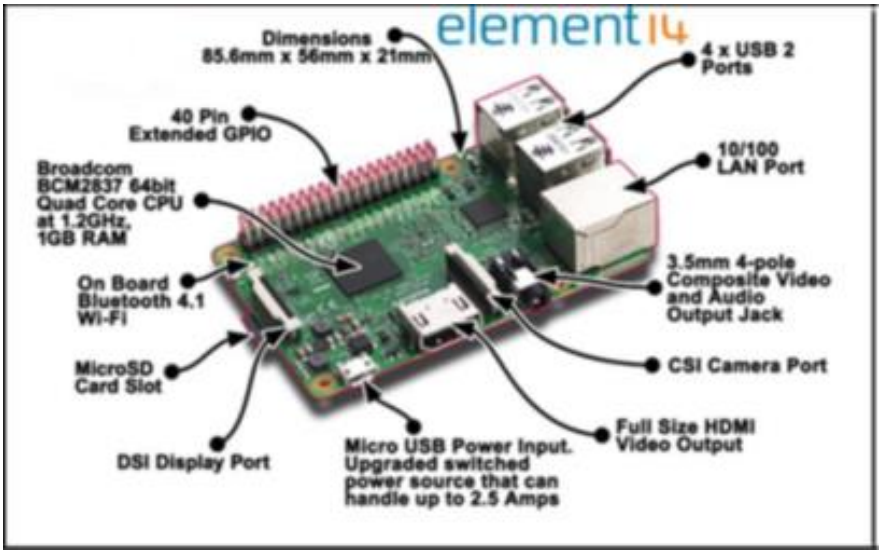
Siempre es necesario contar con entornos interactivos para crear aplicaciones de software y este hecho se vuelve muy importante cuando se trabaja en los campos de la ciencia de datos, la ingeniería y la investigación científica. (Urooj, 2019)

2.5 Raspberry Pi

Se trata de una tarjeta electrónica para el procesamiento computacional de tamaño pequeño la cual puede ser conectada a un monitor para su configuración. Cuenta con un procesamiento con la capacidad de utilizar software que normalmente son utilizados en computadoras de escritorio o computadoras personales, por ejemplo: Excel, Word, notas de texto y juegos de bajo consumo, al igual que reproductores de música y video de alta definición. Sus versiones desde el 2012 sufren mejoras significativas, aumentado su capacidad de procesamiento como de almacenamiento, lo cual representa un costo más elevado de las misma. (Cristian Erazo, 2019)

Figura 7

Raspberry Pi 4 Model B



Nota. Tomado de (Raspberry Pi Foundation, 2016)

2.6 Cámara Ip

Una cámara de red es descrita como la integración de una computadora en una cámara, la cual realiza la adquisición de imágenes de video a través de un cable de red o por vía inalámbrica, permitiendo a los usuarios administrar y almacenar video en la nube o dispositivos de almacenamiento compatibles con el protocolo IP. Debido a su fácil instalación y atributos que brindan este tipo de cámaras, se utilizan en mayor parte para sistemas cerrados de vigilancia o CCTV. (Ycezalaya, s.f.)

Una cámara IP costa de un lente, un sensor, un procesador, un chip de compresión de video y un chip de Ethernet para la conexión de red y transmisión de datos en tiempo real. (REDATEL, 2016)

Para el proyecto a desarrollarse se utiliza una cámara IP A12 Wifi para la adquisición de la secuencia de imágenes (video) en tiempo real para su posterior procesamiento e identificación de los lugares de estacionamiento disponibles.

2.6.1 Características principales

Características principales de la cámara IP A12 Wifi R80X20-PQ:

- Tiene una resolución aproximadamente HD de 1080 píxeles
- Robotizada (giro 90º verticales, 355º horizontales)
- Compatible con Sistema P2P, no necesita IP fija ni sistema ddns.
- Toma fotos y transite video en tiempo real.
- 4 led infrarrojos array led para visión nocturna de 30mts

2.7 Base de datos

Una base de datos es un sistema de almacenamiento de datos con el propósito de conservar la información y mantenerla disponible para los usuarios. Estos datos almacenados en la base son normalmente de procesos automáticos y de toma de decisiones.

Las principales componentes de los que se encuentra estructurado una base de datos son los siguientes: datos, programas, dispositivos de almacenamiento de información (texto, documentos, etc) y listas de usuarios presentando diversas ventajas en la integración de datos y la compartición de los mismos. (Martinez, 2015)

En caso de este proyecto, servira para que la información almacenada en la Base de Datos actualice el estado en tiempo real del los lugares de estacionamiento en la intefaz, mostrando la disponiblidad de los mismos asi como tambien los peatones presentes dentro del todo el estacionamiento.

2.7.1 Caracteristicas de la base de datos

Entre las principales objetivos de los sistemas de base de datos se puede mencionar los siguientes: (Ortiz, 2000)

- Independecia lógica y física de los datos.
- Redundancia mínima

- Acceso concurente por parte de múliples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditorìa.
- Respaldo de recuperación.
- Acceso a través de lenguajes de programación estándar.

2.7.2 Firebase

Firebase es una plataforma BaaS (Backend as a Service), desarrollada por Google que facilita el desarrollo de apps, proporcionando un servidor Backend para las aplicaciones. Además, el mismo Backend puede ser utilizado de forma común en diversas plataformas: Android, IOS y web.

Firebase proporciona una solución eficaz frente no solo a problemas de desarrollo, sino también de escalabilidad a medida que la base de usuarios de la aplicación crece, ya que los servidores son proporcionados por Google. Entre sus funcionalidades se encuentra un servicio de autenticación, base de datos en tiempo real, almacenamiento de archivos, solución de errores, funciones Backend, testeo, y medida de estadísticas recogidas de los usuarios. (Baltazar, 2020)

Para el desarrollo de este proyecto una creado el entorno de Spyder para el desarrollo de la programación en Python, para la base de datos se lo realiza en Firebase. (Baltazar, 2020)

2.8 OpenCV

2.8.1 Historia del OpenCV

OpenCV o biblioteca libre de visión artificial desarrollada por la empresa Intel. En enero de 1999 fue presentada por primera vez su versión alfa, este software proporciona cientos de funciones para la captura, análisis y manipulación de datos

visuales y puede eliminar algunos de los problemas que enfrentan los programadores al desarrollar aplicaciones que dependen de la visión por computadora. Algunas partes de la biblioteca también proporcionan funciones de interfaz de usuario y reconocimiento de patrones. OpenCV se ha empleado en aplicaciones prácticas y creativas, incluidos vehículos autónomos y nuevas formas de arte digital. (OpenCV, s.f.)

Figura 8

Logotipo de OpenCV

Nota. Tomado de (OpenCV, s.f.)

OpenCV proporciona funciones comunes o capacidades complejas que los desarrolladores pueden utilizar en sus programas. La biblioteca OpenCV contiene cientos de funciones que apoyan la captura, análisis y manipulación de información visual alimentada a una computadora por cámaras web, archivos de video u otros tipos de dispositivos. Se pueden usar funciones que contienen algoritmos para detectar rostros, rastrear movimiento y analizar formas. (OpenCV, s.f.)

El siguiente proyecto proporciona un entorno amigable con el usuario, además de una gran variedad de funciones que se desempeñan con visión artificial, lo cual se lo implementó con programación en código Python en conjunto con una aplicación desarrollada en un software de realidad aumentada para dispositivos móviles.

2.8.2 Características principales

A continuación, se muestran las principales características que ofrece OpenCV:

- Interfaces compatibles con C++, Python y Java
- Software para el procesamiento y tratamiento en tiempo real de imágenes aplicando técnicas de visión artificial.
- Desarrollado principalmente para procesadores de Intel y Linux.
- Es compatible con sistemas: Windows, Linux y Mac.

2.8.3 Módulos de OpenCV

OpenCV posee una estructura modular descritos a continuación en la Tabla 1.

Tabla 1

Características de los módulos de OpenCV

MÓDULO	FUNCIÓN
Núcleo	Módulo con estructura básica de los datos y las funciones básicas para el procesamiento de imágenes.
Highgui	Presenta una interface de usuario sencilla, para el tratamiento de imágenes y captura de video en tiempo real. Además, permite manipular las ventanas de imagen, manejo de barras etc.
Imgproc	Provee de una interface de usuario para el tratamiento de imágenes y video.
Video	Sirve para el análisis de video incluyendo métodos para el seguimiento y reconocimiento de objetos, mediante sustracción de fondo entre otras técnicas.
Objdetect	Contiene métodos y algoritmos para la detección y reconocimientos de objetos en una secuencia de imágenes.

Nota. Tomada de Módulos de la librería OpenCV, (OpenCV, 2012)

2.9 Realidad Aumentada

Se puede evidenciar que en última década han aparecido una serie de entornos tecnológicos avanzados especialmente idóneos para el desarrollo y la evaluación de las

competencia (Redecker, 2013). La realidad aumentada es conocido como un conjunto de tecnologías que combinan imágenes reales y virtuales, de forma interactiva y en tiempo real, de manera que permite añadir la información virtual a los elementos que el usuario dispone dentro del mundo real.

Este tipo de tecnología permite la interacción con otros seres, así como la manipulación de objetos que existan en un entorno específico (Beck, 2009). Teniendo en cuenta que la realidad aumentada (AR) no sustituye la realidad física, sino que sobrepone información virtual añadiendo esa información en el contexto de la realidad existente.

2.9.1 Usos de la realidad aumentada en la ingeniería

Para los proyectos de ingeniería, la realidad aumentada es un recurso excelente para los ingenieros, ya que permite al usuario seleccionar el modo de visualización con el que desea trabajar y prever cómo se desarrollarán las diferentes fases de un proyecto de ingeniería o construcción. La realidad aumentada es utilizada sobre todo por los ingenieros, para definir una visión a través de un dispositivo tecnológico, de un entorno físico del mundo real, cuyos elementos se combinan con elementos virtuales, creando una realidad mixta en tiempo real. Con la ayuda de la tecnología, la información sobre el mundo real se convierte en interactiva y digital como una forma de ampliar el mundo real.

La edición de imágenes con software informático, ha sido una herramienta que los ingenieros han usado para representar sus datos y proyectos de una forma más

visual, consiguiendo mayores aplicaciones interactivas y la combinación de objetos reales con otros virtuales. (DYNATEC, 2015)

2.9.2 ¿Qué nos ofrece la realidad aumentada?

Nos permite la superposición de datos en tiempo real, creados virtualmente con el fin de recrear elementos ficticios en un entorno real con los que podemos interactuar por medio del reconocimiento de patrones que realiza un software.

Son muchas las áreas de ingeniería en las que la realidad aumentada son de gran ayuda para trabajar con dispositivos eléctricos o mecánicos, circuitos electrónicos, modelos a escala, etc. estableciendo de esta manera el vínculo entre los conceptos teóricos y la práctica, sobre todo en las fases en las que se realizan las pruebas finales con dispositivos reales. (DYNATEC, 2015)

2.9.3 Unity

Considerado como un motor para videojuegos desarrollado por Unity Technologies, principalmente usado para la creación de videojuegos 2D, 3D y simulaciones para computadoras, consolas y dispositivos móviles, utilizando lenguajes de programación C y JavaScript.

Unity puede usarse junto con Blender, 3ds Max, Maya, Softimage, Modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks y Allegorithmic Substance. Los cambios realizados a los objetos creados con estos productos se actualizan automáticamente en todas las instancias de ese objeto durante todo el proyecto sin necesidad de volver a importar manualmente. (Unity, 2020)

Figura 9

Logo oficial de Unity

Nota. Tomado de (Unity, 2020)

2.9.4 Vuforia

Es una plataforma que cuenta con el respaldo de Qualcomm, por lo que es una herramienta muy buena para desarrollar aplicaciones con cualquier tipo de experiencia de Realidad Aumentada para iOS y para Android sin importar el dispositivo o soporte en el que será ejecutado que contiene objetos prefabricados y scripts para dar funcionalidad de realidad aumentada al proyecto de Unity. (Developer, 2018)

El SDK Vuforia, es un kit de desarrollo que sirve para la creación de realidad aumentada y está disponible para Android e iOs. Para instalarlo es necesario registrarse en la página web de Vuforia y descargar el SDK, luego en Unity se debe importar el paquete del proyecto, en una cuenta web, se tiene una base de datos que contiene información sobre los marcadores o targets, que reconocerá el dispositivo en los proyectos de realidad aumentada.

Al importar el paquete aparecerá el siguiente contenido:

Editor: Contiene los scripts que permiten interactuar con los datos de los targets.

Plugins: Contiene archivos binarios para integrar el SDK con la aplicación de Unity para Android y iOs. (Tamayo, 2016)

2.9.5 Image target

Son imágenes que Vuforia SDK se puede detectar y rastrear. A diferencia de los marcadores de referencia tradicionales, códigos de matriz de datos y códigos QR, el image target no necesita regiones blancas o códigos negro especial para ser reconocido. El SDK detecta y rastrea las características que se encuentran en la imagen comparándolas contra una base de datos, una vez que se las reconoce realiza las tareas que se hayan programado. (Vuforia, 2019)

Image target

Figura 10

Image target

Nota. Tomado de (Vuforia, 2019)

Capítulo III

3 Diseño e Implementación

En el presente capítulo, se describe de manera detallada el diseño e implementación del sistema de seguridad y gestionamiento de lugares de aparcamiento, la base de datos utilizada y la forma de empleo del software Anaconda Navigator y como editor de texto de Python para la determinación de las plazas de estacionamiento disponibles, al igual que la detección de movimiento en el estacionamiento usando Spyder y realidad aumentada (Unity) para la administración y monitoreo de las plazas de estacionamiento mediante la creación de una aplicación para dispositivos móviles.

3.1 Especificaciones del software y equipos

Para el desarrollo del proyecto se tienen las siguientes especificaciones de software y equipos requeridos para el mismo.

- Python versión 3.8.9
- Aplicación Spyder 5
- Cámara IP A12 Wifi
- Router Tp-Link
- Raspberry PI 4 model B
- OpenCV 4.5.1
- Software de Realidad Aumentada (Unity)

Al realizar el análisis de una imagen primeramente se debe realizar un proceso de identificación de características relevantes al momento del procesamiento.

En la siguiente, sección se detallarán los pasos para la instalación de softwares utilizados para el desarrollo de este proyecto.

3.2 Anaconda Navigator

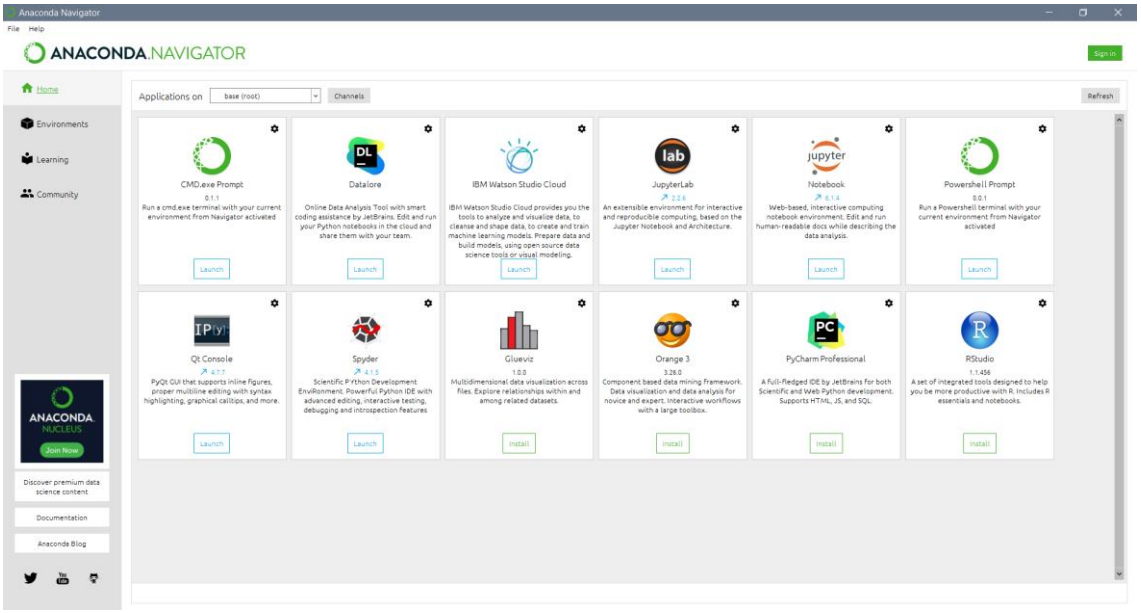
Anaconda Navigator es una interfaz la cual contiene aplicaciones para desarrollar código R y Python, por medio de esta aplicación se puede gestionar entornos, paquetes y librerías.

A continuación, se describe el proceso para crear un entorno en el cual se instalarán los paquetes que necesite el editor de lenguaje Python (Spyder), para el uso de visión artificial (OpenCV).

En la aplicación Anaconda Navigator, Figura 11, se selecciona Environments.

Figura 11

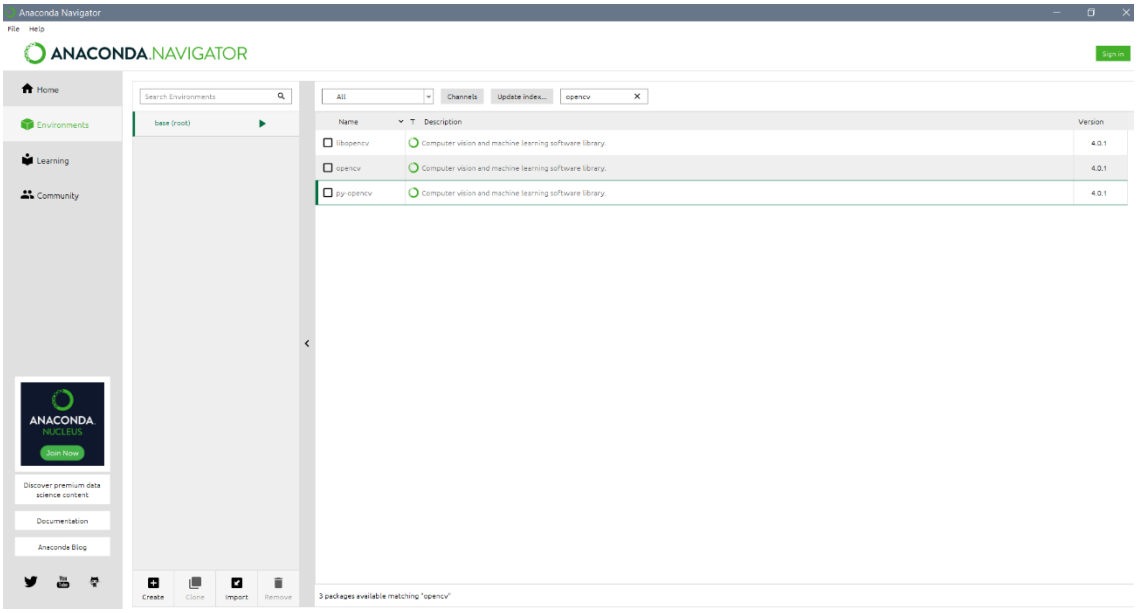
Aplicación Anaconda Navigator



En Environments, Figura 12, se selecciona el ícono Create el cual nos permitirá crear un entorno de trabajo con las aplicaciones necesarias para el desarrollo del proyecto.

Figura 12

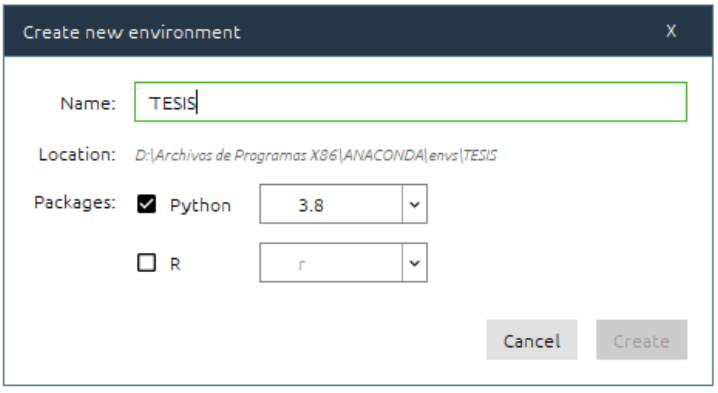
Aplicación Anaconda – Environments



Ingresamos el nombre que deseamos dar al entorno de trabajo, Figura 13, en este caso se llamará TESIS, luego elegimos la versión de Python, que será 3.8.9 y se selecciona Create.

Figura 13

Ventana para Crear un Nuevo Entorno



En el nuevo entorno creado en Anaconda Navigator llamado TESIS se realizará la instalación de los paquetes y librerías requeridos para el desarrollo del proyecto.

3.3 Instalación de las librerías de Python

Ventana Anaconda Prompt.

Para la activación de nuestro entorno para el desarrollo del proyecto se digita conda actívate TESIS (nombre del entorno), como se muestra en la Figura 14, asegurando que las librerías se instalaran en el entorno seleccionado.

Figura 14

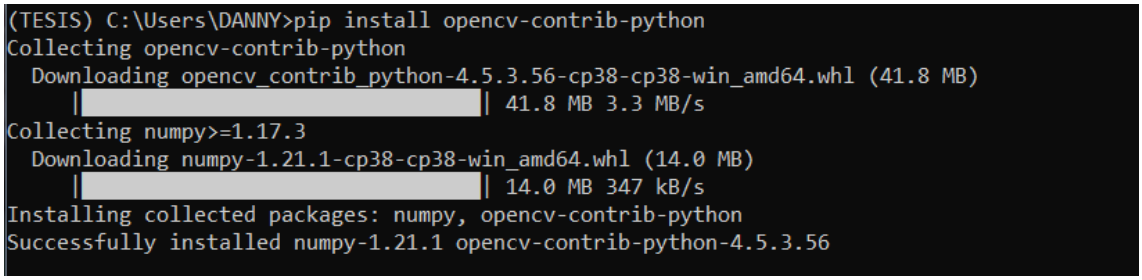
Activación del Entorno para Instalación de Librerías

Para la instalación de librerías se puede usar el comando conda install (nombre de librería) o pip install (caracteres + nombre de librería), para conocer la forma correcta de instalar la librería requerida, es necesario visitar la página oficial de la librería y verificar sus requisitos de la librería seleccionada dado que algunas pueden depender de otras.

En la Figura 15, se realiza la instalación de los paquetes de visión artificial de OpenCV en el entorno TESIS con el comando *pip install opencv-contrib-python*, la cual nos sirve para el procesamiento de imágenes aplicando visión artificial con el editor de código de Python, Spyder versión 5.

Figura 15

Instalación de librerías de OpenCV en el entorno TESIS creado en Anaconda

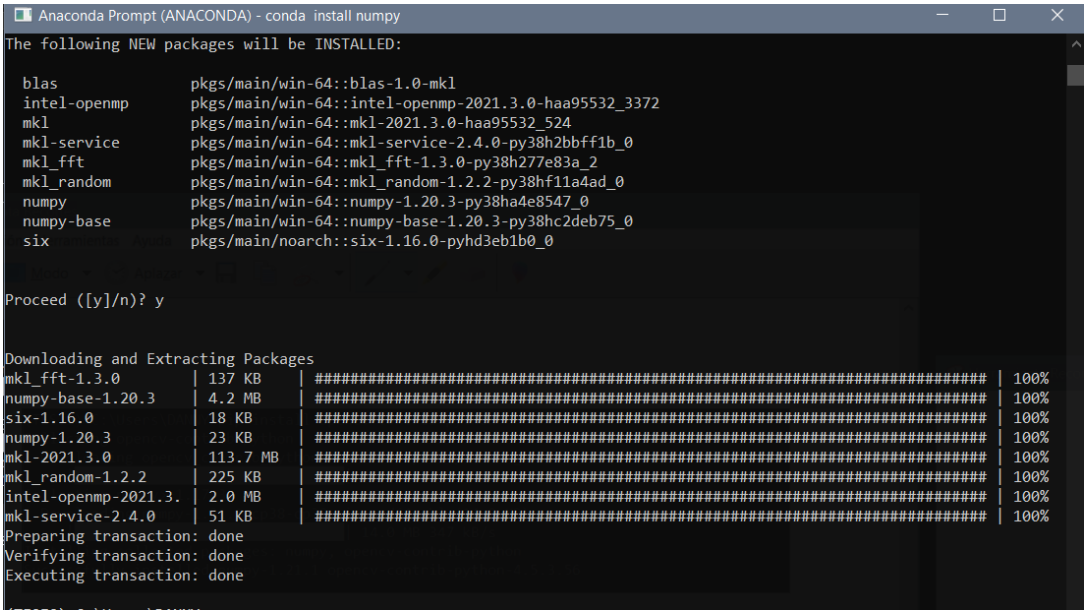


A continuación, en la Figura 16 se instala la librería de numpy con el comando.

- *conda install numpy*

Figura 16

Instalación de librerías de OpenCV en el entorno TESIS creado en Anaconda

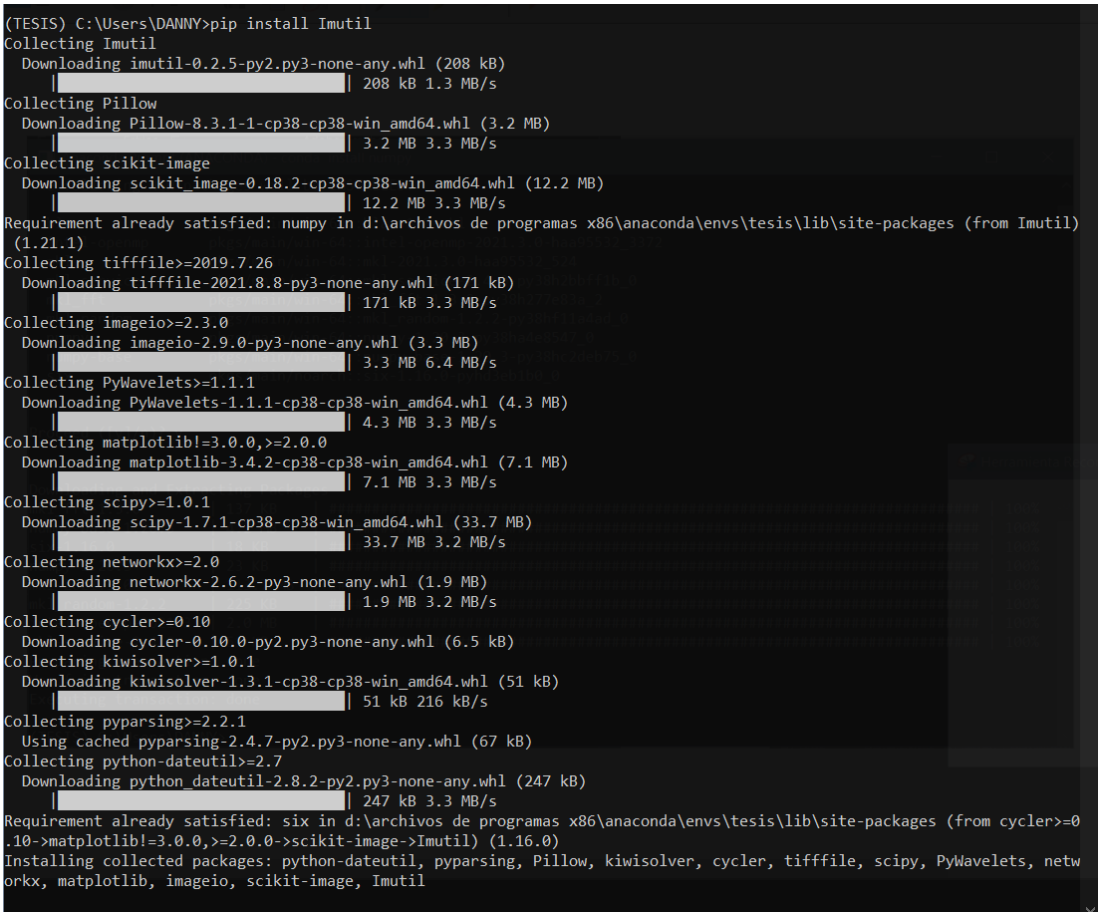


De la misma manera en la Figura 17, se instala la librería Imutils mediante comandos ejecutados por el símbolo de sistema de Anaconda Prompt.

- pip install Imutils

Figura 17

Instalación de librerías de Imutils en el entorno TESIS desde en el editor de líneas de código de Anaconda Prompt



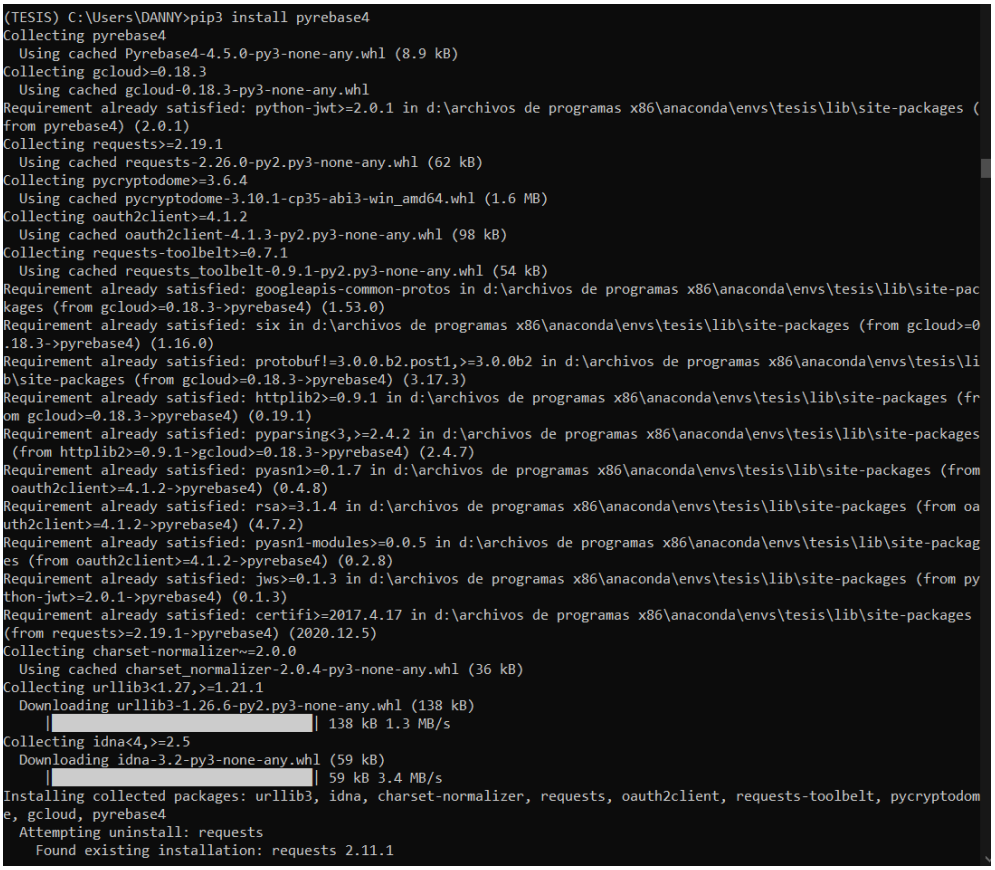
De la misma forma se instalarán las librerías de matplotlib, pyrebase, picamera.

- conda install matplotlib
- pip3 install pyrebase4
- conda install picamera

En la Figura 18, utilizamos como ejemplo la librería de pyrebase y se ingresa el código *pip3 install pyrebase4* en el símbolo del sistema de anaconda Prompt.

Figura 18

Instalación de librerías de Pyrebase4 en el entorno TESIS desde en el editor de líneas de código de Anaconda Prompt



3.4 Raspberry Pi

Raspberry Pi, es una tarjeta electrónica de computador en la cual se ejecutará la programación desarrollada, que permite establecer la comunicación entre la tarjeta y la aplicación implementada en unity mediante el acceso a una base de datos remota (Firebase) dedicada específicamente para el proyecto y con la cual se determina las plazas de aparcamiento disponibles al igual que la detección de movimiento existente en todo el estacionamiento.

Para comenzar con la implementación es necesario la instalación de la base de datos remota en la tarjeta raspberry Pi, para lo cual se necesita descargar las librerías y

complementos, que permiten la comunicación y acceso a la base de datos desde diferentes dispositivos conectados a una red de internet.

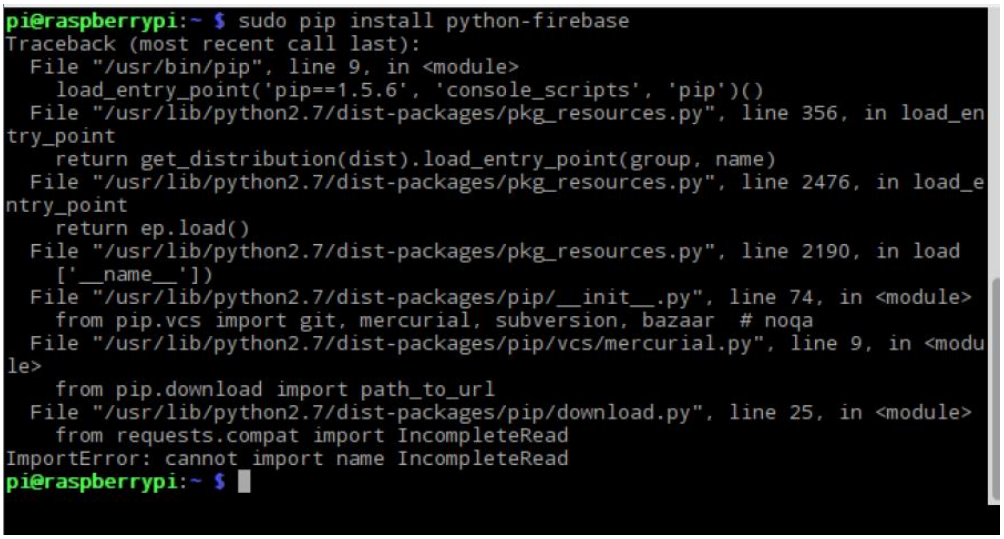
A continuación, se muestran los comandos requeridos para la instalación de la base de datos Firebase en la raspberry Pi.

- sudo pip install python-firebase
- sudo pip install pyasn
- sudo apt-get install python.rpi. gpio

Las cuales servirán para la comunicación con la base de datos, y la conexión física de Python con las raspberry Pi como se muestra en la Figura 19, respectivamente para cada comando.

Figura 19

Instalación de librerías de pyrebase raspberry Pi



3.5 Spyder

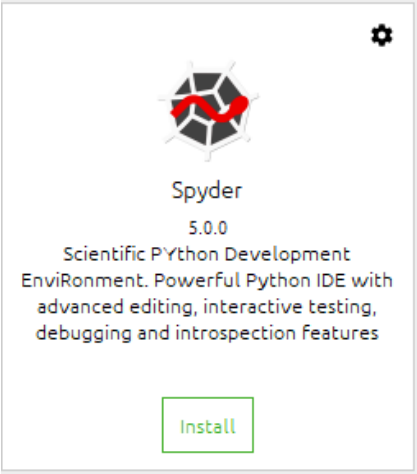
Es un entorno de desarrollo integrado multi-idioma para desarrollar código Python, mediante esta aplicación se pueden gestionar entornos y paquetes. Proporciona análisis de código en tiempo real, sugerencias de llamadas y funciones de

acceso. Es un navegador de funciones y clases para una experiencia de edición más eficiente.

Para la utilización de la aplicación Spyder versión 5, seleccionamos Install para su instalación, como se observa en la Figura 20.

Figura 20

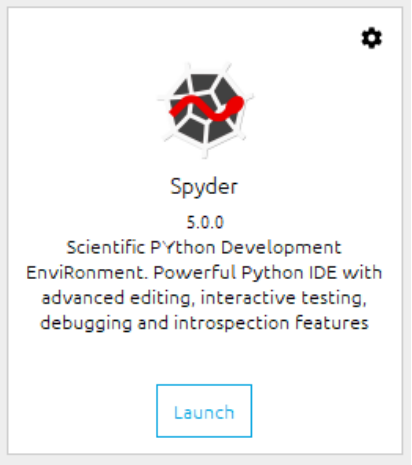
Instalación de Spyder versión 5.0.0



Después ejecutamos la aplicación, seleccionando Launch, Figura 21.

Figura 21

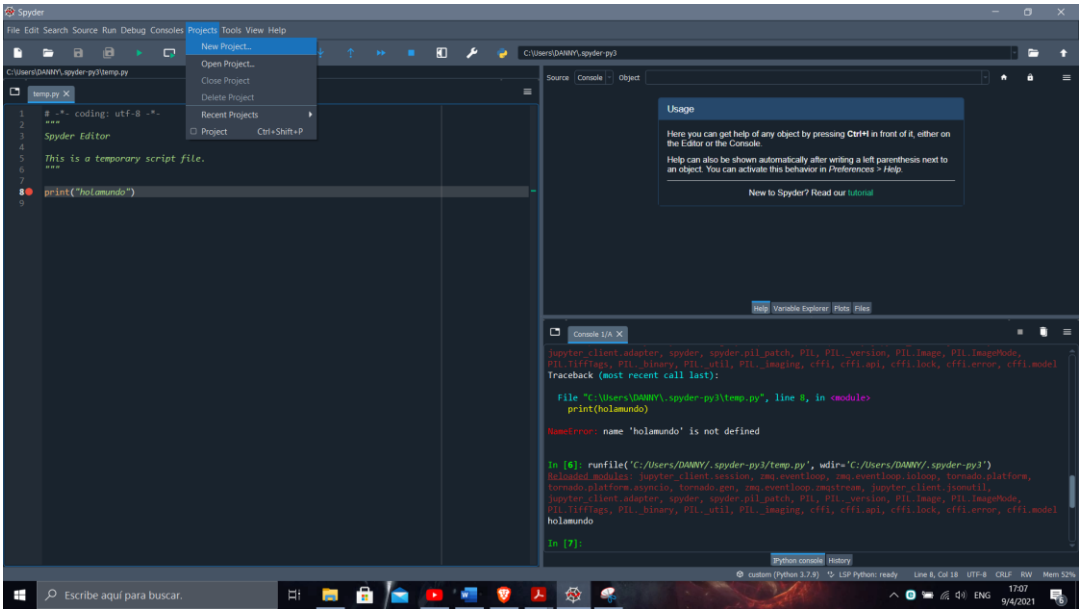
Ejecución de Spyder versión 5.0.0



Se observa, en la Figura 22 el área en la cual se editará y ejecutará el código del proyecto.

Figura 22

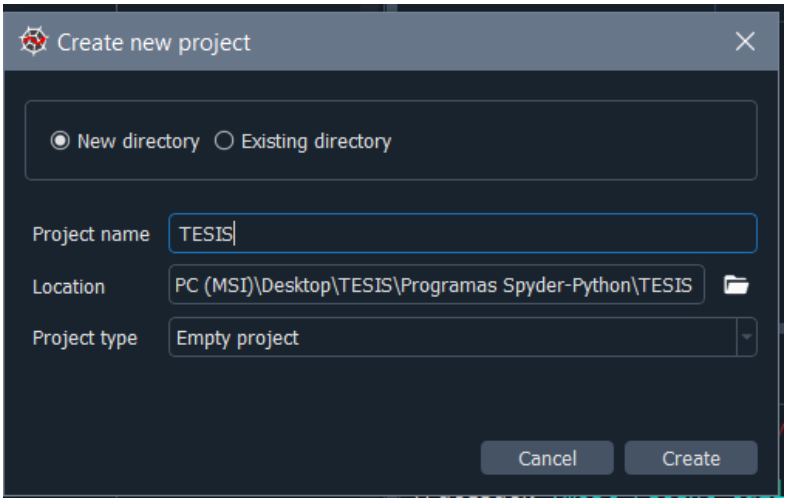
Ventana Spyder para edición de código Python



Finalmente, en la barra de herramientas se da click en Projects, se selecciona New Project lo que nos despliega la siguiente ventana como se muestra en la Figura 23.

Figura 23

Ventana para crear un nuevo Proyecto en entorno Python



En New Project seleccionamos el nombre de nuestro nuevo archivo y la ubicación donde va a guardar sus avances, por defecto el programa selecciona la

versión de Python más actual instalada, y para finalizar se selecciona el icono Create que se encuentra en el inferior de la ventana.

3.6 Estructura del sistema

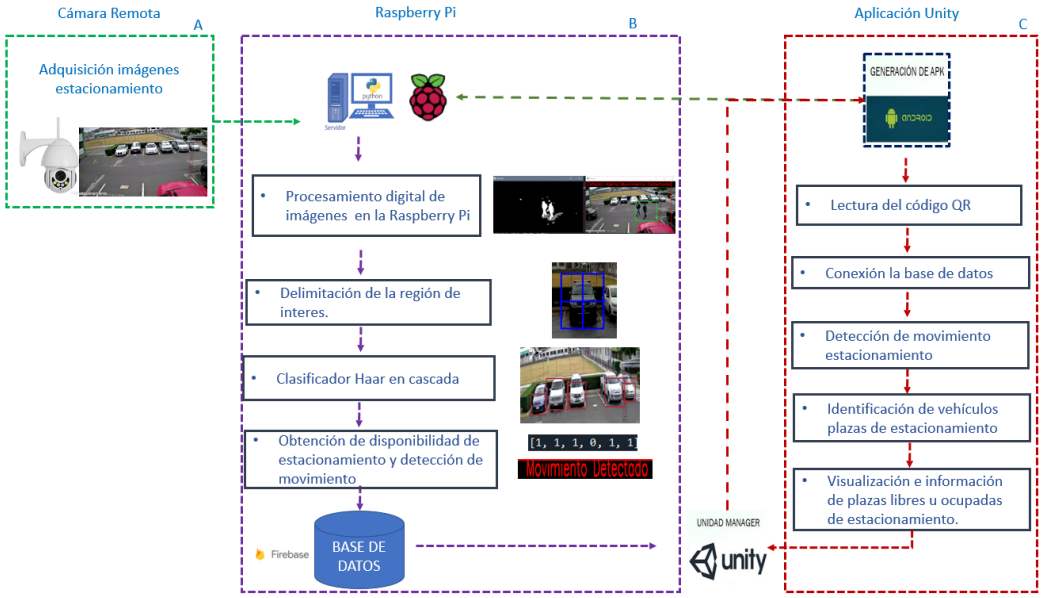
En la Figura 24, en el boque A, se representa la adquisición de las secuencias de imágenes de las plazas de estacionamiento, para su posterior tratamiento.

Para el bloque B, se utiliza el editor de código Spyder, en el cual se realiza el procesamiento digital de imágenes obtenidas del estacionamiento por medio de una computadora de placa Raspberry Pi, con el fin de realizar la detección de moviento e identificación de vehículos en cada plaza de estacionamiento.

En el bloque C, se desarrolla la aplicación para dispositivos móviles de realidad aumentada, mediante la adquisición de imágenes almacenadas en la base de datos Firebase, por medio del escaneo por código QR, detectando movimiento e identificación de vehículos en las plazas de estacionamiento y representado mediante un entorno 3D las plazas libres u ocupadas del estacionamiento.

Figura 24

Estructura del sistema de estacionamientos

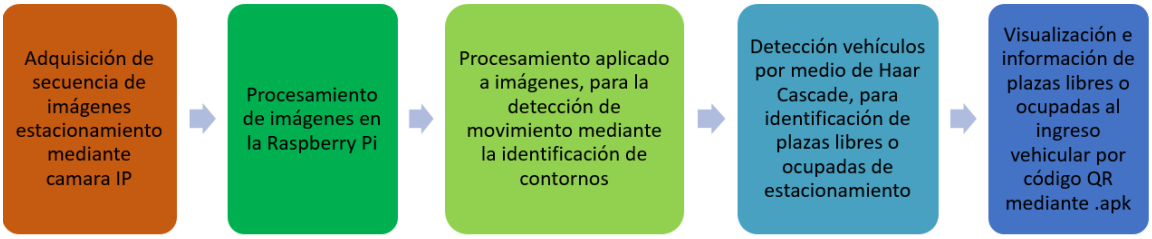


3.7 Diagramas de bloques

En la Figura 25, se describe las etapas que se cumple en cada bloque del sistema desarrollado.

Figura 25

Diagrama de bloques del sistema de estacionamientos

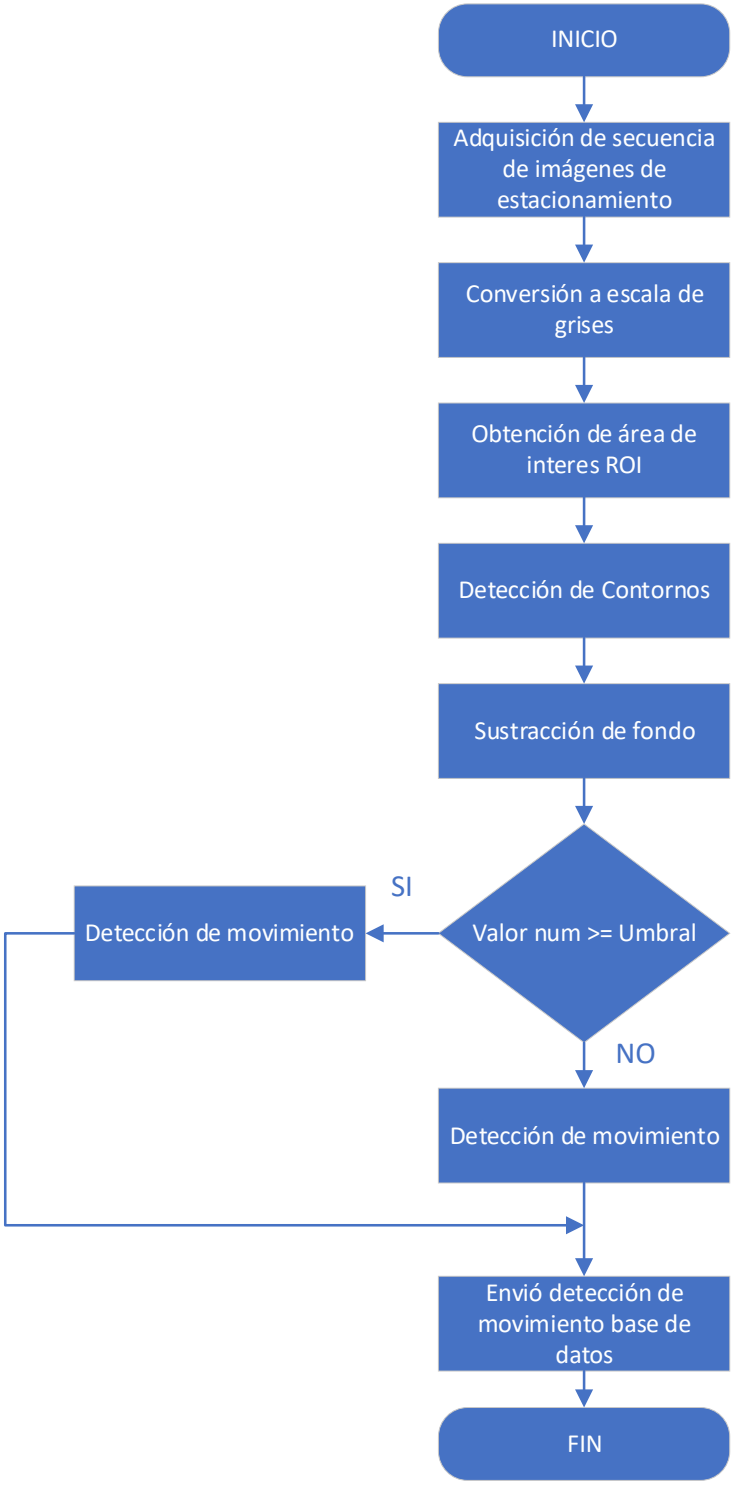


3.8 Procesamiento digital secuencia de imágenes estacionamiento

Para realizar la detección de moviento de las secuencias de imágenes adquiridas del estacionamiento, se utilizó el procedimiento que se representa en el diagrama de flujo de la Figura 26.

Figura 26

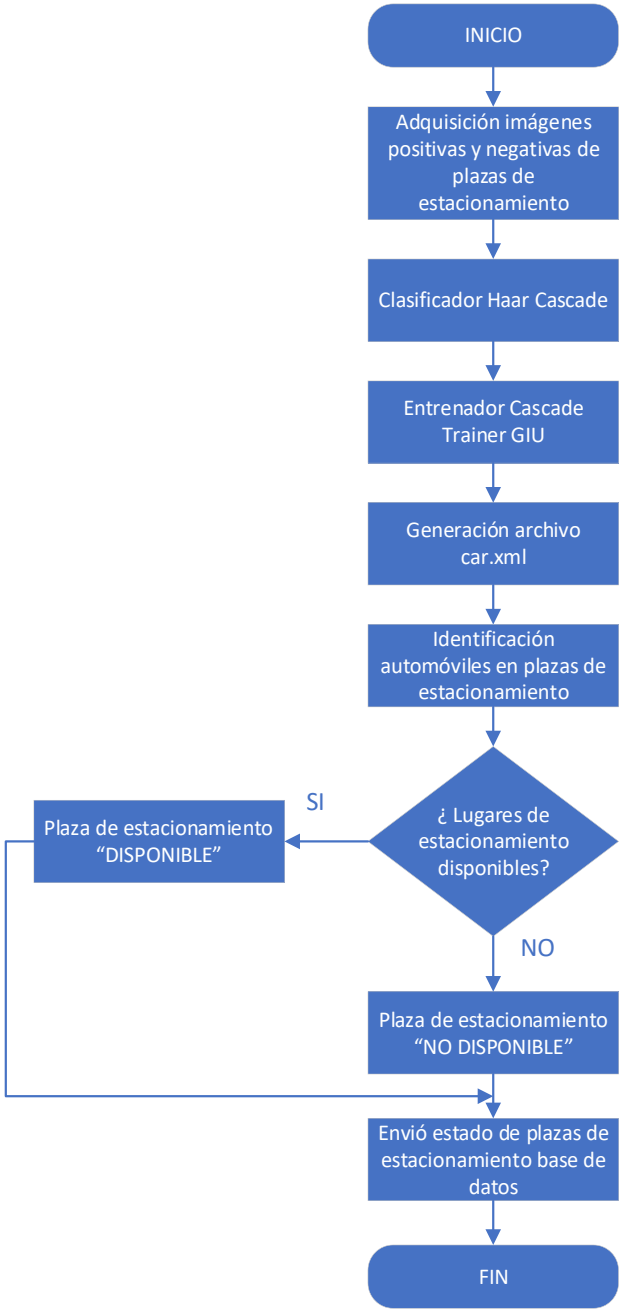
Diagrama de flujo del procesamiento digital para identificación de movimiento dentro del estacionamiento



A continuación, en la Figura 27 se presenta el diagrama de flujo para la detección de vehículos de cada plaza de estacionamiento y determinar su disponibilidad.

Figura 27

Diagrama de flujo del procesamiento digital para identificación de vehículos de las plazas de estacionamiento disponibles u ocupados



A continuación, se explica el código desarrollado en Spyder para la implementación de esta etapa del sistema diseñado.

Una vez creado el entorno en Anaconda e instalar las librerías para el procesamiento de imágenes digitales, el primer paso es importar estas librerías de OpenCV para el tratamiento de la secuencia de imágenes, numpy para el cálculo y análisis de datos, pyrebase para la comunicación con la base de datos para el almacenamiento de datos y threading para la ejecución de hilos o subprocesos simultáneamente en el software Spyder (TESIS).

```
import cv2
import time
import numpy as np
from datetime import datetime
import pyrebase
from threading import Thread
```

Para la adquisición de imagen del estacionamiento se utiliza el comando cap.open (URL), la dirección URL de puerto de la cámara Ip que se utilizó en el proyecto es rtsp://admin@192.168.61.10:554/user=admin_password=d2gNs1nj_channel=1_stream=1.sdp?real_stream, donde se obtiene la señal de la cámara IP en tiempo real leyendo el video cuadro por cuadro con el comando cap.read() para su posterior procesamiento digital.

```
cap.open(´rtsp://admin@192.168.61.10:554/user=admin_password=d2gNs1nj_channel=1_stream=1.sdp?real_stream´)
```

La Figura 28, muestra la imagen de video adquirida de estacionamiento mediante la conexión entre Spyder y la dirección URL de la cámara IP.

Figura 28

Imagen Estacionamiento adquirida mediante cámara Ip

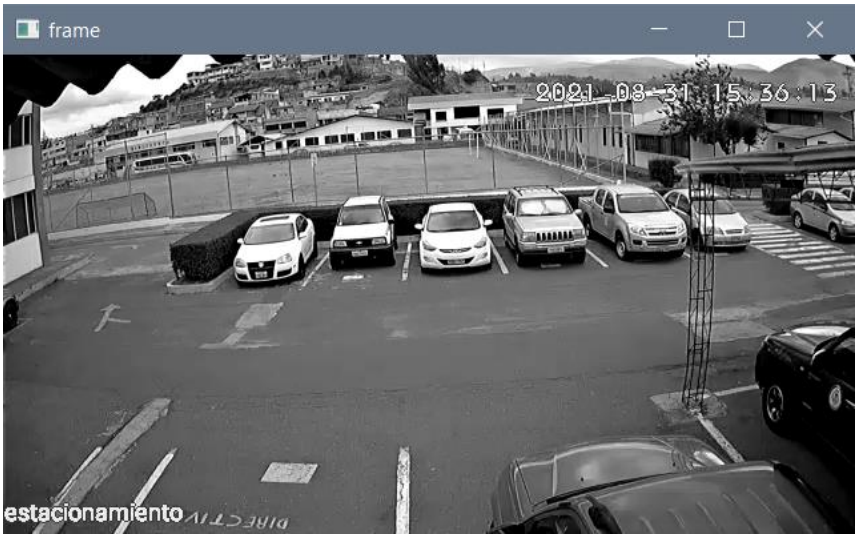
El procesamiento digital de las imágenes se lo realiza aplicando técnicas de visión artificial por medio de los comandos cv2.cvtColor(), para la conversión de la imagen a escala de grises. Aplicamos la transformación morfológica de apertura para delimitar los pixeles de la imagen que estén cerca de los bordes de los objetos que se encuentren en la imagen, de igual manera ayuda a la eliminación de puntos negros o ruido dentro de nuestra región de interés mediante la convolución de los pixeles por una matriz o kernel que nos proporciona una nueva capa con características específicas de la imagen original para diferenciar un objeto en movimiento.

```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
fgmask = cv2.morphologyEx(fgmask, cv2.MORPH_OPEN, kernel)
kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(3,3))
```

La Figura 29, muestra la imagen de video de estacionamiento convertida a escala de grises.

Figura 29

Conversión Imagen Estacionamiento escala de grises



Se aplica sustracción de fondo o background sustractor, la cual es una técnica que permite detectar objetos en movimiento, descartando el resto de la escena o fondo, ya que permanecerá sin movimiento. En el proyecto se aplicó Background sustractor MOG2, el cual es un algoritmo de segmentación de fondo basado en una mezcla gaussiana que proporciona una mejor adaptabilidad en escenas con cambios de iluminación, debido a esto se debe emplear una cámara estática que estará captando toda la escena.

fgbg=cv2.createBackgroundSubtractorMOG2()

Para la segmentación de la imagen aplicamos técnicas de visión artificial con la función findContours(), la cual nos permite encontrar los contornos externos de los objetos en movimiento de una imagen binaria y para dibujar los contornos utilizamos imAux = cv2.drawContours(). En el proyecto se recuperaron los contornos externos mediante la función cv2.RETR_EXTERNAL para almacenarlos como un vector. De igual manera se comprimieron los segmentos horizontales, verticales y diagonales solo dejando sus puntos finales mediante la

función cv2.CHAIN_APPROX_SIMPLE ahorrando memoria al no almacenar puntos redundantes.

```
cv2.rectangle(frame,(0,0),(frame.shape[1],40),(0,0,0),-1)
area_pts = np.array([[0,100], [600,100], [600,400], [0,400]])
imAux = np.zeros(shape=(frame.shape[:2]), dtype=np.uint8)
imAux = cv2.drawContours(imAux, [area_pts], -1, (255), -1)
image_area = cv2.bitwise_and(gray, gray, mask=imAux)
fgmask = fgbg.apply(image_area)
fgmask = cv2.morphologyEx(fgmask, cv2.MORPH_OPEN, kernel)
fgmask = cv2.dilate(fgmask, None, iterations=2)
cnts = cv2.findContours(fgmask, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)[0]
```

A continuación, se muestra en la Figura 30 la detección de movimiento que se aplicara en el proyecto como parte de seguridad para el estacionamiento.

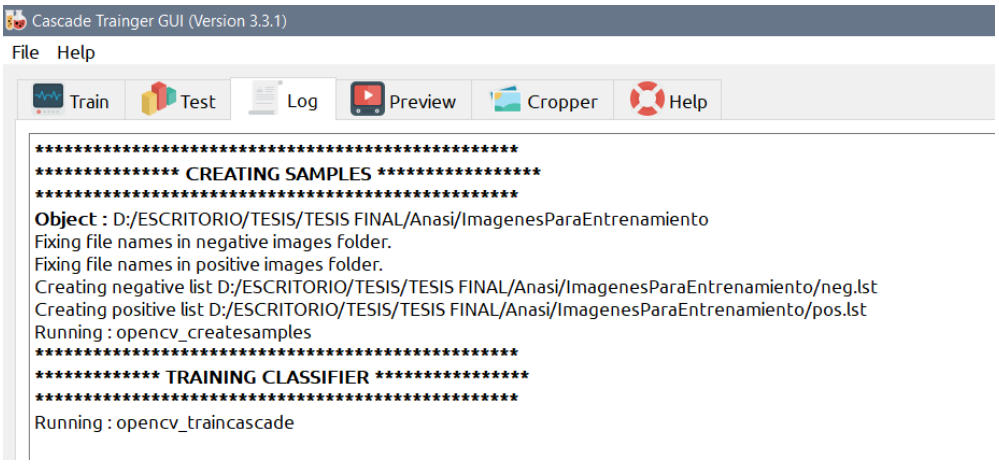
Figura 30

Segmentación de contornos para la detección de movimiento en las plazas de estacionamiento

Para la identificación de vehículos en las plazas de estacionamiento se implementó el clasificador Haar Cascade, entrenado por medio de Cascade Trainer GUI que mediante la carga de imágenes positivas (plaza de estacionamiento con vehículo) e imágenes negativas (plazas de estacionamiento vacías),como se muestra en la Figura 31, creando un archivo con extensión Cascade.xlm que permite reconocer si las plazas de estacionamiento se encuentran disponibles u ocupadas en las regiones de interes señaladas dentro del estacionamiento.

Figura 31(a)

Entrenamiento Cascade Trainer GUI para identificación de automóviles



Una vez obtenido el archivo. xlm se procede a la identificación de vehículos en cada plaza de estacionamiento por separado, ajustando la sensibilidad de reconocimiento en cada una de ellas para reducir los errores posibles debido a la luminosidad variante en el transcurso del día.

```
frames = cv2.resize(frames, (600, 450))
car0 = cv2.CascadeClassifier('cars.xml')
roi0=gray_frames [128:236, 146:221]
blur = cv2.bilateralFilter(roi0,1,150,150)
cars0 = verificar1(roi0,1.1,2)
```

El reconocimiento de vehículos se aplicó a cada plaza de estacionamiento mediante la segmentación de las áreas de interes y la aplicación de clasificador de imágenes pre-entrenado cars.xlm en el estacionamiento ubicado en el club de robótica, como se observa en la Figura 32(a).

Figura 32(a)

Clasificador haar cascade cars.xml aplicado a las plazas de estacionamiento

Firebase proporciona las direcciones URL únicas para cada proyecto a ser desarrollado, permitiendo la identificación de los usuarios de una app que deseen acceder a las características.

Además, nos permite un sistema de autentificación que permite el acceso mediante email y contraseña, gestionando el almacenamiento y descarga de la nube de archivos de los usuarios de forma más rápida y fácil.

A continuación, se describe el código desarrollado en Spyder (TESIS) para la conexión de la base de datos remota de Firebase mediante las direcciones URL.

```
config = {
'apiKey': "AIzaSyCBcjMEzW_DmA_BTIgjMyLaIKXUZ2U42Ac",
'authDomain': "tesis-estacionamiento-6a045.firebaseapp.com",
```

```
'databaseURL' : "https://tesis-estacionamiento-6a045-default-
rtdb.firebaseio.com/",
'projectId': "tesis-estacionamiento-6a045",
'storageBucket': "tesis-estacionamiento-6a045.appspot.com",
'messagingSenderId': "323976556435",
'appId': "1:323976556435:web:ab5af7c57be9d502f48266",
 'measurementId': "G-G68HGKCFWL"
}
```

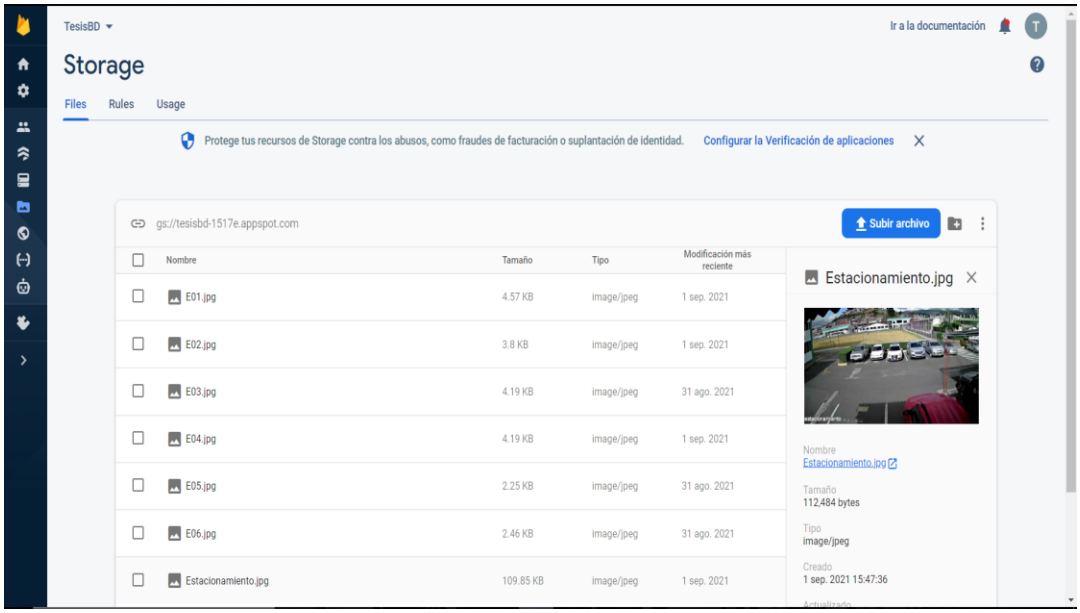
Para la configuración de la base de datos se debe inicializar con el código firebase= pyrebase. initialize_app(config) que permite acceder a las diferentes funciones con las que cuenta Firebase, utilizamos la función Storage con el comando firebase.storage(), para el almacenamiento de las imágenes del estacionamiento que permite la carga y descarga de las imágenes en tiempo real de las plazas de estacionamiento establecidas como se muestra en la Figura 33.

```
firebase = pyrebase.initialize_app(firebaseConfig)
database = firebase.database()
storage = firebase.storage()
estac = ({"E01": cars0,"E02": cars1,"E03": cars2,"E04": cars3,"E05": cars4,
 "E06": cars5})
 database.child("Estacionamientos")
 database.set(estac)
 database.child("Movimiento")
 movi= ({"P”: Cm. getm ()})
 database.set(movi)

estac = ({"E01": cars0,"E02": cars1,"E03": cars2,"E04": cars3,"E05": cars4,
        "E06": cars5})
database.child("Estacionamientos")
database.set(estac)
database.child("Movimiento")
movi=({"P":Cm.getm()})
database.set(movi)
```

Figura 33

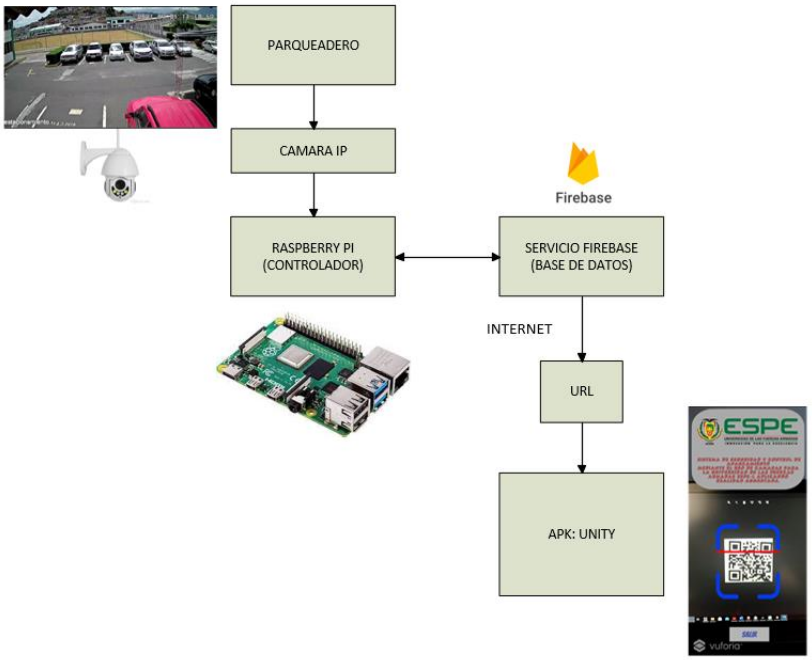
Imágenes del Estacionamiento en el servidor Firebase



En la Figura 34 se presenta el esquema de funcionamiento utilizando el Servidor Firebase para la carga y descarga de las imágenes de las plazas de estacionamientos disponibles u ocupadas desde Raspberry Pi a la aplicación de realidad aumentada desde cualquier sitio o dispositivo móvil.

Figura 34

Esquema de funcionamiento con el Servidor Firebase

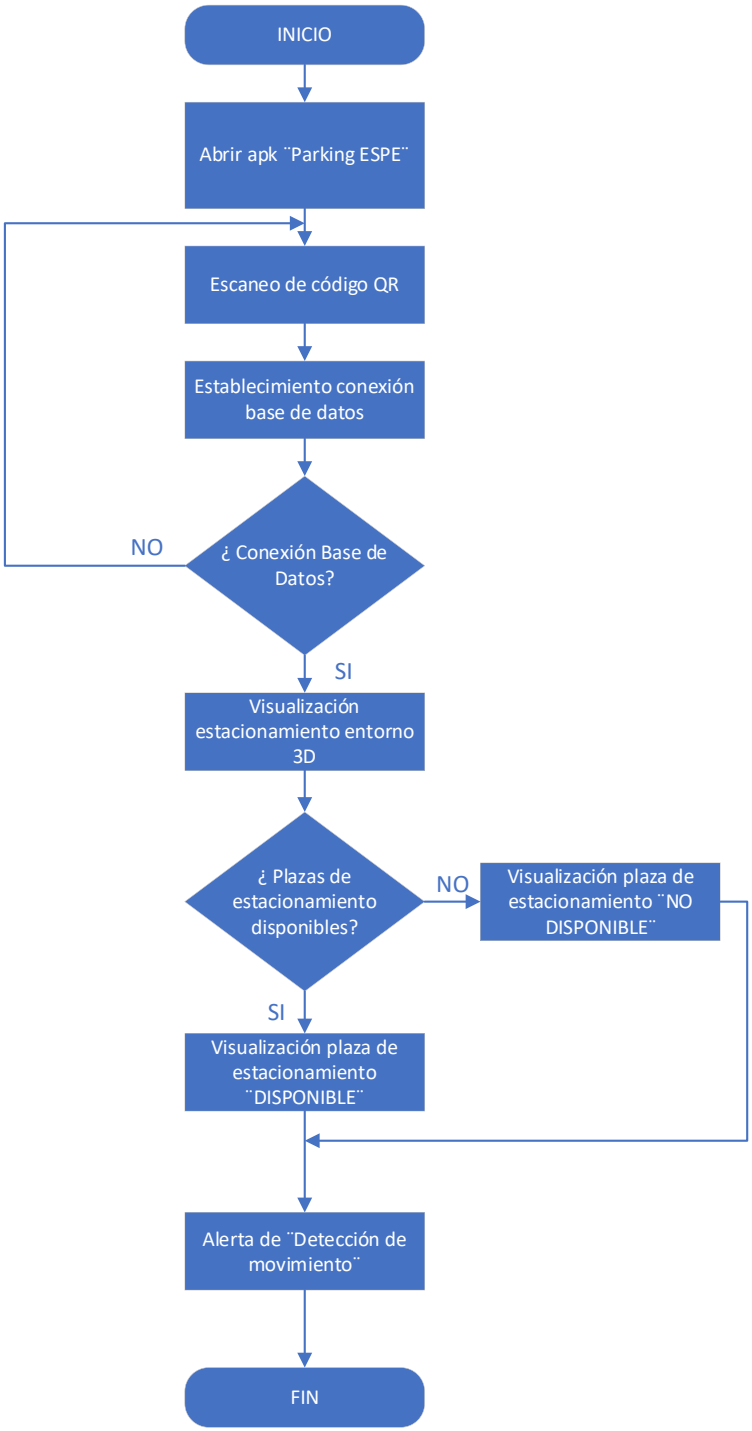


3.9 Estructura de funcionamiento apk estacionamiento unity

En la Figura 35, se muestra en diagrama de flujo para la sección de desarrollo de la aplicación móvil en unity para la detección de movimiento y la identificación de los lugares de estacionamiento disponibles.

Figura 35

Diagrama de flujo para la identificación de movimiento y reconocimiento de plazas de estacionamiento disponibles mediante la aplicación de realidad aumentada



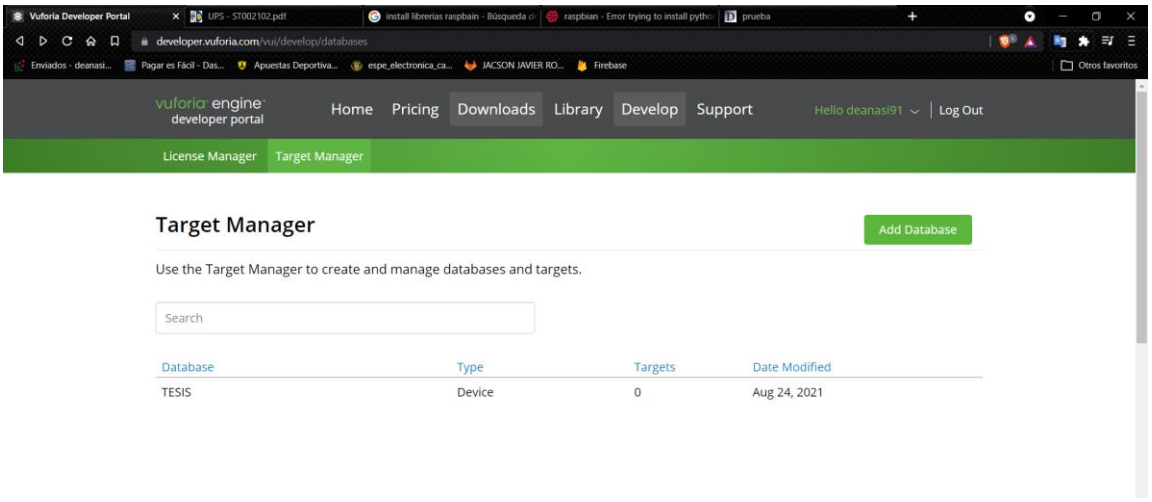
3.9.1 Creación de marca

La marca o image target es la imagen que al ser captadas por la aplicación mostrará la realidad aumentada que se ha programado en ellas.

Para la aplicación del estacionamiento se creó una marca de tipo código QR, la cual se almacenará en la base de datos oficial de vuforia, para ello se debe registrarse en la página web y dar clic en la opción de Develop, se elige la pestaña Target Manager y se presiona el botón Add Database debiendo ingresar el nombre de la base de datos. También se debe seleccionar el tipo, en este caso la dejamos por defecto como se muestra en la Figura 36.

Figura 36

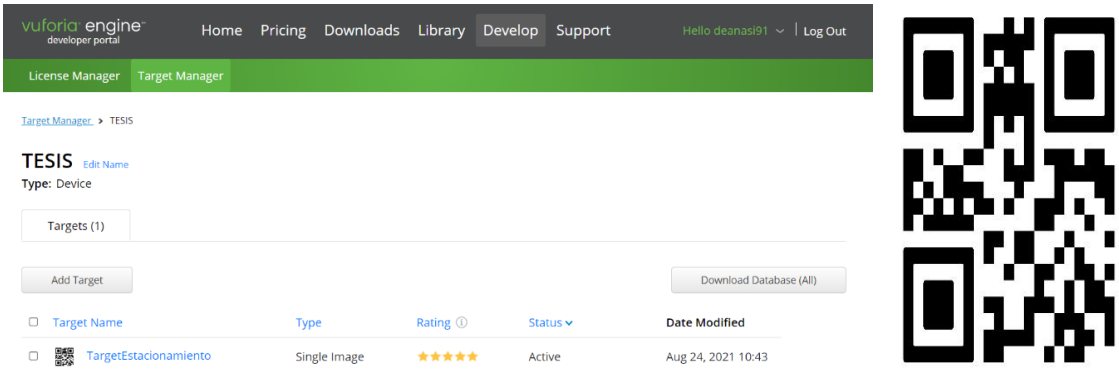
Creación de base de datos vuforia para acceso a estacionamiento



A continuación, seleccionamos la base de datos creada y presionamos el botón Add Target. Aparecerá una ventana para cargar la imagen que puede ser de formato jpg o png con un máximo de tamaño de 2MB, en la Figura 37 se presenta la image target de código QR cargada para el proyecto.

Figura 37

Carga de image target para acceso a estacionamiento

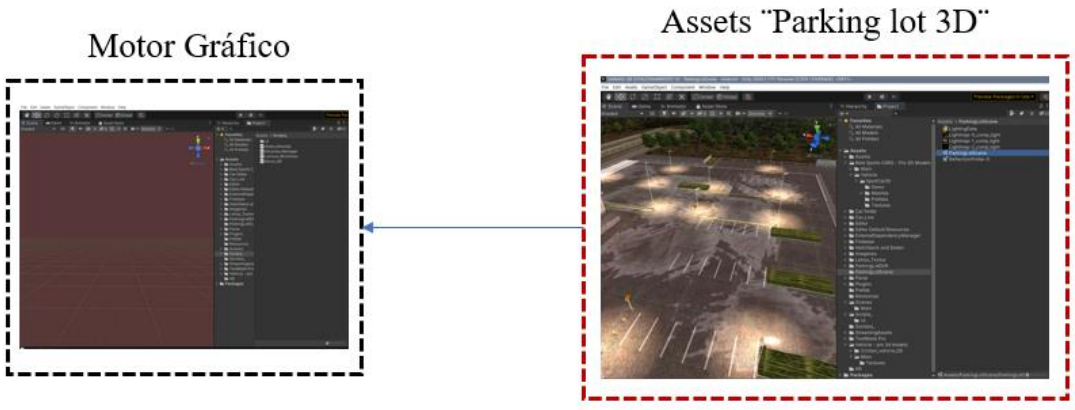


3.9.2 Implementación del entorno virtual en unity 3D

Los modelos tridimensionales para nuestro proyecto son de mucha importancia y pueden ser importados o creados, nos ayudan a poder tener una representación ordenada por medio de un entorno virtual de las plazas de estacionamiento y su disponibilidad.

Figura 38

Integración de componentes gráficos en el software Unity 3D

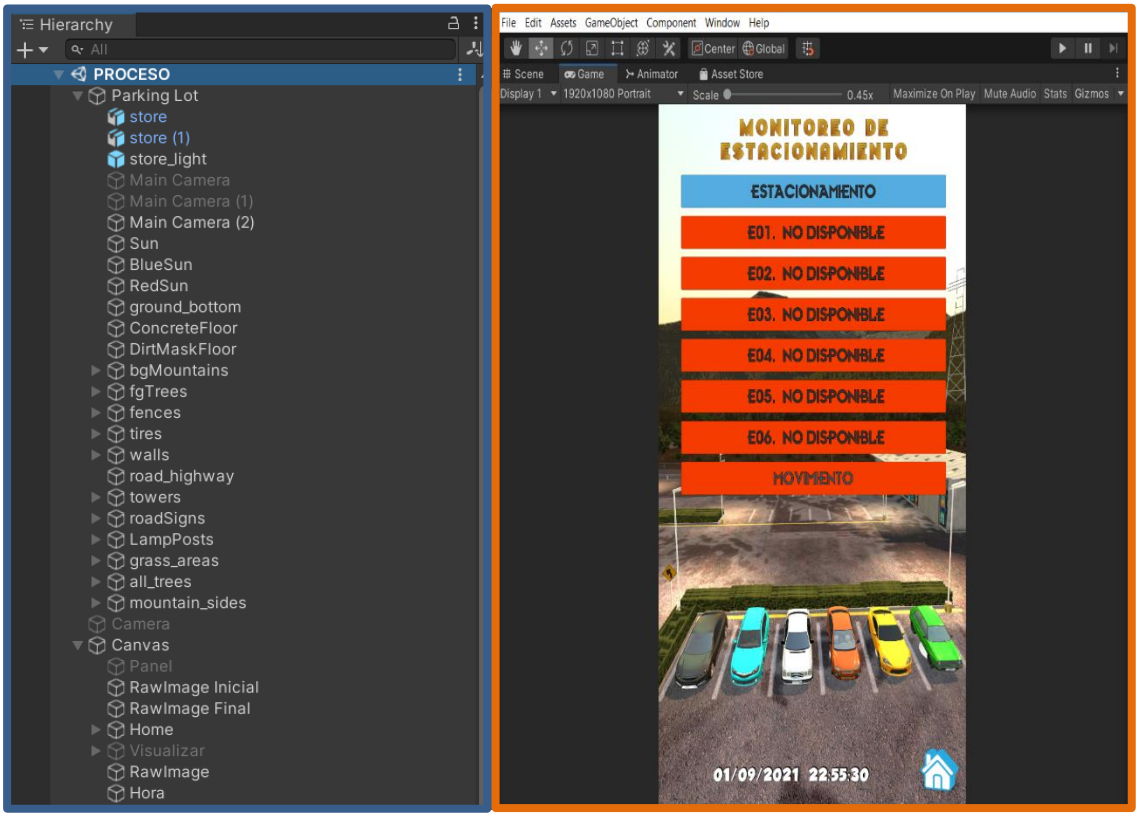


El Assets seleccionado consta de elementos 3D y se denomina “Assets Parking lot”, el cual es necesario arrastrarlos a la escena “Scene” y configurarlo según sea necesario, dicho Assets pasara a formar parte de la Jerarquía “Hierarchy”. Esto opera

de manera similar con los componentes 3D, 2D, códigos, sonidos, animaciones, que están presentes en la Jerarquía y son los que implementan en la aplicación como se muestra en la Figura 39.

Figura 39

Integración de Assets en la escena



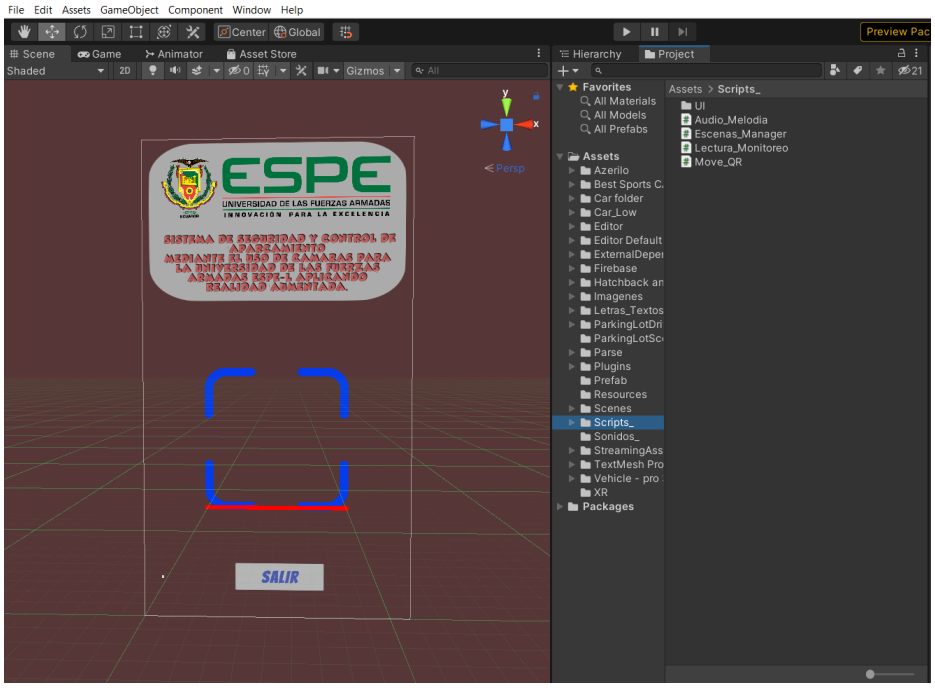
Para formar la aplicación, los Assets se importan en la sección Project, se los arrastra a la jerarquía o a la escena y se los ubica en la sección para el escaneo del código QR y en la representación de los lugares de estacionamiento disponibles u ocupados.

La pantalla de inicio de la aplicación que se observa en la Figura 40 permite el ingreso hacia la pantalla principal mediante el escaneo del código QR, hacia la ventana principal para realizar el monitoreo y gestionamiento de los lugares de estacionamiento,

implementando con image target de vuforia, con un código QR específico para la zona de estacionamiento establecida. De igual manera, al realizar la lectura del código, establece la conexión con la base Firebase que permite la adquisición de imágenes en tiempo real del estacionamiento para su posterior visualización en la ventana principal.

Figura 40

Ventana de inicio aplicación de realidad aumentada mediante de lectura de código QR



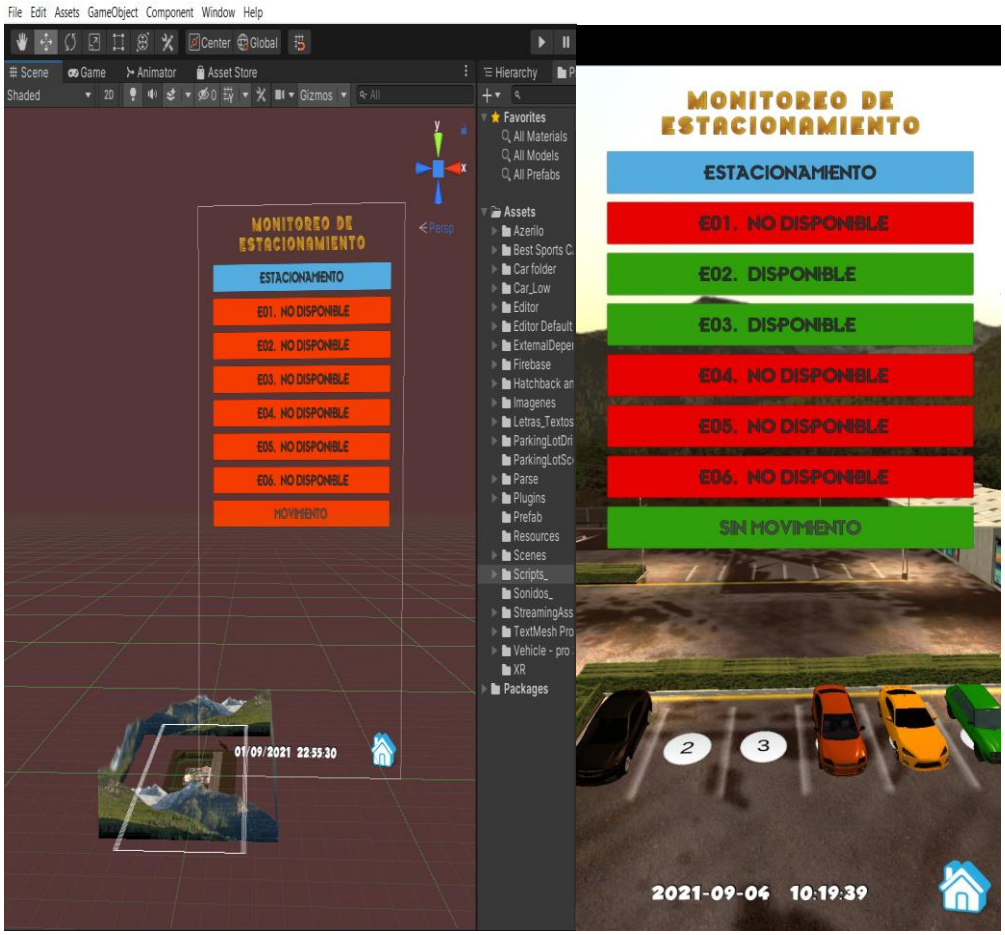
La ventana principal del proyecto contendrá el entorno 3D y script que permitirá identificar las plazas disponibles de estacionamiento y la detección de movimiento en el estacionamiento, por medio de los cambios de estados de los identificadores textuales. De igual manera, permite verificar el estado emitido por medio de la visualización de la imagen de cada una de las plazas de estacionamiento almacenadas en la base de datos Firebase.

En la Figura 41 podemos observar la ventana principal, la cual consta de una tabla donde se especifica el lugar y disponibilidad de cada plaza de estacionamiento en la fecha y hora del ingreso a la aplicación. Los parqueaderos están distribuidos y

representados gráficamente en 3D en la ventana principal, mediante el cual se podrá visualizar el parqueadero monitoreado en tiempo real presentado una imagen de éste. Los indicadores gráficos están destinados a mostrar la disponibilidad de las plazas de estacionamiento al igual que una alerta de movimiento detectado en el mismo.

Figura 41

Ventana pantalla principal aplicación de realidad aumentada Unity



Capítulo IV

4 Análisis Y Resultados

En este capítulo se presentará las pruebas y resultados para establecer eficiencia y precisión del sistema de gestionamiento de parqueadero. Las distintas pruebas se realizan en diferentes horarios con distintos niveles de luminosidad, con el propósito de verificar el correcto funcionamiento del sistema implementado, evaluando las etapas de procesamiento digital de las imágenes para la detección e identificación de plazas de estacionamiento disponibles. Se incluye la detección de movimiento dentro del estacionamiento por medio de una aplicación para dispositivos móviles desarrollada en un software de realidad aumentada (Unity).

4.1 Análisis de la etapa de procesamiento digital de las plazas de estacionamiento

Al momento de aplicar la etapa de procesamiento digital en la secuencia de imágenes del estacionamiento se determina las plazas de estacionamiento y la disponibilidad de cada una de ellas. En caso de presentarse una plaza de estacionamiento libre se emite el mensaje de ¨DISPONIBLE¨ caso contrario emite un mensaje de ¨NO DISPONIBLE¨, de igual manera detecta el movimiento dentro del estacionamiento presentado un mensaje de alerta de ¨MOVIMIENTO DETECTADO¨.

Este sistema evalúa estas etapas contrastando los resultados obtenidos, mostrando una imagen del estacionamiento en tiempo real, al igual que las imágenes de las plazas de estacionamiento disponibles o no disponibles, donde se verifica que la información concuerda con la proporcionada por la aplicación.

Para realizar las diferentes pruebas en el sistema diseñado de parqueadero es necesario mencionar que se lo realizó mediante la aplicación de filtros y clasificadores de imágenes que permiten obtener características específicas de una secuencia de imágenes para determinar o detectar la presencia de movimiento en el estacionamiento

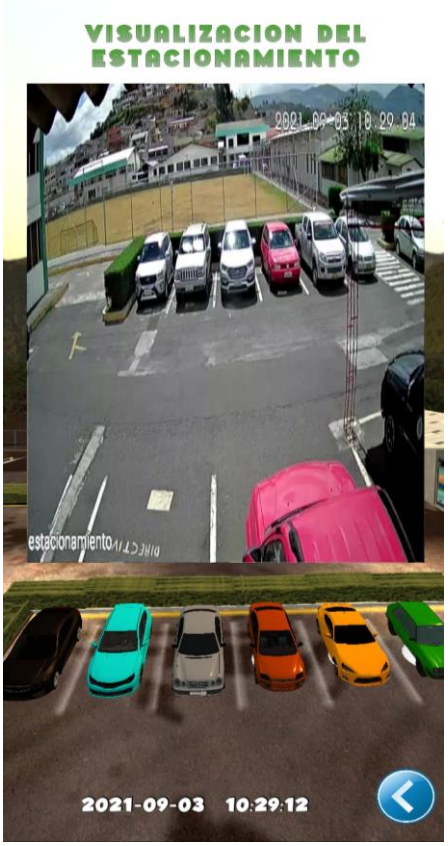
y de igual forma realizar la identificación de vehículos en cada plaza de estacionamiento que nos proporcionará los lugares que están disponibles u ocupados basado en una imagen predeterminada guardada en una base de datos e importada a una aplicación móvil desarrollada en un software de realidad aumentada.

4.2 Pruebas realizadas en el sistema de estacionamiento

En la Figura 42, se presenta el parqueadero con todas las plazas de estacionamiento ocupadas. Como podemos observar, la aplicación presenta todos los lugares de aparcamiento ocupados, para lo cual, a partir de esta imagen se realizan las pruebas respectivas para determinar el correcto funcionamiento del sistema.

Figura 42

Parqueadero con plazas de estacionamiento ocupadas



En la Tabla 2 se presenta en detalle cada plaza de parqueadero con sus respectivas observaciones.

Tabla 2

Estado del estacionamiento de la Figura 41, de las plazas de aparcamiento ocupadas sin error.

Número de Estacionamiento	Estacionamiento ocupado	Estacionamiento disponible	Observaciones
E01	X		Sin error
E02	X		Sin error
E03	X		Sin error
E04	X		Sin error
E05	X		Sin error

En la siguiente sección de imágenes del estacionamiento se presentarán las imágenes obtenidas en el transcurso de todo el día, mostradas en las Figuras 43 (a-e), en las cuales se puede observar los diferentes casos de cada una con sus respectivas tablas descriptivas.

Figura 43(a)

Sistema de estacionamiento con plazas de estacionamiento libres.

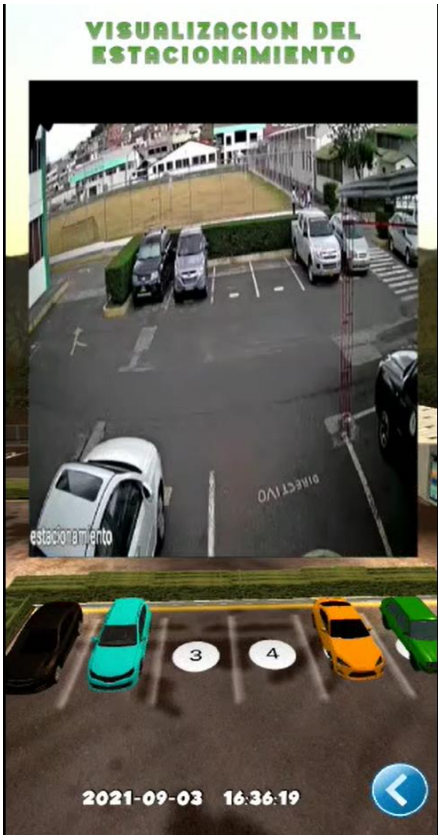


Tabla 3

Estado del parqueadero de la Figura 43(a) con plazas de estacionamiento libres.

Número de Estacionamiento	Estacionamiento ocupado	Estacionamiento disponible	Observaciones
E01	X		Sin error
E02	X		Sin error
E03		X	Sin error
E04		X	Sin error
E05	X		Sin error
E06	X		Sin error

Figura 43(b)

Sistema de estacionamiento con error en una plaza de estacionamiento

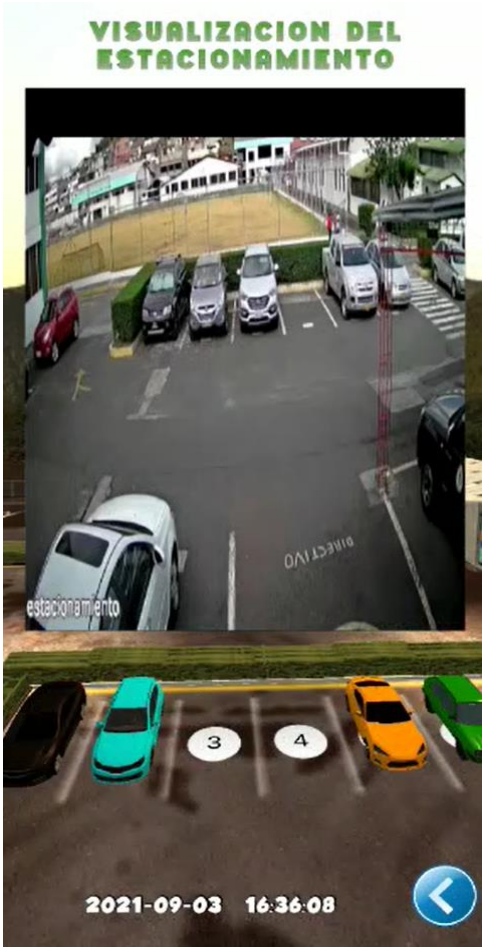


Tabla 4

Estado del parqueadero de la Figura 43(b) con error de detección de plaza de estacionamiento ocupada.

Número de Estacionamiento	Estacionamiento ocupado	Estacionamiento disponible	Observaciones
E01	X		Sin error
E02	X		Sin error
E03	X		Error
E04		X	Sin error
E05	X		Sin error
E06	X		Sin error

En la Tabla 4, se presenta un error al comparar el estado del estacionamiento con los datos proporcionados por la aplicación, como se muestra en la Figura 43(b), el estacionamiento E03 está ocupado, pero la representación en 3D presenta un espacio disponible, esto es ocasionado por las diferentes características de los automóviles y niveles de luminosidad; esta diferencia de lúmenes afecta el reconocimiento de vehículos produciendo un error, los cuales disminuyen al reducirse la intensidad luminosa.

Figura 43(c)

Sistema de estacionamiento con nivel de luminosidad variable

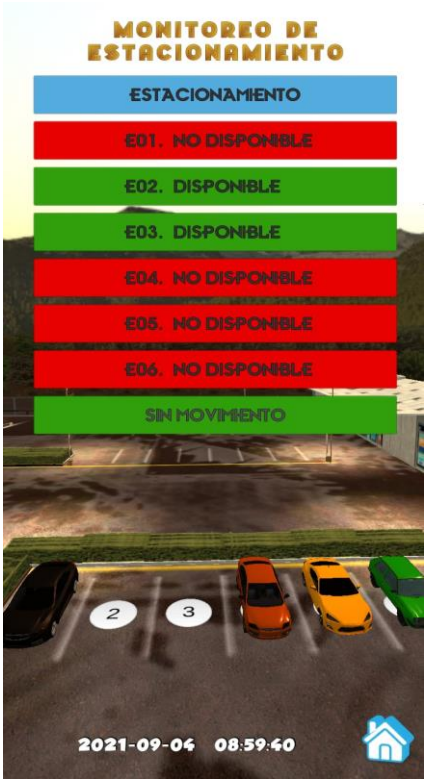


Tabla 5

Estado del parqueadero de la Figura 43(c), con nivel de luminosidad variable

Número de Estacionamiento	Estacionamiento ocupado	Estacionamiento disponible	Observaciones
E01	X		Sin error
E02		X	Sin error
E03		X	Sin error
E04	X		Sin error
E05	X		Sin error
E06	X		Sin error
MOVIMIENTO			NO

Figura 43(d)

Sistema de estacionamiento con nivel de luminosidad variable

Tabla 6

Estado del parqueadero de la Figura 43(d), con nivel de luminosidad variable

Número de Estacionamiento	Estacionamiento ocupado	Estacionamiento disponible	Observaciones
E01	X		Sin error
E02		X	Sin error
E03		X	Sin error
E04	X		Sin error
E05	X		Sin error
E06	X		Sin error
MOVIMIENTO			SI

Figura 43(e)

Sistema de estacionamiento con nivel de luminosidad variable

Tabla 7

Estado del parqueadero de la Figura 43(e), con nivel de luminosidad variable

Número de Estacionamiento	Estacionamiento ocupado	Estacionamiento disponible	Observaciones
E01	X		Sin error
E02	X		Sin error
E03	X		Sin error
E04		X	Sin error
E05	X		Sin error
E06		X	Error
MOVIMIENTO			SI

Al examinar las Figuras 43(b) Y 43(e) y mediante la Tabla 3 y 7 se comprueba que existe un error debido a los niveles luminosidad.

La Tabla 8 indica la modificación en los valores de los pixeles de las imágenes del estacionamiento, cuando varia la luminosidad y afecta al sistema de transmisión del video en tiempo real, los valores a continuación son los valores con niveles de luminosidad alta, media y baja con sus respectivos rangos de valores.

Tabla 8

Valores de pixeles a distintos niveles de luminosidad.

NIVELES DE LUZ SOLAR	RANGO DE PIXELES	
	DE	HASTA
Luminosidad Alta	180	255
Luminosidad Media	71	181
Luminosidad Baja	0	70

Se puede apreciar en las Figuras 44(f, g, h), a distintos niveles de luminosidad alta, media y baja respectivamente, las cuales se utilizarán para el análisis de las plazas de estacionamientos en el transcurso del día. Para nuestro caso de análisis, se utilizará el estacionamiento E04 con diferentes cambios de luminosidad.

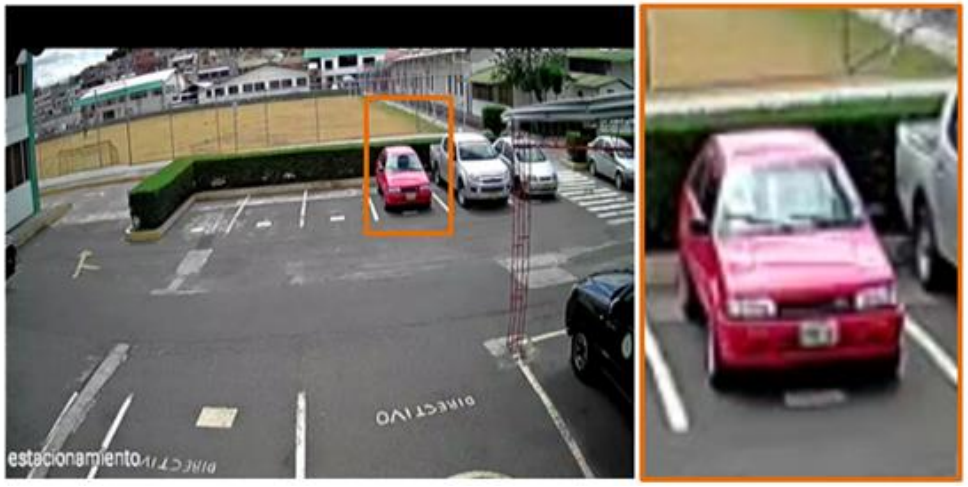
Figura 44(f)

Imagen estacionamiento luminosidad alta

SECCIONE DEFINIDA PARA EL PROCESAMIENTO DE IMAGENES

Figura 44(g)

Imagen estacionamiento luminosidad media



SECCIONE DEFINIDA PARA EL PROCESAMIENTO DE IMAGENES

Figura 44(h)

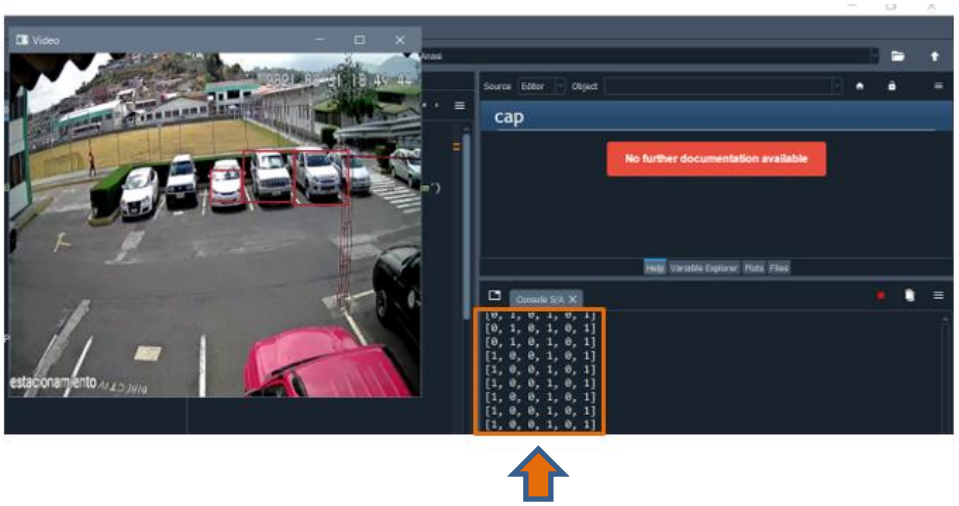
Imagen estacionamiento luminosidad baja

SECCIONE DEFINIDA PARA EL PROCESAMIENTO DE IMAGENES

Los valores de los pixeles varían entre 0 a 255, una respuesta óptima del sistema se presenta cuando el nivel de pixeles está en un nivel bajo. Se debe recordar que el sistema está diseñado utilizando un clasificador Haar Cascade para el reconocimiento de automóviles en las plazas de estacionamientos que identifican la disponibilidad de cada una de ellas, generando un nivel 1 si se encuentran disponibles y 0 si está ocupada, como se analiza en la Figura 45.

Figura 45

Detección de automóvil en plazas de estacionamientos



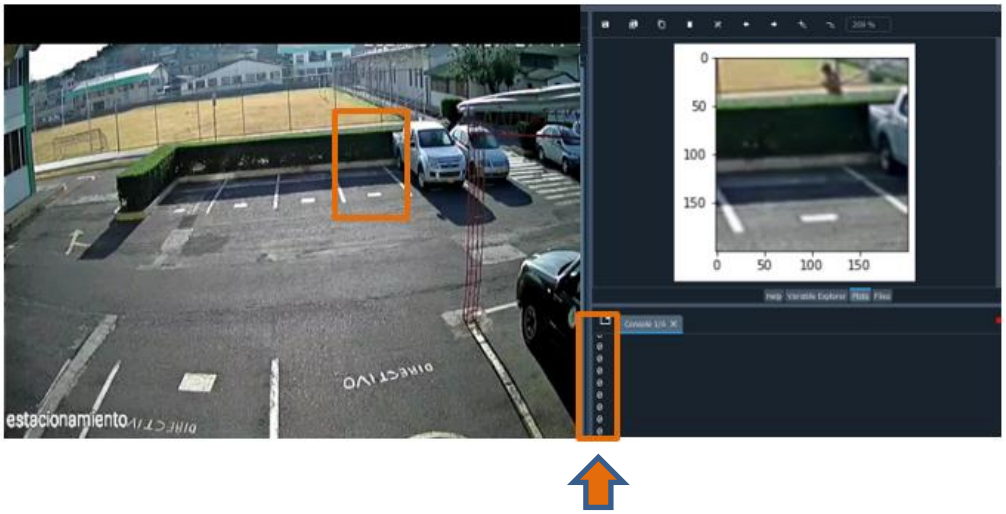
ESTACIONAMIENTO DISPONIBLES
Y OCUPADOS

La identificación de vehículos de la secuencia de imágenes anterior, presenta los errores por la variación niveles de luminosidad modificando los valores de pixeles de cada imagen, provocando errores en la detección de vehículos en las plazas de estacionamientos para determinar su disponibilidad.

En la Figura 46 se realizó la prueba a partir de las 9 am donde la intensidad de luminosidad es alta en la zona de estacionamiento establecida, debido a esto los valores de pixeles se modifican y el sistema detecta que la plaza de estacionamiento se encuentra OCUPADA cuando en realidad no lo está.

Figura 46

Sistema parqueadero intensidad de luz alta

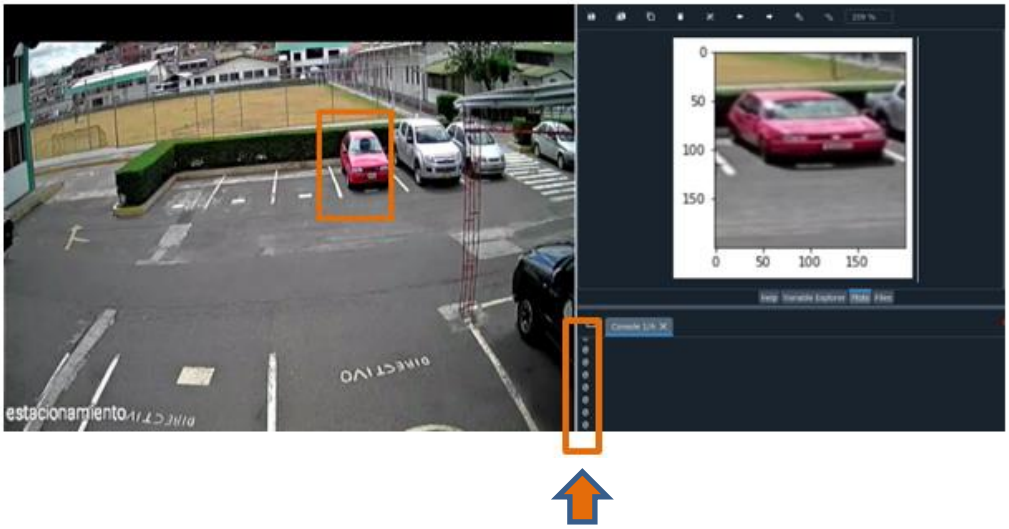


ESTACIONAMIENTO OCUPADO

En la Figura 47, se puede observar el sistema a una variación de luz media, por lo que el valor de los pixeles disminuye para las plazas de estacionamientos por lo que la detección de vehículos muestra el estacionamiento ocupado lo que es correcto.

Figura 47

Sistema parqueadero intensidad de luz media

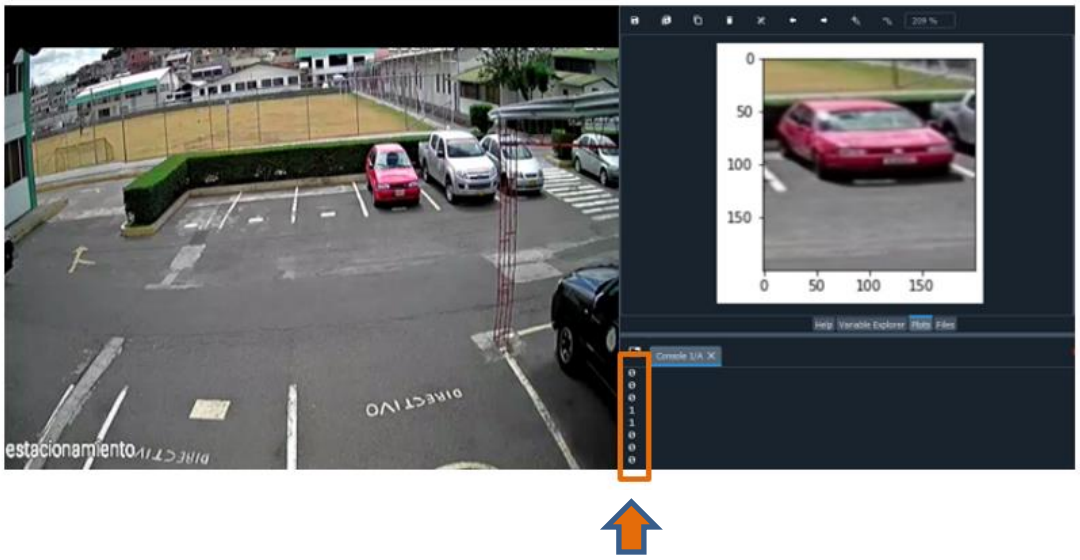


ESTACIONAMIENTO OCUPADO

De igual manera, en la Figura 48, se observa el sistema de parqueadero a una variación de luz baja al acercarse las horas de la noche, por lo que el valor de los pixeles disminuye significativamente para las plazas de estacionamientos por lo que la detección de vehículos tiene una variación minima.

Figura 48

Sistema Parqueadero intensidad de luz baja



ESTACIONAMIENTO CON CAMBIOS DE ESTADO

4.3 Pruebas a diferentes niveles de luminosidad

Por los errores mostrados en las pruebas a diferentes niveles de luminosidad se realizará varias pruebas para determinar los problemas ocasionados debido a la variación de niveles de luminosidad al momento de realizar la detección de vehículos en las plazas de estacionamientos en horarios diferentes, como se presenta en las Figuras 49(a-f).

Figura 49(a)

Verificación del sistema de estacionamiento con un nivel de luminosidad alta, tomada a las 08h59

En la Figura 49(a), se observa que el nivel de intensidad luminosa alta afecta el reconocimiento como se muestra en la Tabla 9, que presenta los errores en los parqueaderos E01 y E02 los cuales se encuentran disponibles y el sistema los presenta como plazas de estacionamiento ocupadas.

Tabla 9

Estado del parqueadero de la Figura 49(a)

Número de Estacionamiento	Estacionamiento ocupado	Estacionamiento disponible	Observaciones
E01	X		Error
E02	X		Error
E03		X	Sin error
E04	X		Sin error
E05	X		Sin error
E06	X		Sin error
MOVIMIENTO			NO

Figura 49(b)

Verificación del sistema de estacionamiento con un nivel de luminosidad alta, tomada a las 10h19

En la Figura 49(b), se observa que el nivel de intensidad luminosa es media-alta afecta el reconocimiento como se muestra en la Tabla 10, que presenta los errores en los parqueaderos E01 el cual se encuentra disponible y el sistema lo presenta como plaza de estacionamiento ocupada.

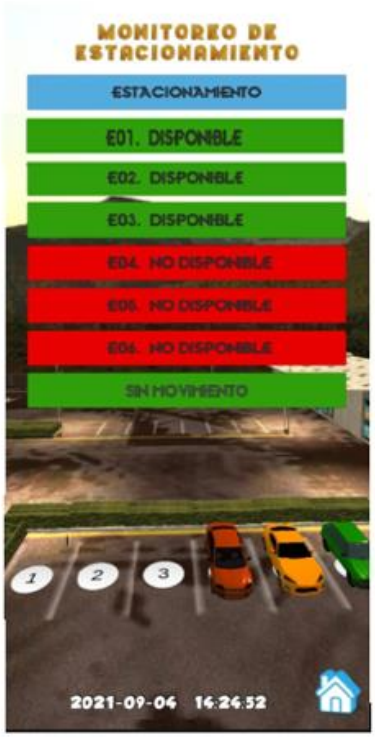
Tabla 10

Estado del parqueadero de la Figura 49(b)

Número de Estacionamiento	Estacionamiento ocupado	Estacionamiento disponible	Observaciones
E01	X		Error
E02		X	Sin error
E03		X	Sin error
E04	X		Sin error
E05	X		Sin error
E06	X		Sin error
MOVIMIENTO			NO

Figura 49(c)

Verificación del sistema de estacionamiento con un nivel de luminosidad alta, tomada a las 14h24



En la Figura 49(c), se observa que el nivel de intensidad luminosa media no afecta el reconocimiento como se muestra en la Tabla 11, que se presenta sin errores en los parqueaderos encuentran disponible u ocupados del sistema.

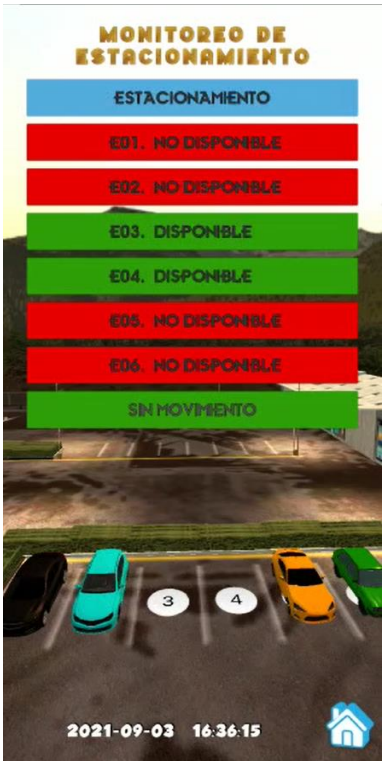
Tabla 11

Estado del parqueadero de la Figura 49(c)

Número de Estacionamiento	Estacionamiento ocupado	Estacionamiento disponible	Observaciones
E01		X	Sin error
E02		X	Sin error
E03		X	Sin error
E04	X		Sin error
E05	X		Sin error
E06	X		Sin error
MOVIMIENTO			NO

Figura 49(d)

Verificación del sistema de estacionamiento con un nivel de luminosidad alta, tomada a las 16h36



En la Figura 49(d), se observa que el nivel de intensidad luminosa es media-baja afecta el reconocimiento como se muestra en la Tabla 12, que se presenta errores en los parqueaderos E01 y E03 que se encuentran disponible y ocupado respectivamente y el sistema los presenta erróneamente.

Tabla 12

Estado del parqueadero de la Figura 49(d)

Número de Estacionamiento	Estacionamiento ocupado	Estacionamiento disponible	Observaciones
E01		X	Error
E02		X	Sin error
E03	X		Error
E04		X	Sin error
E05	X		Sin error
E06	X		Sin error
MOVIMIENTO			NO

Figura 49(e)

Verificación del sistema de estacionamiento con un nivel de luminosidad alta, tomada a las 18h06

En la Figura 49(e), se observa que el nivel de intensidad luminosa es bajo y afecta el reconocimiento como se muestra en la Tabla 13, se presenta los errores en los parqueaderos E01 y E02 que se encuentran disponibles el sistema los presenta erróneamente como ocupados.

Tabla 13

Estado del parqueadero de la Figura 49(e)

Número de Estacionamiento	Estacionamiento ocupado	Estacionamiento disponible	Observaciones
E01	X		Error
E02	X		Error
E03		X	Sin error
E04		X	Sin error
E05	X		Sin error
E06	X		Sin error
MOVIMIENTO			NO

Figura 49(f)

Verificación del sistema de estacionamiento con un nivel de luminosidad alta, tomada a las 18h37

En la Figura 49(f), se apreciar el estacionamiento con nivel de intensidad luminosa baja afecta el reconocimiento como se muestra en la Tabla 14, que se presenta errores en los parqueaderos E01 y E02 que se encuentran disponibles el sistema los presenta erróneamente como ocupados.

Tabla 14

Estado del parqueadero de la Figura 49(f)

Número de Estacionamiento	Estacionamiento ocupado	Estacionamiento disponible	Observaciones
E01	X		Error
E02	X		Error
E03		X	Sin error
E04		X	Sin error
E05	X		Sin error
E06	X		Sin error
MOVIMIENTO			NO

De acuerdo al análisis de la secuencia de imágenes de la Figura 49(a-f) se puede apreciar que existen errores, producidos por las variaciones del nivel de intensidad luminosa que se obtiene en el transcurso del día y que es demasiado variable, lo cual, por las características de la cámara y factores de luminosidad, ocasiona disminución de visibilidad al momento de la adquisición de las imágenes del estacionamiento.

Mediante las pruebas realizas en las secciones anteriores, en la Tabla 15 muestra la precisión del sistema de gestionamiento para el estacionamiento, que se realizó mediante muestras tomadas de cada plaza de estacionamiento con distintos niveles de luminosidad y en horarios diferentes, para determinar el error presentado en el momento de detectar el vehículo estacionado en los lugares de parqueo.

Tabla 15

Porcentaje de error de cada plaza de estacionamiento de acuerdo al nivel de luminosidad

Número de Estacionamiento	Luminosidad alta	Luminosidad media	Luminosidad baja
E01	70%	5%	20%
E02	60%	5%	20%
E03	20%	0%	15%
E04	40%	0%	15%
E05	10%	0%	10%
E06	30%	10%	20%
Movimiento	5%	0%	5%

La Figura 50 y la Figura 51 se puede observar el error en el sistema de gestionamiento de parqueaderos para cada plaza de estacionamiento con variación de luminosidad (alto, medio, bajo).

Figura 50

Gráfico porcentaje de error de sistema de estacionamiento

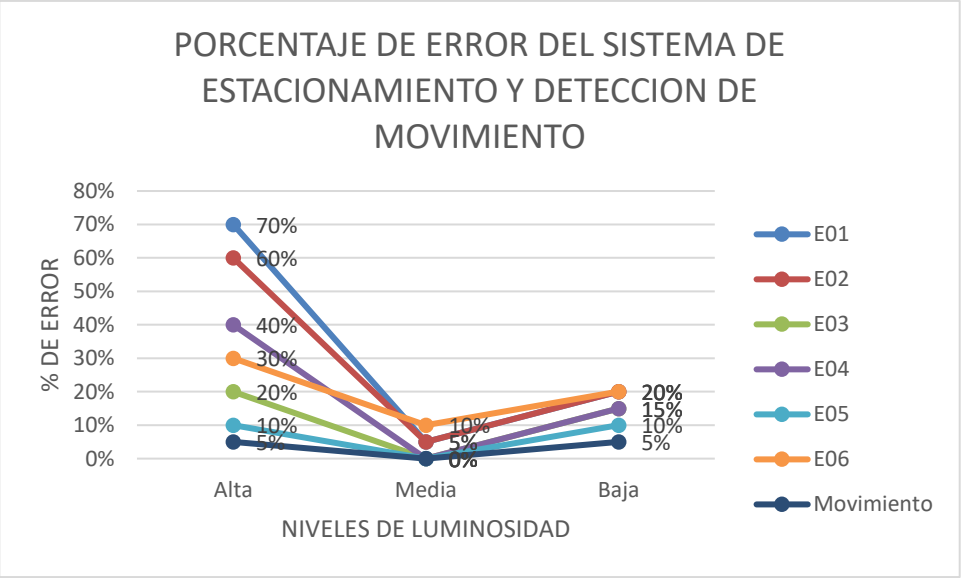
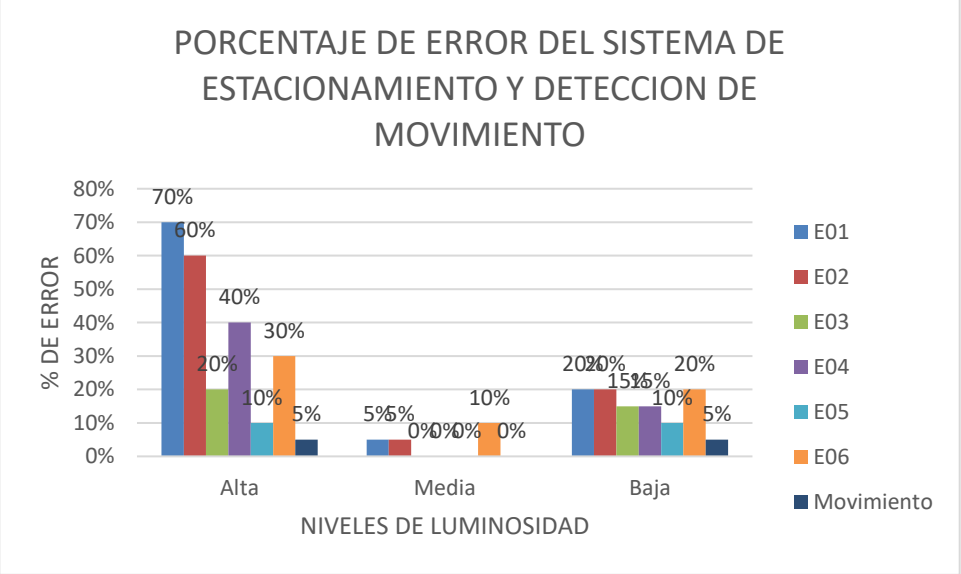


Figura 51

Gráfico porcentaje de error de sistema de estacionamiento



Los datos tomados para realizar los gráficos de errores porcentuales fueron obtenidos mediante la aplicación de diversas pruebas en un horario de 8h00 am hasta las 19h00 pm con variación de luminosidad incluyendo las condiciones ambientales.

Los errores presentados en la etapa de pruebas a distintos niveles de luminosidad sin filtros tienen una eficiencia de 83.81% para determinar la disponibilidad de las plazas de estacionamiento y detectar moviento dentro del mismo en niveles de luminosidad variable y en variación de condiciones climáticas.

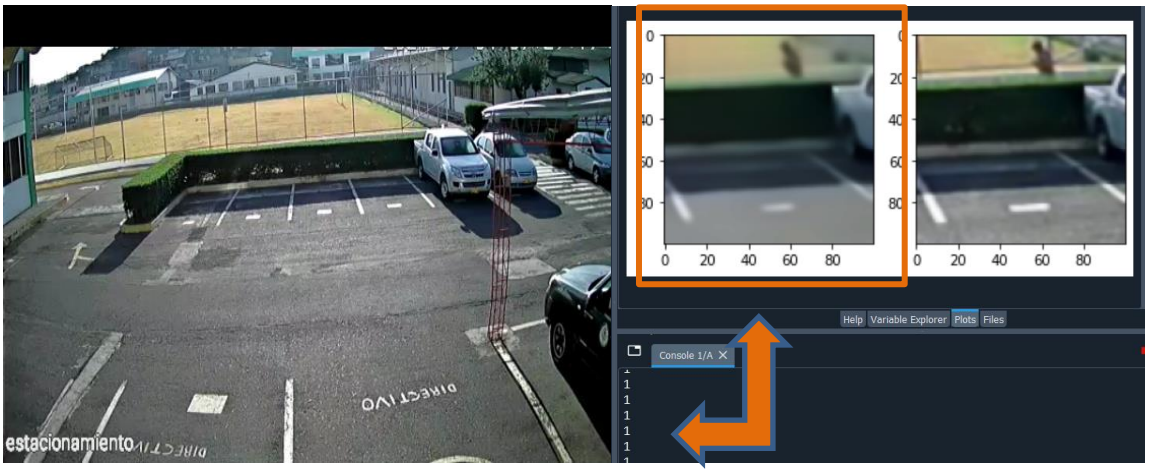
4.4 Pruebas a diferentes niveles de luminosidad aplicando filtro bilateral

Al realizar el análisis con las diferentes intensidades luminosas, se determina que el sistema trabaja sin problemas cuando la intensidad es media, con una confiabilidad del 100%, debido a este motivo se procedió a realizar la corrección de los errores por medio de la aplicación de un filtro bilateral para la disminución de ruido preservando los bordes de los vehículos de las plazas de estacionamiento, al aplicar este filtro el valor de cada pixel de la imagen sustituido por la media de los valores de los píxeles cercanos, proporcionándonos una intensidad media constante en el transcurso del día a la tarde.

A continuación, se muestra las secuencias de imágenes con la aplicación del filtro bilateral para corregir los errores presentados según la intensidad de luminosidad como se presentan en la Figura 52 (a-c).

Figura 52(a)

Verificación del sistema con un nivel de luminosidad alta



ESTACIONAMIENTO DISPONIBLE

APLICANDO FILTRO BILATERAL

En la Figura 52(a) se puede apreciar en la plaza de estacionamiento E04 con una intensidad de luminosidad alta y muestra que se encuentra disponible por lo que el sistema no produce errores, lo cual se muestra en la Figura 50(a), presentando en la Tabla 16, tomada desde la aplicación móvil diseña para el proyecto.

Figura 52(a)

Tabla de estados de estacionamiento desde aplicación de realidad aumentada sin errores.

Tabla 16

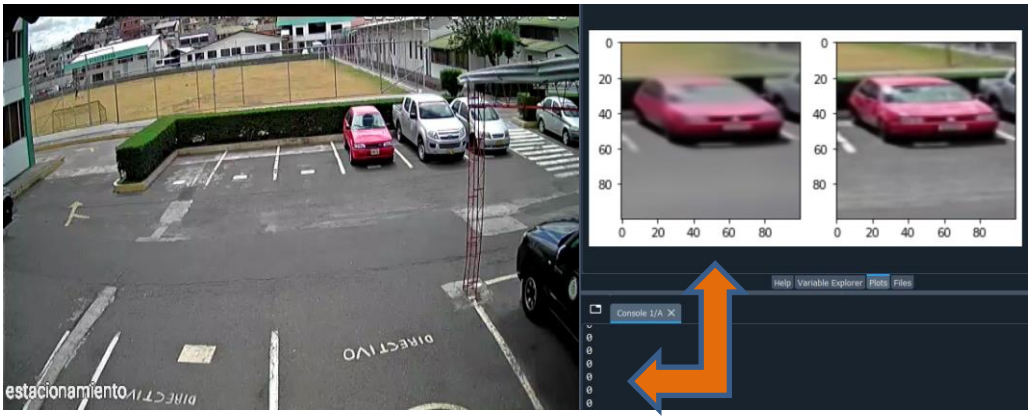
Estado del parqueadero de la Figura 52(a)

Número de Estacionamiento	Estacionamiento ocupado	Estacionamiento disponible	Observaciones
E01		X	Sin error
E02		X	Sin error
E03		X	Sin error
E04		X	Sin error
E05	X		Sin error
E06	X		Sin error
MOVIMIENTO			NO

Como se puede observar el sistema no presenta errores al exponerse a una intensidad de luz alta, aplicando el filtro bilateral que reduce significativamente los valores de los pixeles, lo cual ayuda a la detección de vehículos ubicados en las plazas de estacionamiento.

Figura 52(b)

Verificación del sistema con un nivel de luminosidad media



ESTACIONAMIENTO OCUPADO

APLICANDO FILTRO BILATERAL

En la Figura 52(b), se puede apreciar el parqueadero con una intensidad de luminosidad media, en la Tabla 17 obtenida de la aplicación móvil se observa sin errores en los parqueaderos durante media luminosidad.

Figura 52(b)

Tabla de estados de estacionamiento desde aplicación de realidad aumentada sin errores.

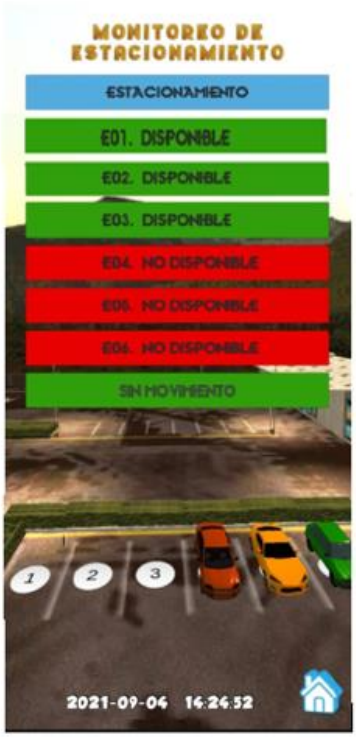


Tabla 17

Estado del parqueadero de la Figura 52(b)

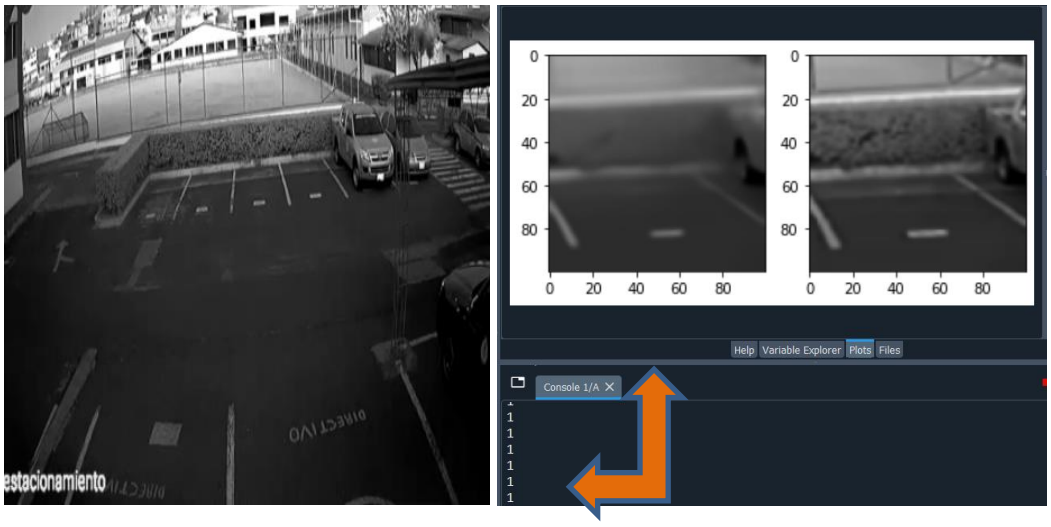
Número de Estacionamiento	Estacionamiento ocupado	Estacionamiento disponible	Observaciones
E01		X	Sin error
E02		X	Sin error
E03		X	Sin error
E04	X		Sin error
E05	X		Sin error
E06	X		Sin error
MOVIMIENTO			NO

En la Figura 52(b), se puede apreciar el estacionamiento con una intensidad luminosa media, en la Tabla 17, obtenida desde la aplicación móvil se observa sin errores en los parqueaderos con luminosidad media, lo cual verifica que las condiciones

óptimas para el correcto funcionamiento eficiente del sistema de estacionamiento es una luminosidad media reduciendo el error al mínimo posible.

Figura 52(c)

Verificación del sistema con un nivel de luminosidad baja.



ESTACIONAMIENTO DISPONIBLE
APLICANDO FILTRO BILATERAL

En la Figura 52 (c), se puede apreciar el estacionamiento con una intensidad de luminosidad baja, en la Tabla 18, obtenida de la aplicación móvil se observa mínimos errores en los parqueaderos durante baja luminosidad.

Figura 52(c)

Tabla de estados de estacionamiento desde aplicación de realidad aumentada sin errores

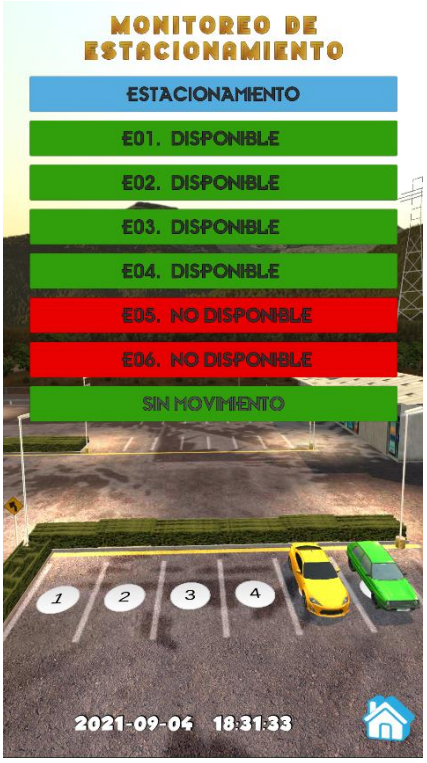


Tabla 18

Estado del parqueadero de la Figura 52(c)

Número de Estacionamiento	Estacionamiento ocupado	Estacionamiento disponible	Observaciones
E01		X	Sin error
E02		X	Sin error
E03		X	Sin error
E04		X	Sin error
E05	X		Sin error
E06	X		Sin error
MOVIMIENTO			NO

En la Figura 52 (c), se observa el parqueadero con una intensidad luminosa baja, en la Tabla 18 obtenida de la aplicación se observa mínimos errores en los parqueaderos con luminosidad baja, demostrando que mientras más se aproxime la noche el sistema tiende a perder eficiencia por la pérdida de luminosidad en su totalidad

como se muestran en los gráficos porcentuales de error aplicando filtro bilateral en la Figura 53 y la Figura 54.

Tabla 19

Porcentaje de error del sistema de estacionamiento aplicando filtro bilateral

	Luminosidad alta	Luminosidad media	Luminosidad baja
E01	5%	1%	10%
E02	5%	0%	5%
E03	0%	0%	5%
E04	0%	1%	5%
E05	5%	0%	2%
E06	5%	1%	10%
Movimiento	0%	0%	10%

Figura 53

Gráfico porcentaje de error de sistema de estacionamiento aplicando filtro bilateral

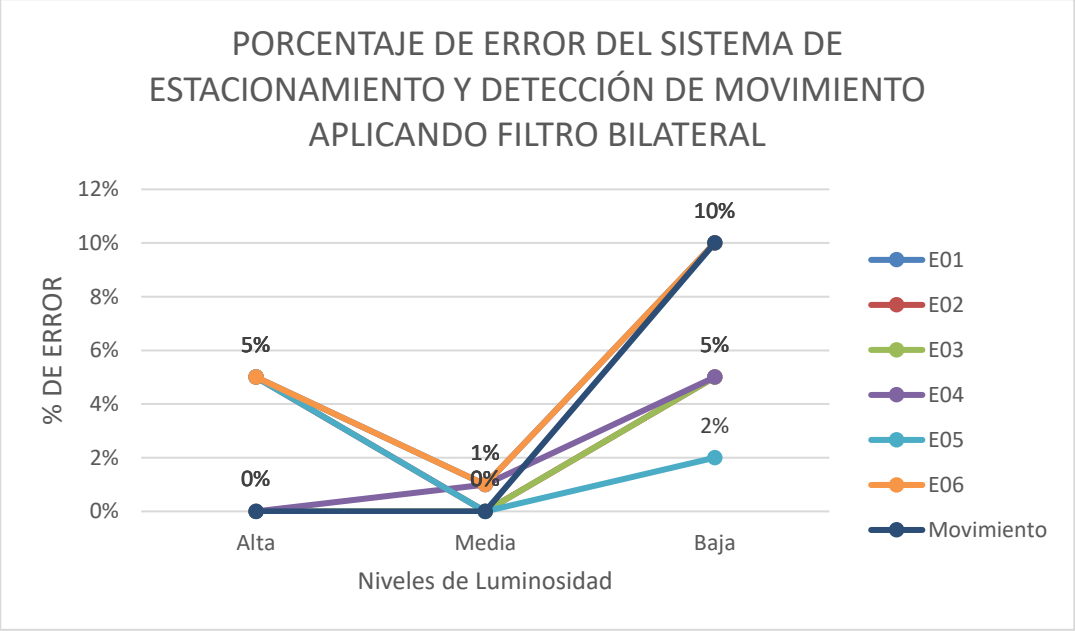
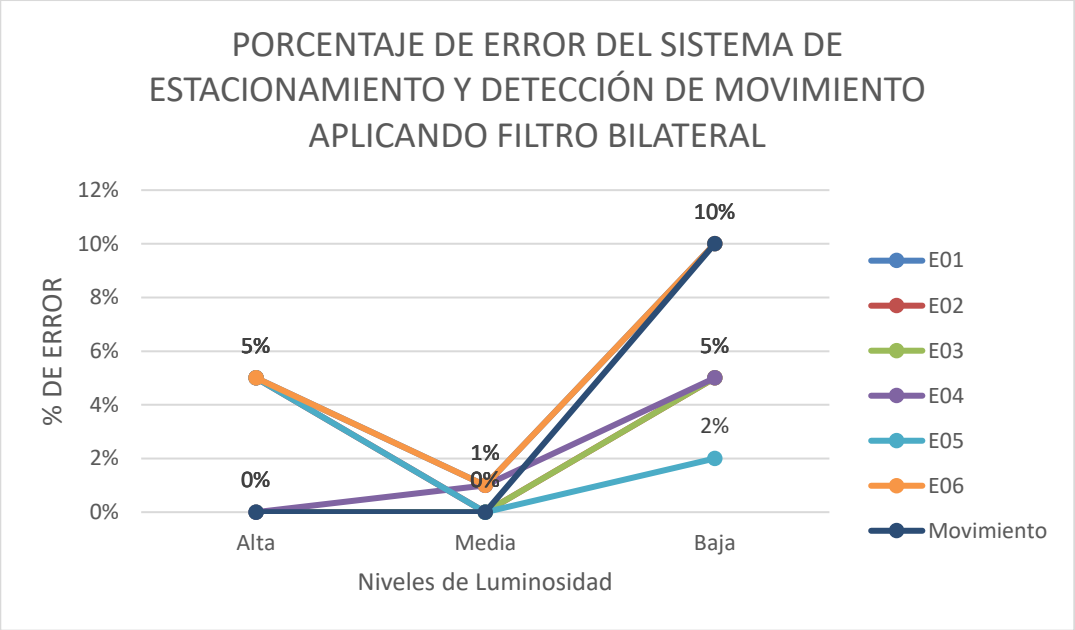


Figura 54

Gráfico porcentaje de error de sistema de estacionamiento aplicando filtro bilateral



En comparación con los errores presentados en la etapa de pruebas a distintos niveles de luminosidad sin filtros. La eficiencia del sistema al aplicar filtro bilateral aumenta a un 96.66%, para determinar la disponibilidad de las plazas de estacionamiento y detectar moviento dentro del mismo en niveles de luminosidad variable y en variación de condiciones climáticas.

4.5 Comprobación de la hipótesis

Para comprobar la hipótesis establecida se realizó un análisis dividido en dos partes.

La primera parte consiste en evaluar el proceso de reconocimiento de vehículos en las plazas de estacionamiento el cual se lo realiza en la sección 4.2, para esto se realizó algunas pruebas en diferentes horarios del día. Al contrastar visualmente los estados de las plazas de estacionamiento, con los resultados de la etapa de identificación de vehículos por medio de la aplicación, se tiene un buen porcentaje de reconocimiento por

parte del sistema, aunque estos resultados variar según las diferentes características de los automóviles y niveles de luminosidad.

La segunda parte es evaluar el clasificador Haar Cascade, con y sin filtro bilateral, este análisis se lo realizó en el apartado 4.3 y 4.4 en el cual se realizó pruebas de reconocimiento con intensidad de luminosidad alta, media y baja, obteniendo un porcentaje de precisión del sistema de 0.838 (83.81%) y 0.966 (96.66%) respectivamente, así se concluye que el clasificador tiene un alto grado de exactitud de reconocimiento de vehículos en las plazas de estacionamientos.

4.6 Alcances

• El sistema desarrollado e implementado realiza la detección de movimiento y el reconocimiento de vehículos, convirtiéndose en un sistema de gran ayuda para la administración de las plazas de estacionamientos, al momento de identificar los estacionamientos disponibles u ocupados.

• La utilización de la aplicación móvil es fluida, la cual posee indicadores textuales que proporcionan información al usuario, presentando solo la información necesaria, describiéndola como simple, eficiente y concreta. Su función es visualizar y mostrar la disponibilidad de cada plaza de estacionamiento.

4.6 Limitaciones

• El sistema depende en gran porcentaje de la intensidad de luminosidad, existiendo casos en los cuales no se obtendrá una respuesta concisa del estado de las plazas de estacionamiento mediante el indicar textual.

• Debido a las diferentes características de los automóviles y dependiendo del ángulo que se realiza el reconocimiento, el sistema emite un mensaje con las plazas de estacionamiento disponibles u ocupadas con una eficacia del 96.66%, para evitar estos posibles errores el usuario puede verificar la información visualizando cada plaza de estacionamiento en tiempo real.

• Para aplicar un procesamiento de imágenes basado en métodos de visión artificial para la aplicación del clasificador Haar Cascade se requiere de un equipo computacional con una velocidad de procesamiento alta y de buena calidad llegando a tener un valor de alto costo.

Capítulo V

5 Conclusiones y Recomendaciones

5.1 Conclusiones

- El sistema de seguridad y gestionamiento de lugares de estacionamiento utilizando procesamiento digital de imágenes junto con métodos de visión artificial, permiten la detección de movimiento y la identificación de las plazas de parqueo disponibles en las zonas del estacionamiento designadas.
- Mediante el almacenamiento de las imágenes en tiempo real en una base de datos, el sistema se puede administrar por una aplicación de realidad aumentada la cual, al realizar la petición para la visualización del estacionamiento o plaza específica del mismo, proporciona un monitoreo constante y un gestionamiento de plazas de estacionamiento eficiente en tiempo real.
- El sistema se desarrolló totalmente en software libre, para esto se utilizó la plataforma Anaconda Navigator para administrar las aplicaciones, paquetes y entornos, dedicados al desarrollo de lenguaje Python, utilizando como editor de código el software Spider y sus diferentes librerías para realizar: análisis de datos, procesamiento digital de imágenes, reconocimiento de imágenes y almacenamiento de archivos a base de datos en tiempo real. Para el desarrollo de la aplicación de realidad aumentada se utilizó el software Unity el cual permite la edición y creación de entornos 3D para representación y administración de sistemas de automatización amigables con el usuario.
- La etapa de procesamiento digital de imágenes es realizada por medio de una computadora de placa Raspberry para la detección de movimiento en el

estacionamiento. Esto se lo realizó mediante la aplicación de filtros, detección de contornos y sustracción de fondo para separar el fondo del objeto de interés, en nuestro caso los objetos en movimiento. Para la identificación de plazas de estacionamiento se extrajo el área de cada una de las plazas de estacionamiento y mediante la aplicación de filtros y clasificadores de imágenes se calificó a las plazas de estacionamiento disponibles u ocupadas. La eficiencia de esta etapa es limitada por la cantidad de luminosidad presente en un instante de tiempo.

- Al emplear el clasificador Haar Cascade para la identificación de automóviles en tiempo real se concluye que el clasificador opera de manera eficiente y precisa si la intensidad de luminosidad no es ni muy alta ni muy baja, debido a estos parámetros característicos una imagen presenta ciertas variaciones generando errores en la identificación de la presencia de vehículos. Los cuales pueden ser disminuidos al realizar la variación de la sensibilidad de reconocimiento con el fin de seleccionar los puntos más relevantes de la imagen.
- Uno de los factores que provocaron mayor porcentaje de error en el sistema es el cambio de luminosidad en el transcurso del día, variando las intensidades de los pixeles de las imágenes de las plazas de estacionamiento. La utilización de filtros que ayudan con la reducción de los valores de pixeles, en nuestro caso el filtro bilateral, es de mucha importancia en estos tipos de trabajos que se instalan en ambientes exteriores de intensidad luminosa muy variable.
- El sistema está desarrollado en su totalidad por medio de una red inalámbrica por lo que se producen retrasos al momento de la carga y descarga de las imágenes del estacionamiento. Es necesaria la transmisión

de las imágenes debido al procesamiento necesario para la identificación de movimiento, así como el estado de las plazas de estacionamiento para su almacenamiento en la base de datos. Debido a estos factores varían los retardos dependiendo de la velocidad de red en la que opera el sistema.

- La aplicación del sistema se desarrolló por medio del software de realidad aumentada Unity, la cual mediante el escaneo de un código QR muestra los resultados de identificación de los lugares de estacionamiento disponibles.
- La aplicación proporciona una imagen de los parqueaderos con un mínimo retraso de actualización entre imagen, lo cual depende de la velocidad de la red de comunicación, al igual que la detección de movimiento dentro del estacionamiento. La aplicación móvil es sencilla, proporciona y presenta solo información necesaria sobre el estacionamiento.

5.2 Recomendaciones

- Se requiere de una base de datos gratuita y eficiente, la cual pueda contener un número considerable de imágenes y sea compatible con Unity. Además, debe ser confiable y accesible para varios usuarios evitando errores al momento de transmitir las imágenes de las plazas de estacionamiento disponibles y la detección de movimiento en el mismo.
- Es necesario el uso de una computadora de placa con una memoria RAM de media-alta capacidad, ya que para el procesamiento la GPU aumenta la velocidad con la que se realiza el procesamiento digital de imágenes para la identificación de las plazas de estacionamiento mediante Haar Cascade que representa un coste computacional ligeramente alto, al igual que para la detección de movimiento por la aplicación de filtros y sustracción de fondo.

- Para lograr un mejor entrenamiento del clasificador Haar Cascade se recomienda cargar un número considerable de imágenes tanto positivas como negativas es decir imágenes de las plazas de estacionamiento con presencia y ausencia de vehículos. Esto nos permitirá, tener un sistema más eficiente al momento de realizar el reconocimiento de plazas de estacionamiento disponibles u ocupadas.
- Para mejorar la eficiencia del sistema se debe utilizar una cámara que proporcione una buena sensibilidad de luminosidad, lo que brindará una mejor calidad de las imágenes al momento de trasmitirlas, reduciendo considerablemente los errores del sistema.
- Es recomendable realizar las pruebas de eficiencia del sistema por varios días debido a que los cambios de luminosidad y los cambios en los factores ambientales afectan el correcto funcionamiento del reconocimiento de las plazas de estacionamiento al igual que la detección de movimiento dentro de él.

Bibliografía

Anaconda Documentation. (2020). Recuperado el 5 de julio 2021, Obtenido de https://docs.anaconda.com/anaconda/navigator/

Baltazar, C. P. (2020). Repositorio Universidad Técnica de Ambato. Recuperado el 10 de julio 2021, Obtenido de https://repositorio.uta.edu.ec/bitstream/123456789/30726/1/Tesis_t1678si.pdf

Beck, H. A. (2009). Educational applications of virtual world environments. American Society of Agricultural and Biological Engineers Annueal International Meeting, 36-44.

Berrones Reyes, M. C. (2019). Clasificación de mamografías mediante redes neuronales convolucionales. Universidad Autónoma de Nuevo León.

Chicaiza, F. E. (octubre de 2017). Repositorio UTA. Recuperado el 10 de julio 2021, Obtenido de https://repositorio.uta.edu.ec/bitstream/123456789/26945/1/Tesis_t1343ec.pdf

Costa Campos, A., & Fernandez Bozal, J. (08 de Noviembre de 2019). La Imagen Digital. Recuperado el 12 de julio de 2021, Obtenido de Revista española de Ortodoncia: http://www.revistadeortodoncia.com/files/2005_35_3_255-266.pdf

Cristian Erazo, S. N. (marzo de 2019). Repositorio UTN. Recuperado el 16 de julio de 2021, Obtenido de http://repositorio.utn.edu.ec/bitstream/123456789/9075/1/04%20RED%20216%20TRABAJO%20DE%20GRADO.pdf

Developer, V. (03 de diciembre de 2018). Vuforia. Recuperado el 18 de julio de 2021, Obtenido de https://library.vuforia.com/articles/Training/Object-Recognition

DYNATEC. (20 de octubre de 2015). Recuperado el 10 de agosto de 2021, Obtenido de https://dynatec.es/2015/10/20/usos-realidad-aumentada-ingenieria/

Gómez, D., & Guerrero, A. (2016). Estudio y análisis de técnicas para procesamiento digital de imágenes. Universidad tecnológica de Pereira, Colombia.

Guamán, L. R. (julio de 2015). Repositorio UPM. Recuperado el 21 de agosto de 2021 Obtenido de http://oa.upm.es/39016/1/TESIS_MASTER_LUIS_RODRIGO_BARBA_GUAMAN.pdf

Holden, S. (13 de Noviembre de 2018). Python. Recuperado el 23 de agosto de 2021, Obtenido de https://wiki.python.org/moin/FrontPage

J, E. (11 de 7 de 2015). Fundamento de procesamiento de Imágenes. Recuperado el 13 de junio de 2021, Obtenido de Universidad Autonoma de Baja California: https://www.academia.edu/16801512/Fundamentos_de_procesamiento_de_im%C3%A1genes_digitales

Juan Chiza, J. V. (mayo de 2016). Diseño e implementacion de una sistema para la gestion del parqueaderos de la Univesidad de las Fuerzas Armadas ESPE utilizando algoritmo surf y software. Recuperado el 08 de enero de 2021, Obtenido de Repositorio Espe: http://repositorio.espe.edu.ec/xmlui/bitstream/handle/21000/12405/T-ESPEL-ENI-0374.pdf?sequence=1&isAllowed=y

Kumar Amit, M. H. (2015). Sistema basado en el procesamiento de imagenes para la clasificacion de los vehiculos para fines de estacionamiento. IEEE Xplore.

Martinez, M. A. (2015). TESIS EN RED. Recuperado el 05 de noviembre de 2020, Obtenido de https://www.tesisenred.net/bitstream/handle/10803/31767/3de3.MAMcap5_conclusiones_bibliograf%C3%ADa.pdf?sequence=4

Mendoza Chipantasi, D. J. (octubre de 2012). Repositorio Fuerzas Armadas Espe. Recuperado el 16 de febrero de 2021, Recuperado el 06 de marzo de 2021, Obtenido de http://repositorio.espe.edu.ec/handle/21000/5878

Minardi, F. (octubre de 2014). Jeuazarru.com. Recuperado el 26 de mayo de 2021, Obtenido de http://jeuazarru.com/wp-content/uploads/2014/10/computer_vision.pdf

OpenCV. (2012). Recuperado el 22 de abril de 2021, Recuperado el 02 de septiembre de 2021, Obtenido de OpenCV: https://digitimagen.blogspot.com/2012/08/modulos-de-la-libreria-opencv.html

OpenCV. (s.f.). Recuperado el 21 de septiembre de 2021, Obtenido de https://docs.opencv.org/master/da/d22/tutorial_py_canny.html

OpenCV. (s.f.). Recuperado el 21 de septiembre de 2021, Obtenido de https://opencv.org/

Ortiz, A. M. (2000). Rediris. Recuperado el 11 de agosto de 2021, Obtenido de http://elies.rediris.es/elies9/4-1-2.htm

Raspberry Pi Foundation. (2016). Recuperado el 15 de abril de 2021, Obtenido de https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

Rebaza, J. V. (2007). Deteccion de Bordes mediante el algoritmo de Canny. ResearchGate, 1-4.

REDATEL. (22 de Marzo de 2016). Recuperado el 18 de agosto de 2021, Obtenido de https://www.redatel.net/html/camara-ip-conceptos-basicos.html#:~:text=Una%20c%C3%A1mara%20IP%20consiste%20principalmente,red%20para%20transmisi%C3%B3n%20de%20datos.

Redecker, C. (2013). The use of ICT for the assessment of key competences. Joint Research Centre, Institute for Prospective Technological Studies. European Commission.

SÁNCHEZ, J. A. (Junio de 2011). SISTEMA DE RECONOCIMIENTO DE OBJETOS REMOVIDOS DE UNA ESCENA. Recuperado el 02 de septiembre de 2021, Obtenido de Repositorio Javeriana: https://repository.javeriana.edu.co/bitstream/handle/10554/7073/tesis535.pdf?sequence=1&isAllowed=y

Sarcos, G. B. (2009). Desarrollo de un software que permita mostrar las caracteristicas propias de un melanoma. Tesis pregrado, Escuela Superior Politecnica del Litoral, Guayaquil.

SPYDER. (2020). Recuperado el 02 de agosto 2021, Obtenido de https://www.spyder-ide.org/

Tamayo, M. A. (enero de 2016). Repositorio UPS. Recuperado el 04 de abril de 2021, Obtenido de https://dspace.ups.edu.ec/bitstream/123456789/12425/1/UPS%20-%20ST002102.pdf

Unity. (2020). Obtenido de Recuperado el 05 de septiembre de 2021, https://unity.com/es

Urooj. (11 de septiembre de 2019). Edureka. Recuperado el 10 de mayo de 2021, Obtenido de https://medium.com/edureka/spyder-ide-2a91caac4e46#:~:text=Spyder%20is%20an%20open%2Dsource,features%20which%20are%20discussed%20below.

Villacis, M. A. (Agosto de 2008). Repositorio Fuerzas Armadas ESPE. Recuperado el 02 de abril de 2021, Obtenido de http://repositorio.espe.edu.ec/xmlui/bitstream/handle/21000/3299/T-ESPEL-0550.pdf?sequence=1&isAllowed=y

Vuforia, Q. (06 de marzo de 2019). Image Targets. Recuperado el 10 de septiembre de 2021, Obtenido de

https://web.archive.org/web/20190306123126/https://library.vuforia.com/articles/Training/Image-Target-Guide

Ycezalaya, J. P. (s.f.). RNDS.com.ar. Recuperado el 04 de abril de 2021, Obtenido de http://www.rnds.com.ar/articulos/031/rnds_084w.pdf

Anexos