
DISEÑO DE SISTEMA EXPERTO PARA
BÚSQUEDA DE PLAZAS LIBRES EN
UN APARCAMIENTO MEDIANTE
APRENDIZAJE AUTOMÁTICO
Y VISIÓN ARTIFICIAL



Trabajo de Fin de Máster
Curso 2018-2019

Juan Manuel Carrera García

Directores

María Guijarro Mata-García

Joaquín Recas Piorno

Máster en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

Diseño de Sistema Experto para Búsqueda de Plazas Libres en un Aparcamiento mediante Aprendizaje automático y Visión Artificial

Trabajo de Fin de Máster

**Departamento de Arquitectura de Computadores y
Automática**

Autor

Juan Manuel Carrera García

Directores

María Guijarro Mata-García

Joaquín Recas Piorno

Convocatoria: *Junio 2019*

Calificación: *9*

Máster en Ingeniería Informática

Facultad de Informática

Universidad Complutense de Madrid

11 de julio de 2019

Autorización de difusión

El abajo firmante, matriculado en el Máster en Ingeniería Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente trabajo de fin de máster: “Diseño de Sistema Experto para Búsqueda de Plazas Libres en un Aparcamiento mediante Aprendizaje automático y Visión Artificial”, realizado durante el curso académico 2018-2019 bajo la dirección de María Guijarro Mata-García y Joaquín Recas Piorno en el Departamento de Arquitectura de Computadores y Automática, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en internet y garantizar su preservación y acceso a largo plazo.

Juan Manuel Carrera García

*El tiempo no es más que una
ilusión persistente*

Albert Einstein

Agradecimientos

A María, por nunca decir *no* y bajar a la tierra ideas disparatadas.

A Joaquín, por subir al barco y terminar de ajustar el timón.

A mi familia, en especial padres y hermano, por estar siempre a mi lado.

A la UCM, por cederme la infraestructura necesaria para este trabajo.

Resumen

Encontrar sitios libres en aparcamientos dedicados es un problema cotidiano al que se enfrentan los conductores. Debido a la falta de conocimiento de la situación exacta de las posiciones libres dentro de los aparcamientos, pierden una cantidad de tiempo innecesario en la búsqueda de huecos en los que poder aparcar. Un sistema capaz de estimar las características del vehículo a estacionar y de ubicar los huecos libres válidos para el tamaño de ese vehículo, disminuiría los tiempos de búsqueda empleados por los conductores y conseguiría un reparto óptimo de las plazas del aparcamiento.

El sistema propuesto combina técnicas de *inteligencia artificial*, para determinar la ocupación y distribución de los vehículos dentro del aparcamiento, y de *entornos distribuidos*, para optimizar los cálculos y escalabilidad del sistema. Mediante técnicas de *visión artificial* y *redes neuronales* el sistema implementa los módulos necesarios para la detección de vehículos dentro de un aparcamiento. Primero, el sistema genera un mapa computacional con las posiciones y tamaños de las plazas de aparcamientos. Segundo, identifica las características de los vehículos entrantes empleando técnicas de visión por computador para establecer las medidas. Tercero, identifica y diferencia los vehículos de otros objetos dentro del aparcamiento para conocer su ubicación exacta y tamaño empleando redes neuronales. Finalmente, el sistema identifica a los vehículos entrantes asignándoles la ubicación óptima en función de sus características, gracias al conocimiento de la ocupación del aparcamiento en tiempo real.

Se evalúa la precisión del sistema obteniendo resultados en la de detección de la ocupación del 93,39%, lo que indica el gran potencial del sistema como mejora a los enfoques actuales basados en la detección de aparcamiento en recintos exteriores mediante inteligencia artificial, proponiendo nuevos enfoques y características no estudiadas hasta el momento.

Palabras clave

Detección de ocupación de aparcamiento, visión por computador, aprendizaje automático, redes neuronales, entornos distribuidos, OpenCV, Tensorflow.

Abstract

Finding free places in dedicated parking lot is a daily problem faced by drivers. Due to the lack of knowledge of the exact location of free positions within the car parks, they waste an unnecessary amount of time searching for parking space. A system capable of estimating the characteristics of the vehicle to be parked and of locating the free spaces valid for the size of that vehicle would reduce the search times used by drivers and achieve an optimum distribution of parking spaces.

The proposed system combines artificial intelligence techniques, to determine the occupation and distribution of vehicles within the car park, and distributed environments, to optimize the computations and scalability of the system. By means of artificial vision techniques and neural networks, the system implements the necessary modules for the detection of vehicles inside a car park. First, the system generates a computer map with the positions and sizes of the parking spaces. Second, it identifies the characteristics of incoming vehicles using computer vision techniques to establish the measurements. Third, it identifies and differentiates vehicles from other objects within the car park in order to know their exact location and size using neural networks. Finally, the system identifies incoming vehicles by assigning them the optimum location according to their characteristics, thanks to the knowledge of the occupation of the car park in real time.

The precision of the system is evaluated, obtaining results in the detection of occupancy of 93.39%, which indicates the great potential of the system as an improvement to the current approaches based on the detection of parking in outdoor areas by artificial intelligence, proposing new approaches and features not studied so far.

Keywords

Detection of parking occupancy, computer vision, automatic learning, neural networks, distributed environments, OpenCV, Tensorflow.

Índice

Autorización de difusión	v
Agradecimientos.....	ix
Resumen	xi
Palabras clave	xii
Abstract	xiii
Keywords	xiv
Índice	xv
Índice de tablas.....	xix
Índice de abreviaturas.....	xxi
Capítulo 1	1
Introducción.....	1
1.1. Identificación del problema	1
1.2. Motivación.....	3
1.3. Objetivos del trabajo.....	5
1.4. Estructura del documento	6
Capítulo 2.....	9
Estado del arte.....	9
2.1. Sistemas no basados en visión artificial	9
2.1.1. Sensores.....	9
2.1.2. Aparcamientos robotizados	11
2.2. Sistemas basados en visión artificial	13
2.2.1. Sistemas basados en redes neuronales convolucionales	19

2.2.2. Clasificación del estado del arte	21
2.3. Comparación entre los estudios analizados y el sistema propuesto.....	24
Capítulo 3	27
Sistema propuesto	27
3.1. Descripción general.....	27
3.2. Modelado del entorno de pruebas	29
3.3. Visión artificial y aprendizaje automático	30
3.3.1. Detección de plazas de aparcamiento	31
3.3.2. Detección de vehículos en el aparcamiento.....	48
3.3.3. Medición sobre la escena	53
3.3.4. Detección de movimiento.....	54
3.4. Diseño	57
3.4.1. Clases	57
3.4.2. Software.....	61
3.5. Arquitectura	63
3.5.1. Descripción general.....	63
3.5.2. Nodos de computación	64
3.5.3. Comunicaciones	67
Capítulo 4	69
Resultados	69
4.1. Precisión del sistema.....	69
4.2. Tiempos de ejecución.....	71
Capítulo 5	73
Conclusiones.....	73
5.1. Análisis de resultados	73
5.2. Cumplimiento de objetivos	74
5.3. Trabajo futuro	74
Chapter 1.....	77
Introduction	77
1.1. Problem identification.....	77

1.2. Motivation	78
1.3. Work objectives	80
1.4. Structure of the document	81
Chapter 5	83
Conclusions	83
5.1. Analysis of results	83
5.2. Objectives achieved.....	83
5.3. Future Work	84
Bibliografía.....	85
Anexo 1.....	93
Resultados.....	93

Índice de tablas

Tabla 2.1. Clasificación de los estudios analizados.....	22
Tabla 4.1. Resultados en la precisión en el cálculo del área real.....	70
Tabla 4.2. Resultados de la precisión en la detección de la ocupación del aparcamiento en binario —ocupado y vacío—.....	71
Tabla 4.3. Resultados de la precisión en la detección de la ocupación del aparcamiento.....	71
Tabla 4.4. Resultados de la precisión en la detección de la ocupación del aparcamiento.....	72
Tabla A.1.1. Resultados en la precisión en la determinación del área de las plazas..	93

Índice de abreviaturas

2D	2 Dimensiones
3D	3 Dimensiones
AA	Aprendizaje Automático
API	Interfaz de Programación de Aplicaciones
BHDF	Marco de Detección Jerárquico Bayesiano (Bayesian Hierarchical Detection Framework)
FPN	Red Piramidal de Características (Feature Pyramid Network)
HGO	Histogramas de Gradientes Orientados
JSON	Notación de Objeto de JavaScript (JavaScript Object Notation)
K-NN	K-Vecinos más Cercanos (K-Nearest Neighbors)
IA	Inteligencia Artificial
R-RNC	Red Neuronal Convolutiva basada en Regiones
RFID	Identificación por Radio Frecuencia (Radio Frequency Identification)
RGB	Rojo, Verde y Azul (Red, Green and Blue)
RN	Red neuronal
RNC	Red Neuronal Convolutiva
RoI	Región de Interés (Region of Interest)
RRP	Red de Regiones Propuestas
SURF	Speed-Up Robust Features
SVM	Máquina de Vectores de Soporte (Support Vector Machines)
VA	Visión Artificial

Capítulo 1

Introducción

1.1. Identificación del problema

¿Cuánto tiempo perdemos buscando aparcamiento? ¿Quién no se ha sentido estresado por tener que dar vueltas con el coche buscando sitio mientras ve que llega tarde a una cita importante, una reunión o un examen? Esto son realidades y problemas a los que nos enfrentamos en las ciudades pobladas con un alto nivel de tráfico. Es más, si preguntamos a los conductores cuál es la razón que más frustración les produce al volante tendremos una respuesta clara, *la búsqueda de aparcamiento*.

Según Ludvik Eide (Eide, 2018), en un estudio realizado en Norteamérica para evaluar la frustración causada por la búsqueda de aparcamiento se concluye que: “El 61% de los conductores refirieron haberse sentido estresados mientras buscaban un sitio donde dejar el coche, el 42% no había podido acudir a una cita por no encontrar estacionamiento, el 34% había abandonado un viaje por problemas de aparcamiento y el 23% había experimentado ira al volante”.

Pero no es solo cuestión de salud. El problema del aparcamiento repercute negativamente en temas económicos y medioambientales. Según un estudio de INRIX Research los americanos gastan un total de \$73.000.000.000 en la búsqueda de aparcamiento. Concluye que invierten una media de 17 horas al año por conductor en lo que ellos llaman “*the parking pain*”, alcanzando su máximo en Nueva York, donde pierden 107 horas por conductor y año (Searching for Parking Costs Americans \$73 Billion a Year, 2017).

Otro estudio de la compañía revela que en 2017 el coste de la propiedad de un vehículo ascendió a 10.288 dólares de media por conductor en Estados Unidos. Entre los gastos de propiedad establece la existencia de dos tipos de coste. Los costes directos que incluyen mantenimientos, seguros, combustible y tarifas de parking y peaje. Los costes indirectos u ocultos que incluyen el aparcamiento y las multas de estacionamiento. Los costes indirectos, entre los que se incluye el tiempo perdido en buscar aparcamiento, representan más de un 30% del coste total. El Dr. Graham Cookson, Economista Jefe de INRIX, afirma que: "El verdadero coste de conducir era asombroso, pero lo que realmente sorprendió fue el tamaño y el desglose de los costos ocultos. El estacionamiento, por ejemplo, representó un tercio del costo total de la propiedad de un vehículo" (New INRIX Study Finds Parking is the Largest Cost of Driving, 2017). Como se observa en la [figura 1.1](#), el coste total de la búsqueda de aparcamiento supuso un 14% sobre el coste de la propiedad anual del vehículo.

Country	Direct Costs		Indirect Costs		Total Cost of Driving
	Car Ownership	Parking	Congestion	Parking Pain	
U.S.	\$5,645	\$1,607	\$1,642	\$1,394	\$10,288
U.K.	\$4,260	\$1,244	\$1,398	\$1,337	\$8,239
Germany	\$4,376	\$1,062	\$2,347	\$1,556	\$9,341

Figura 1.1. Coste de propiedad de vehículo en Estados Unidos en el año 2017 (New INRIX Study Finds Parking is the Largest Cost of Driving, 2017)

En el plano europeo, INRIX Research analiza la situación de Reino Unido. Como se puede observar en la [figura 1.2](#) los conductores británicos pierden 44 horas al año buscando aparcamiento, lo que supone un coste de 733 libras esterlinas por conductor, haciendo un cómputo total en el país de 23.300 millones de libras esterlinas. Luego, la búsqueda de aparcamiento, en el caso de UK, supone más de 16% del coste de la propiedad del vehículo —[figura 1.2](#)—.

Según un estudio realizado por Xerox en las 19 ciudades más grandes de europa (El 9% de los conductores pasan más de 15 minutos al día buscando aparcamiento, según un estudio de Xerox, 2016), se concluye que el 9% de los conductores están de media 96 horas anuales buscando aparcamiento, lo que supone unos 15 minutos diarios. Esta cifra aumenta en algunas capitales europeas incluyendo a un 19% de los conductores y suponiendo un tiempo de búsqueda que puede llegar a los 30 minutos.

Con los datos evaluados, podemos concluir que existe un claro problema con el estacionamiento en las grandes poblaciones que, lejos de mejorar, tiende a empeorar.

Rank	UK City	Average 2-Hour Parking Cost (within one mile of city centre)	On-Street Search Time (minutes per trip)	Off-Street Search Time (minutes per trip)	Annual Search Time (hours per driver per annum)	Annual Search Cost Per Driver	Annual Search Cost Per City
1	London	£13	12	10	67	£1,104	£4.3bn
2	Belfast	£3	10	8	56	£928	£134m
3	Leeds	£5	8	7	47	£772	£297m
4	Bristol	£4	9	8	46	£768	£169m
5	Birmingham	£3	8	7	46	£757	£373m
6	Cardiff	£6	9	7	44	£737	£126m
7	Manchester	£4	8	7	41	£688	£169m
8	Glasgow	£4	9	7	40	£660	£226m
9	Edinburgh	£5	8	7	38	£625	£167m
10	Southampton	£2	8	7	35	£588	£98m
	UK	£3	8	7	44	£733	£23.3bn

Figura 1.2. Horas perdidas buscando aparcamiento en Reino Unido (Searching for Parking Costs the UK £23.3 Billion a Year, 2017)

1.2. Motivación

Es un hecho que la mayoría hemos llegado tarde alguna vez a una reunión, a una cita o con la película empezada al cine por perder demasiado tiempo buscando aparcamiento. Es más, en lugares como centros comerciales en los que existen sistemas que indican la ocupación de las plazas, las vueltas buscando el preciado hueco se dan de manera constante. Inclusive, obviando el factor extra del copiloto que dice —“vuelve, vuelve, que te has saltado uno”— para dar la vuelta y encontrar un Smart mal aparcado, perder el tiempo buscando aparcamiento es un problema real que no es del agrado de nadie.

De hecho, sin salir de nuestra ciudad, en Madrid es muy común calcular el tiempo de viaje y añadir, al menos, 15 minutos pensando en la búsqueda de aparcamiento. En trayectos cortos en la capital, con buenas condiciones de tráfico, no se exceden los 30 minutos, lo que supone que gastamos un tercio del tiempo de nuestro viaje en la búsqueda de aparcamiento.

Si a esto le sumamos que hoy en día la falta de tiempo es una de las principales razones del estrés, y los costes ocultos que suponen la búsqueda de aparcamiento son considerables; vemos que estamos ante un problema grave que todos damos por hecho y asumimos.

Y, aunque es cierto que cada vez se apuesta más por el transporte público y se hace más hincapié en alternativas de movilidad sostenibles, como Madrid Central, en las

que tenemos que poner el foco para el futuro; el uso masificado del vehículo personal, es una realidad hoy en día que presenta difícil solución a corto y medio plazo.

Ante este problema, hay aparcamientos que han buscado dar una solución con sensores que indican con una luz verde si la plaza está libre y roja en caso contrario, e incorporan paneles informativos en las avenidas del aparcamiento indicando el número de plazas libres en cada zona. En el capítulo 2 se explican en detalle las soluciones existentes en la literatura. A modo de introducción, la utilización de sensores es una solución poco efectiva debido a que: (i) no proporciona un conocimiento al conductor de la posición exacta de las plazas libres haciendo que el conductor necesite “dar vueltas” para encontrar un sitio; y (2) depende de la colocación del vehículo dentro de la plaza de aparcamiento para que el sensor detecte correctamente la ocupación. Si a esto añadimos que la implementación de estos sistemas supone un coste elevado y un despliegue complejo debido a la necesidad de utilizar un sensor por cada espacio de estacionamiento, lo que implica una necesidad elevada de dispositivos, cableado y sistemas de control y monitorización; encontramos los primeros problemas de este tipo de sistema.

Si ya observamos que esta solución presenta problemas en espacios cerrados donde se dispone de puntos de anclaje, tomas de corriente y canalizaciones donde se precisen, es lógico pensar que su implementación tiene aún más inconvenientes en espacios abiertos, hasta el punto de ser inviables.

Además, con el avance tecnológico que vivimos a diario resulta difícil entender que las opciones más punteras en el área tengan más de diez años (Lee et al, 2008). Si a esto le sumamos el auge que estamos experimentando en inteligencia artificial (IA) (Russell and Norvig, 2016), viéndose reflejado en declaraciones como la de Hanoi Morillo, experta en IA en Google, que afirma que: "El mundo será AI First" (Balbi, 2018). Es lógico pensar en un sistema capaz de mejorar la gestión del aparcamiento haciendo uso de las ventajas que puede aportar la inteligencia artificial.

Por estos motivos, la idea de este trabajo es utilizar la inteligencia artificial para diseñar e implementar un sistema inteligente, distribuido y adaptativo, capaz de:

- clasificar los objetos dentro de un aparcamiento exterior, diferenciando cuáles son vehículos;
- determinar la ocupación y distribución de estos vehículos dentro del aparcamiento en tiempo real;
- determinar el tamaño del vehículo que accede al aparcamiento;

- determinar la ubicación óptima para el vehículo entrante en función de sus dimensiones y la ocupación del aparcamiento.

1.3. Objetivos del trabajo

El objetivo principal de este trabajo es la creación de un *sistema experto para la búsqueda de plazas libres* en un aparcamiento utilizando *visión artificial* (VA) y *aprendizaje automático* (AA). Con el fin de diseñar un sistema robusto, eficiente, distribuido y adaptativo, se plantea la siguiente división en objetivos generales a alcanzar en el sistema final.

- **Investigación.** Se realiza una primera fase de investigación para evaluar las diferentes alternativas existentes en el mercado, e identificar las tecnologías, algoritmos y herramientas necesarias para la implementación del sistema.
- **Diseño e implementación.** El diseño y la implementación son las partes centrales de este trabajo y se pueden dividir en los siguientes objetivos específicos a cumplir por el sistema:
 - **Identificación de plazas de aparcamiento.** El sistema debe ser capaz de detectar las líneas del aparcamiento con el fin de obtener un mapa computacional de las plazas. Para ello, se aplican técnicas de *visión por computador* para analizar y buscar patrones y líneas en las imágenes.
 - **Detección de objetos.** Puesto que dentro de un parking coexisten vehículos con otros elementos (personas, árboles, papeleras, etc.), el sistema debe ser capaz de identificar cuáles son vehículos con el fin de determinar la posición y tamaño de los mismos dentro de las plazas de aparcamiento. Para ello, se hace uso de *redes neuronales* (RN) para dotar de aprendizaje y conocimiento al sistema y, junto con la VA, sea capaz de hacer una clasificación de las entidades.
 - **Mediciones del aparcamiento y de los vehículos.** Esta es una de las grandes mejoras con respecto a los sistemas actuales. Para evitar depender de la colocación exacta de los vehículos dentro una plaza, problema que aparece en los sistemas con sensores cuando un vehículo aparca entre dos plazas ocupando ambas y generando falsos positivos, se plantea que el sistema sea capaz de establecer el tamaño libre y ocupado dentro de cada plaza. Para ello, debe ser capaz de determinar las

medidas de cada plaza y el tamaño y posición de los coches estacionados en ellas.

- **Identificación y medición del vehículo entrante.** Puesto que el sistema conoce en todo momento el espacio restante en cada una de las plazas del aparcamiento, se plantea la medición del vehículo que accede al recinto con el fin de determinar su ubicación óptima.
- **Algoritmo de búsqueda de plaza.** Se diseña un algoritmo que determine la posición idónea para el vehículo teniendo en cuenta el tamaño y diversos factores especificados en la descripción del sistema.
- **Arquitectura.** Puesto que una de las premisas es obtener un sistema distribuido y escalable, la arquitectura hardware del sistema tiene varios nodos de computación. En cada uno de estos nodos se implementan los diferentes objetivos específicos enunciados anteriormente.
- **Prueba de concepto.** El objetivo final es obtener un sistema experto con el que poder realizar una prueba de concepto con el fin de evaluar los resultados obtenidos. Para ello se diseña el despliegue del sistema y, mediante una batería de imágenes, se evalúa el rendimiento y la exactitud que ofrece.
- **Análisis de resultados y conclusiones.** Se realiza un análisis final de los resultados obtenidos estableciendo las conclusiones a las que se ha llegado en el trabajo junto con los posibles trabajos futuros.

1.4. Estructura del documento

La estructura de este documento se compone de 5 capítulos en los que se describen los distintos aspectos relativos a este trabajo.

El capítulo 1 de Introducción, en el que se encuentra el apartado presente, es una introducción en la que se describe: la problemática encontrada (1.1), la motivación que implica la resolución del problema (1.2), los objetivos a alcanzar en el trabajo para dar una solución viable (1.3) y la estructura del documento (1.4).

El capítulo 2 de Estado del arte contiene una revisión de los estudios previos al desarrollo del sistema haciendo una división en dos secciones: *sistemas no basados en visión artificial* (2.1), que incluye los sistemas basados en sensores y los aparcamientos robotizados; y *sistemas basados en visión artificial* (2.2), prestando especial atención a los sistemas basados en redes neuronales. Se añade un apartado

final (2.3) comparando las características de los enfoques previos y el sistema descrito en este trabajo.

El capítulo 3 de Sistema propuesto contiene la parte central de esta memoria describiendo: el sistema de forma detallada (3.1), el modelado del entorno de pruebas utilizado para recrear diferentes casuísticas (3.2), los módulos que incluyen funcionalidades con visión artificial y aprendizaje automático, (3.3) y el diseño (3.4) y arquitectura (3.5) planteados para el sistema.

El capítulo 4 de Resultados muestra las medidas utilizadas para validar el rendimiento y la efectividad del sistema y aporta resultados cuantitativos sobre el trabajo realizado.

El capítulo 5 de Conclusiones contiene el análisis de los resultados obtenidos en el capítulo anterior y las conclusiones y trabajos futuros que se plantean a raíz de este trabajo.

Capítulo 2

Estado del arte

En este capítulo se analizan los sistemas de gestión de plazas de aparcamiento que utilizan diferentes tecnologías y aplicaciones para identificar la ocupación de las plazas y mejorar la eficiencia del aparcamiento. Se establece una división en dos bloques en función de si utilizan visión artificial como método de detección o no.

En el apartado 2.1, para conocer las alternativas al sistema de este trabajo, se analizan los sistemas principales que **no** hacen uso de la VA. En el apartado 2.2, se revisan los trabajos que hacen uso de la VA para conocer las técnicas utilizadas en sistemas similares al propuesto. Este apartado se inicia con un estudio de los trabajos basados en VA, analizando de forma cronológica los más relevantes, describiendo los enfoques y características que presentan cada uno de ellos. En el apartado 2.2.2 se agrupan y categorizan dichos trabajos exponiendo las ventajas e inconvenientes que presentan.

Finalmente, en el apartado 2.3, se finaliza el estado del arte con una comparativa de los estudios analizados y el sistema propuesto, indicando las mejoras que se incluyen con respecto a los enfoques previos.

2.1. Sistemas no basados en visión artificial

2.1.1. Sensores

Actualmente, los sistemas de aparcamiento basados en *sensores* son la opción más utilizada. Estos sistemas utilizan redes de sensores distribuidas por las plazas de aparcamiento que determinan la ocupación de una plaza.

(Paidi et al., 2018) establece, en una revisión de la literatura actual, que los sensores son una de las herramientas más comunes y que han sido ampliamente probados en la literatura. Existen varios tipos de sensores destinados a la detección de la ocupación de las plazas de aparcamiento. En base a este estudio se analizan las distintas alternativas y las ventajas e inconvenientes que presentan.

- *Sensores de infrarrojos.* Los sensores infrarrojos pasivos detectan la radiación que emiten los cuerpos para determinar si un vehículo está ocupando una plaza. Los sensores activos están compuestos por un emisor y un receptor, encargados de emitir y recibir un haz de luz, y detectan los vehículos en función de la cantidad de energía reflejada.
- *Sensores ultrasónicos.* Su comportamiento es similar a los sensores infrarrojos activos, pero haciendo uso de energía sonora. Este sensor utiliza ondas sonoras para detectar la existencia de objetos en función de la energía reflejada.
- *Detectores de bucle de inducción.* Estos detectores utilizan cables subterráneos y principios de electromagnetismo para detectar cambios en el campo magnético y establecer la existencia de un vehículo. Debido al coste de instalación suelen colocarse en la entrada y la salida a modo de conteo de vehículos.
- *Sistemas de guiado para aparcamientos.* Estos sistemas están formados por pantallas colocadas a lo largo del aparcamiento que muestran un conteo de las plazas disponibles. Necesitan otras herramientas para conocer la situación actual del aparcamiento y mostrarla en las pantallas.
- *Identificación por radiofrecuencia (RFID).* Utilizan etiquetas RFID (transpondedores) que almacenan la identificación de los vehículos. En las entradas y salidas del parking se ubican lectores RFID compuestos por un transmisor, un receptor, un decodificador y una antena. Mediante los lectores se detecta el paso de los vehículos conociendo la ocupación total.
- *Magnetómetro.* Estos sensores utilizan principios de electromagnetismo para detectar cambios en el campo electromagnético cuando un vehículo se sitúa encima de ellos.
- *Radar de microondas.* Estos radares se basan en la utilización de microondas y la detección de la señal reflejada para detectar objetos en movimiento.

Las conclusiones que arroja la investigación sobre estos sistemas son:

- Los sistemas basados en sensores infrarrojos, ultrasónicos y magnetómetros, aunque son económicos individualmente, el despliegue y mantenimiento es caro debido a que precisamos de un sensor para cada plaza de aparcamiento. Los infrarrojos y ultrasónicos son sensibles a condiciones atmosféricas, como la luz y la lluvia, lo que hace que no sean recomendables para aparcamientos exteriores. Entre las alternativas vistas son la mejor opción para el conocimiento de la ocupación de cada plaza de manera individual.
- Los sistemas de guiado de aparcamientos por sí solos no pueden determinar la ocupación del parking. No aportan un conocimiento específico de las plazas libres. Muestran la ocupación del parking de forma general o, a lo sumo, de un conjunto de plazas. Son un buen complemento a los anteriores.
- Los sistemas basados en RFID y detectores de bucle de inducción son válidos para interior y exterior. Su instalación es simple debido a que se ubican únicamente en las entradas y salidas, aunque son sistemas caros debido al coste de instalación y del sistema. Proporcionan un conteo de los vehículos en el parking, pero no permiten conocer distribución individual de la ocupación de cada plaza. La utilización de RFID proporciona una ventaja adicional si el aparcamiento precisa identificar los vehículos antes de conceder el acceso.

Hay multitud de trabajos que proponen sistemas basados en sensores. Como caso de uso que, además de los sensores, construye un sistema completo considerando la gestión coordinada de varios aparcamientos distribuidos por la ciudad lo encontramos en el trabajo (Shin & Jun, 2014). Sus autores desarrollan un algoritmo con el que ayudar a los conductores a encontrar el aparcamiento más adecuado teniendo en cuenta el estado en tiempo real de varios de ellos. El algoritmo considera factores como la distancia hasta el aparcamiento, la distancia a pie del aparcamiento al destino final del conductor, el coste de aparcamiento y el tráfico actual. En los aparcamientos se utilizan sensores para la detección de la ocupación de las plazas.

2.1.2. Aparcamientos robotizados

Los *aparcamientos robotizados* son sistemas autónomos controlados por robots y que precisan de plataformas que se encargan de la gestión y distribución de los vehículos en los espacios de aparcamiento. El conductor tiene que estacionar el vehículo en el punto de entrada y el sistema se encarga de colocarlo en la ubicación óptima.

A continuación, se estudian dos tipos de aparcamientos robotizados explicando su funcionamiento y analizando sus ventajas e inconvenientes.

- En (Callao Smart Parking, n.d.) tenemos un parking de grandes dimensiones construido con infraestructuras y grúas internas controladas por un sistema central. Este tipo de aparcamientos pueden parecer una tecnología novedosa, futurista inclusive, pero han cumplido más de una década. En (Penalva, 2006) se analiza el primer aparcamiento robotizado de este tipo construido en Madrid.
- En (“Stanley Robotics”, n.d.) encontramos un robot aparcacoches con una plataforma extensible que puede levantar y mover vehículos de hasta 5,5 metros (“Quién es Stan”, 2019). El sistema requiere de una zona de entrada para que los conductores dejen sus vehículos y un espacio de aparcamiento no accesible a los usuarios en el que el robot los colocará.

Ambos sistemas proporcionan ventajas similares. La distribución del espacio de aparcamiento es óptima puesto que el sistema es el encargado de la colocación de los vehículos conociendo el estado actual del aparcamiento antes de aparcarlo. Se reduce el tiempo de aparcamiento pasando a ser igual al tiempo empleado en colocar el coche en el punto de entrada. Para un mismo espacio entran una mayor cantidad de coches que en un aparcamiento tradicional. En el primer sistema, caben más vehículos por metro cuadrado de espacio ya que se construyen en vertical. En el segundo, aumenta la capacidad del aparcamiento puesto que se eliminan los carriles de circulación de doble sentido para los vehículos ganando plazas de aparcamiento.

El coste de implantación y la infraestructura necesaria es el principal problema de ambos. El primer sistema necesita una infraestructura y maquinaria compleja para su funcionamiento y, si se implanta sobre un aparcamiento convencional, el aprovechamiento de la infraestructura existente es ínfimo. El segundo presenta una ligera ventaja puesto que se puede implementar sobre un aparcamiento convencional, pero necesita la construcción de entradas automatizadas y el cerramiento del parking para evitar la entrada al público. Al igual que el tiempo de aparcamiento es una ventaja, el tiempo de recogida del vehículo es un impedimento debido a que los sistemas no son rápidos al necesitar mover grandes grúas, en el primer caso; y la recolocación de los vehículos en el segundo.

Además, el primer sistema presenta el inconveniente añadido de no poder operarse de forma manual en caso de avería. Esto genera un gran problema si se da la situación de que un usuario no puede sacar su vehículo cuando lo necesita. En (González,

2018) vemos que los fallos en estos sistemas han llevado al cierre de aparcamientos de este tipo.

2.2. Sistemas basados en visión artificial

La *visión artificial* como método de evaluación del estado de ocupación de las plazas de aparcamiento ha sido un tema de investigación recurrente en los últimos años (Sevillano & Màrmol, 2016).

“La visión artificial o visión por ordenador es una disciplina científica que incluye métodos para adquirir, procesar, analizar y comprender las imágenes del mundo real con el fin de producir información numérica o simbólica para que puedan ser tratados por un ordenador. Tal y como los humanos usamos nuestros ojos y cerebros para comprender el mundo que nos rodea, la visión artificial trata de producir el mismo efecto para que los ordenadores puedan percibir y comprender una imagen o secuencia de imágenes y actuar según convenga en una determinada situación.” (Visión artificial, 2018)

En (Wang & Hanson, 1998) se encuentra una de las primeras aproximaciones a la detección de plazas de aparcamiento mediante la utilización de visión. Los autores proponen un sistema basado en la extracción de texturas de superficie y microestructuras (*Surface Texture and Microstructure Extraction*) para detectar los cambios de elevación en una imagen. Estos cambios se llaman protuberancias y se asocian con la aparición de vehículos en la imagen.

Más adelante, se publican los primeros trabajos que utilizan en conjunto visión artificial y aprendizaje automático para la búsqueda de plazas de aparcamiento libres. En (Dan, 2002) se define una patente en la que se utilizan máquinas de vectores de soporte (*Support Vector Machines, SVM*) para la clasificación de los colores dentro de una plaza con el fin de distinguir los coches. En (Funck et al., 2004) se comparan las imágenes de las plazas de aparcamiento con una imagen de referencia de la plaza vacía para detectar los cambios en los píxeles.

En el sistema que se define en (Bong et al., 2006), y mejora en (Bong et al., 2008), los autores utilizan imágenes editadas manualmente para conocer la ubicación de las plazas de aparcamiento. La detección de ocupación de las plazas se realiza aplicando una operación AND al resultado de dos métodos. En el primero, se compara la imagen actual con la de referencia. En el segundo, se realiza una detección de bordes

con el fin de separar las sombras que genere un vehículo sobre las plazas adyacentes. El resultado final es la unión de ambas.

En el trabajo (Lin et al., 2006), los autores plantean un sistema de detección basado en los cambios de color que se producen al existir un vehículo en una plaza. Para la detección del color adecuado de cada plaza de aparcamiento se utiliza un método estadístico basado en varias imágenes. Con este método se resuelve el problema de los cambios de luz. Además, añadieron un análisis de luminancia de la escena con el fin de eliminar sombras que pudiesen dar falsos positivos.

Primeramente en (Wu & Zhang, 2006), y a posteriori en (Wu et al. 2007), se propone un sistema que utiliza aprendizaje automático para la detección de las plazas. El sistema se basa en cuatro características: un preprocesado para la detección de las plazas de aparcamiento, una extracción de características del suelo, un clasificador *SVM* para el reconocimiento de vehículos, y un modelo de campo aleatorio de Markov para comprobar bloques de tres plazas contiguas con el fin de evitar errores por oclusiones o sombras.

Una metodología para calcular homografías 2D aplicando algunos algoritmos de álgebra computacional se presenta en (Sastre et al., 2007). Esta metodología se prueba en un caso de uso de detección de plazas de aparcamiento. El sistema utiliza el cálculo de homografías para generar una pseudo-vista en la que aparezca la ocupación de las plazas de aparcamiento. Se utiliza una extracción de características de texturas mediante filtros de Gabor. Finalmente, se emplea un clasificador *SVM* entrenado con los vectores de textura de cada plaza con diferentes condiciones lumínicas para la detección de la ocupación de cada una.

(True, 2007) plantea una clasificación del histograma de color de las plazas de aparcamiento y un algoritmo de detección de puntos de características del coche para establecer el estado de ocupación. Para la clasificación del histograma se proponen dos métodos de clasificación — k vecinos más cercanos (*k-nearest neighbors*, *k-nn*) y *SVM*—. Se realizan pruebas con ambos, obteniendo mejores resultados el clasificador *SVM*. Para la detección de características de los vehículos se utiliza el algoritmo de detección de esquinas de Harris en la imagen resultante de cada plaza. Las plazas se clasifican y etiquetan de manera manual en la configuración inicial del sistema.

El método propuesto en (Fabian, 2008) aborda el problema de las oclusiones mediante la utilización de varias cámaras distribuidas en el aparcamiento. El algoritmo se basa en tres etapas de procesamiento: (i) se hace un preprocesado de la

imagen para eliminar las sombras y corregir la distorsión, (ii) se define la cámara óptima con la que evaluar el estado de la plaza para eliminar oclusiones, (iii) se evalúa la ocupación de la plaza de aparcamiento.

Huang et al. son los autores que más publicaciones tienen en el área de detección de la ocupación de espacios de aparcamiento. Los trabajos que se detallan a continuación tienen como base la utilización de un marco de detección jerárquico Bayesiano de tres capas (*Bayesian hierarchical detection framework, BHDF*). El marco aparece por primera vez en (Huang et al., 2008) y es mejorado en (Huang & Wang, 2010). Los autores proponen modelar el aparcamiento como una estructura con varias superficies analizadas dentro del *BHDF*. El marco de tres capas resultante contiene: una primera capa de observación de la escena; una segunda capa de etiquetado, mediante la que se define si se trata de un coche, suelo o sombra; y una capa final de modelado de la escena en 3D. Los problemas de luminancia se resuelven con una clasificación de color entre la capa de observación y la de etiquetado. Los problemas de oclusiones, sombras y distorsiones causadas por la perspectiva se resuelven entre la capa de etiquetado y modelado de la escena en 3D. El marco es mejorado de nuevo en (Huang et al., 2012) añadiendo una análisis por cada plaza de aparcamiento, modelada como un prisma rectangular en 3D, en la que se analiza el contenido que se encuentra dentro de la misma. Finalmente, en (Huang et al., 2015b) se añade un método de refuerzo multiclase para integrar clasificadores débiles, y en (Huang et al., 2013) el sistema se prueba con condiciones nocturnas.

Dejando de lado el *BHDF*, en (Huang & Vu, 2015a) los autores proponen un nuevo marco multicapa para la detección de la ocupación de las plazas de aparcamiento. Este marco consta de cuatro capas: (i) una capa de extracción de características para obtener las condiciones lumínicas y eliminar los problemas de sombras e iluminación, (ii) una capa de clasificación de divisiones de la imagen original para detectar oclusiones, (iii) una capa de combinación ponderada en la que se analizan las plazas contiguas en bloques de tres para mejorar los problemas de oclusiones y sombras, y (iv) una capa de inferencia de estado para determinar la ocupación de la plaza en base a las características locales (i), (ii) y (iii).

En (Ichihashi et al. 2009) se propone el sistema ParkLotD que utiliza algoritmos de agrupamiento difuso (*fuzzy clustering*) y un agrupamiento semi-duro para detectar la ocupación de las plazas de aparcamiento. En el agrupamiento semi-duro, en lugar de colocar cada punto en un clúster, los puntos se asignan a todos los clústeres junto con diferentes pesos que van en función de la cercanía a cada clúster. Con esto se consigue la ventaja con respecto al agrupamiento duro de conocer la medida del

punto con respecto a otros clústeres. Este trabajo fue mejorado en (Ichihashi et al. 2010).

Los autores de (Chen et al., 2010) abordan el problema de la gestión de plazas de aparcamientos exteriores mediante el uso de varias cámaras. Las imágenes de todas se sincronizan y fusionan, y se aplican transformaciones afines para obtener la imagen final. La detección de ocupación se hace mediante el modelado de los cambios de color dentro de la cada plaza de aparcamiento. Para los problemas causados por la perspectiva y las oclusiones se proponen modelos geométricos (elipses y rejillas) para representar el espacio de una plaza.

En (Al-Absi et al. 2010) se utiliza un clasificador basado en las características de Haar para detectar si existe dentro de una plaza un coche y determinar su ocupación. “La detección de objetos mediante clasificadores en cascada basados en las características de Haar es un método eficaz de detección de objetos propuesto en (Viola & Jones, 2001). Este enfoque se basa en el aprendizaje automático en el que se entrena una función de cascada a partir de una gran cantidad de imágenes positivas y negativas para utilizarla en la detección de objetos en otras imágenes” (Face Detection using Haar Cascades, 2018-2019).

Choeychuen propone en su trabajo (Choeychuen, 2012) un algoritmo de detección de objetos basado en modelos de fondo adaptativo. Se enmascara la escena con una imagen del fondo para obtener las partes que corresponden a los objetos y analizarlas de manera individual. Para estos objetos se analiza la densidad del histograma de orientación de bordes para determinar la ubicación y el estado de ocupación final de la plaza mediante un módulo de toma de decisiones.

En (Jermurawong, 2012), se propuso un sistema de detección de plazas de aparcamiento libres basado en redes neuronales previamente alimentadas (*feed-forward*) de dos capas. Se utilizaron dos redes neuronales, una diurna y otra nocturna, con el fin de acotar los cambios de iluminación para la clasificación del estado de las plazas. El sistema fue probado con un grabación de 24 horas de un aparcamiento al aire libre, obteniendo una precisión del 99,9% para las plazas ocupadas y del 97,9% para las plazas libres.

Un enfoque basado en el análisis de características sobre la imagen es planteado en (Blumer et al., 2012). Los autores proponen el uso de dos tipos de análisis: uno estático y otro dinámico. En el análisis estático se utiliza un algoritmo de detección de bordes y una clasificación del histograma de colores sobre cada plaza para evaluar los cambios con respecto a la plaza vacía. Ambas características son combinadas

usando algoritmos de aprendizaje automático para crear un árbol de decisión que evalúe el estado final de ocupación. El análisis dinámico detecta el movimiento dentro del aparcamiento y se utiliza como mecanismo de corrección a la decisión resultante en caso de existir cambios en las plazas.

Los autores de (Yusnita et al. 2012) y (Al-Kharusi & Al-Bahadly, 2014a) proponen un método simple de detección basado en el preprocesado de la imagen del aparcamiento. En la imagen del aparcamiento vacío se colocan círculos sobre cada una de las plazas. El algoritmo consiste en la identificación de estos puntos mediante el procesado de la imagen para determinar la ocupación de la plaza. En (Al-Kharusi & Al-Bahadly, 2014b) dan un paso más en su trabajo añadiendo una amplia documentación sobre la arquitectura, comunicaciones, software y hardware empleado para la implementación de un sistema destinado a esta funcionalidad.

En (Almeida et al., 2013), los autores proponen el uso de una *SVM* entrenada con dos descriptores de textura visual para determinar el estado de ocupación de las plazas. Los descriptores utilizados en el sistema para la clasificación son: los patrones binarios locales y la cuantificación de fase local. Ambos descriptores operan de manera local sobre una ventana de píxeles de la imagen asignando un valor al píxel central en función del conjunto analizado. Con la imagen resultante se genera el histograma final con el que se entrena la *SVM* para la clasificación y detección de ocupación. El sistema se mejora y prueba en (Almeida et al., 2015). Los autores crearon un conjunto de datos compuesto por 695.899 imágenes de plazas de aparcamiento. Las imágenes de las plazas fueron extraídas de un conjunto previo de 12.417 imágenes de aparcamientos con diferentes condiciones climáticas. Estas imágenes fueron etiquetadas y segmentadas para identificar cada una de las plazas de aparcamiento que aparecían en ellas.

Un enfoque basado en el volumen de los vehículos es utilizado en (Delibaltov et al., 2013). El sistema modela en 3D las plazas de aparcamiento para determinar el volumen que ocupan en la imagen en función de la perspectiva. Los vehículos son detectados mediante la clasificación de los píxeles de la imagen con clasificadores *SVM*. Finalmente, se determina el volumen de ocupación de la plaza en función de la existencia de vehículos en su interior.

En (Lui et al., 2013) se presenta un sistema que combina el análisis de la densidad de bordes y la densidad de píxeles de fondo y primer plano en cada plaza de aparcamiento. Las plazas son definidas mediante cuatro puntos formando paralelogramos. Son analizadas de manera individual para determinar el estado particular de ocupación de cada plaza.

(Tschentscher et al., 2013) plantea un estudio en el que se comparan cuatro características diferentes a obtener de las imágenes —color, ángulo, histogramas de diferencia de gradiente *gaussiano* y características de Haar—, en combinación con tres algoritmos de clasificación —*k-nn*, análisis de discriminantes lineales y *SVM*—. El sistema que mejores resultados obtiene se basa en las características de color e histograma de la diferencia de los gradientes *gaussianos*, junto un clasificador *SVM*. Los autores continúan su investigación en (Tschentscher et al., 2015).

Los artículos (Masmoudi et al., 2014a) y (Masmoudi et al., 2014b) presentan en su conjunto un sistema que aborda la detección de ocupación y da solución a los principales problemas que presenta. En el primer trabajo se resuelve el problema de las oclusiones entre vehículos. Se plantean un sistema novedoso que, en lugar de monitorizar los vehículos estacionados en las plazas, identifique los vehículos en movimiento, eliminando el resto de información de la escena. Los vehículos resultantes son ubicados con respecto a su posición en las plazas —dentro, fuera o entrando—. En el segundo trabajo, se resuelven los problemas de los cambios lumínicos y sombras con un algoritmo de sustracción de fondo adaptable, y se utiliza el algoritmo *Speeded-Up Robust Features (SURF)* para solventar los problemas de rotación y cambios de escala. Más adelante, en (Masmoudi et al., 2016) se presenta un sistema multiagente dedicado a al análisis del aparcamiento en la ciudad.

(Sevillano et al., 2014) plantea la extracción de características visuales de la imagen —histogramas de gradientes orientados (HGO), histogramas de Gabor y *SURF*— para entrenar algoritmos *k-nn* y *SVM* en base a unas imágenes previamente etiquetadas. El estudio se centra en la detección de la ocupación de plazas de aparcamiento a pie de calle. Se realizan pruebas y mediciones combinando diferentes características visuales con ambos algoritmos. Los mejores resultados se obtienen utilizando HGO como característica visual y *k-nn* para el aprendizaje del sistema. En un estudio posterior, (Sevillano & Màrmol, 2016), dos de los autores plantean un artículo basado en las conclusiones anteriores. Se presenta un sistema de detección de plazas de aparcamiento basado en las siguientes etapas: (i) sustracción de fondo, (ii) localización de objetos y (iii) reconocimiento visual. El sistema utiliza un clasificador *k-nn* entrenado con imágenes anotadas manualmente por los autores, y una base de datos externa de imágenes etiquetadas. El clasificador es capaz de diferenciar entre vehículos y peatones.

En el sistema propuesto en (Gálvez del Postigo et al., 2015) se utilizan dos técnicas: (i) la sustracción de fondo utilizando una mezcla de *gaussianos* para detectar y rastrear vehículos, y (ii) la creación de un mapa de transitorios para detectar el

estacionamiento y la salida. La base del sistema es el análisis temporal de los fotogramas de vídeo para detectar la variación de ocupación de las zonas de aparcamiento.

En (Mateus et al., 2015) se desarrolla un algoritmo basado en un clasificador *SVM*. En el algoritmo se utilizan dos características: la entropía local media y la desviación estándar de las entropías medias de las regiones de cada plaza de aparcamiento. Ambas características se extraen del histograma de dispersión de las plazas de aparcamiento.

El sistema propuesto por (Wang et al., 2015) consta de tres fases para detectar la ocupación de las plazas de aparcamiento. Primero, se extraen las características de primer plano para obtener el fondo usando características de borde y color y se definen las regiones de interés que representan las plazas. Después, para cada nuevo *frame* de video, se genera la imagen de la diferencia adyacente con respecto al anterior *frame*, y se evalúa la región de interés correspondiente a cada plaza. Finalmente, para la determinación de la ocupación, se propone un método de actualización de umbrales adaptativos.

2.2.1. Sistemas basados en redes neuronales convolucionales

Los algoritmos vistos hasta ahora se basan en la utilización de técnicas de segmentación de imágenes con sustracción de fondo, o en el uso de clasificadores como *k-nn* o *SVM* entrenados con características como el color o los bordes. En los trabajos más recientes en el área de la gestión de la ocupación de las plazas de aparcamiento se ve un cambio de tendencia en las técnicas utilizadas para la detección de vehículos.

Las **redes neuronales convolucionales** (RNC) son uno de los métodos más utilizados actualmente para la clasificación de objetos. Debido a la complejidad y necesidades de procesamiento, estos sistemas eran inviables hace unos años, pero ahora se emplean cada vez más. Las RNC son entrenadas antes de poner en marcha el sistema, lo que hace que sean capaces de responder e identificar los objetos en los entornos de producción rápidamente.

“Una red neuronal convolucional es un tipo de red neuronal artificial donde las neuronas responden a campos receptivos de una manera muy similar a las neuronas en la corteza visual primaria (V1) de un cerebro biológico. Son muy efectivas para tareas de visión artificial, como la

clasificación y segmentación de imágenes, entre otras aplicaciones.” (Redes neuronales convolucionales, 2019).

En el área que nos ocupa, las RNC de nuestro interés son las redes neuronales convolucionales basadas en regiones (R-RNC). Las R-RNC están diseñadas para detectar y clasificar objetos dentro de las imágenes, localizando y marcando la posición exacta de los mismos. Las R-RNC aportan información del tipo: “*en la figura con centro en las coordenadas (x, y) tenemos un coche*”.

Una de las primera aproximaciones a la utilización de redes neuronales convolucionales para la detección de vehículos dentro del aparcamiento es planteada en (Xie et al., 2015). El algoritmo propuesto se basa en dos etapas: (i) la RNC entrenada con una ventana deslizante se utiliza para buscar e identificar vehículos, y (ii) se define una matriz de distribución que refleje la densidad de colocación de los vehículos, que se utiliza para eliminar las ventanas redundantes y localizar la posición del vehículo con precisión.

En (Amato et al., 2017) los autores proponen una solución descentralizada y eficiente para la detección de ocupación de aparcamientos diseñada para ser integrable en cámaras inteligentes. La solución se basa en una RNC inspirada en la arquitectura AlexNet (AlexNet, 2019). Los autores simplifican el tamaño de AlexNet obteniendo una arquitectura tres veces menor sin perder rendimiento en la detección. Esto se debe principalmente a que la red original (AlexNet) es capaz de detectar 1000 clases diferentes y los autores simplifican la detección a dos clases —vacío y ocupado— para determinar el estado de ocupación.

(Acharya et al., 2018) presenta un marco para la detección de la ocupación de las plazas de aparcamiento haciendo uso de una RNC y un clasificador *SVM* para determinar la ocupación en aparcamientos de exterior. El clasificador se entrena y prueba con las características aprendidas por la RCN de un conjunto de imágenes (PKLot), en las que se incluyen situaciones con diferentes condiciones lumínicas y climáticas. El sistema reporta una precisión de detección del 99,7% para el conjunto de datos PKLot y 96,7% para un conjunto de datos propio.

En el trabajo (Nieto et al., 2018) se propone un sistema con varias cámaras que realiza un procesamiento en paralelo de cada imagen seguido de la combinación de los resultados. El algoritmo se basa en tres etapas: (i) se detectan los coches en la imagen con la utilización de R-RNC, (ii) se aplican homografías y correcciones de perspectiva para considerar el volumen de los objetos, y (iii) se mapean los vehículos sobre las plazas para detectar el estado de ocupación. El sistema se probó en el

aparcamiento del Aeropuerto Internacional de Pittsburgh reutilizando las cámaras existentes.

Una de las visiones más recientes la proporcionan Vu y Huang en su trabajo (Vu & Huang, 2019). El enfoque propuesto analiza las plazas de aparcamiento en bloques de tres para inferir el estado de la plaza central evitando problemas de oclusiones y se utilizan varias técnicas basadas en RNC. Los autores utilizan: (i) una RNC para transformaciones espaciales dedicada a adaptar los bloques de tres espacios en función del tamaño y posición de los vehículos dentro de las plazas para la siguiente fase, reduciendo así los problemas de distorsión provocados por la perspectiva; (ii) una RNC basada en una arquitectura siamesa entrenada en base a características de las plazas; y (iii) un modelo de regresión logística de dos clases que utiliza la salida de la RNC anterior para inferir el estado final de la plaza central.

2.2.2. Clasificación del estado del arte

Una vez analizado el estado del arte, se propone una clasificación de los trabajos en dos conjuntos de categorías para identificar los enfoques más estudiados y los enfoques que se presentan como alternativas futuras.

Para la primera clasificación se tiene en cuenta el trabajo de (Huang & Wang, 2010). Los autores proponen una categorización general de los sistemas en dos categorías:

- Los sistemas *enfocados a vehículos (car-driven)* se basan en la detección de los vehículos con el análisis posterior de su ubicación sobre las plazas de aparcamiento.
- Los sistemas *enfocados a espacios (space-driven)* se basan en el análisis de cada una de las plazas de aparcamiento para determinar su ocupación en base a si dentro de ella existe un vehículo o no.

La segunda clasificación se basa en la categorización que hacen en su trabajo Nieto et al. (Nieto et al., 2019). Los autores definen tres categorías teniendo en cuenta la técnica utilizada para la clasificación:

- *Segmentación de imágenes (image segmentation)*. Estos sistemas hacen uso de una imagen del aparcamiento vacío que se sustrae a la imagen a analizar con el fin de obtener una máscara de los objetos no presentes en el aparcamiento inicial —vehículos—. Posteriormente, se cartografía la posición de estos objetos en el plano para detectar la plaza que ocupan.

- *Clasificación basada en imágenes de plazas (spots patch classification based systems)*. Estos sistemas hacen uso de técnicas de aprendizaje automático — *SVM*, *RN*, etc.— entrenadas con imágenes de *recortes* de plazas de aparcamiento individuales etiquetadas previamente como libres u ocupadas.
- *Detección de objetos (object detection)*. Estos sistemas utilizan algoritmos de aprendizaje automático, como *RNC*, para identificar los vehículos y cartografiarlos en el aparcamiento detectando la plaza que ocupan.

En la [tabla 2.1](#) se muestra la clasificación de los sistemas basados en visión artificial, analizados en función de los dos grupos de categorías detalladas. En ella se observa que ambos enfoques —*car-driven* y *space-driven*— se alternan, aunque existen más trabajos basados en el enfoque *space-driven*. En cuanto a los métodos de detección, hasta el año 2015 se plantean sobre todo estudios basados en segmentación de imágenes y clasificación basada en imágenes de *recortes* las plazas, siendo en este año cuando se detecta la importancia y mejora que aportan las redes neuronales en la segmentación de imágenes, empezándose a utilizar como base para la detección de objetos.

Tabla 2.1. Clasificación de los estudios analizados

Trabajo	Enfoque seleccionado	Método de detección
Wang & Hanson, 1998	car-driven	image segmentation
Dan, 2002	space-driven	spot patch
Funck et al., 2004	car-driven	image segmentation
Bong et al., 2006	space-driven	image segmentation
Bong et al., 2008	space-driven	image segmentation
Lin et al., 2006	space-driven	image segmentation
Wu & Zhang, 2006	space-driven	spot patch
Wu et al., 2007	space-driven	spot patch
Sastre et al., 2007	space-driven	spot patch
True, 2007	space-driven	spot patch
Fabian, 2008	space-driven	image segmentation
Huang et al., 2008	space-driven	spot patch
Huang & Wang, 2010	space-driven	spot patch

Huang et al., 2012	space-driven	spot patch
Huang et al., 2013	space-driven	spot patch
Huang & Vu, 2015a	space-driven	spot patch
Huang et al., 2015b	space-driven	spot patch
Ichihashi et al. 2009	space-driven	spot patch
Chen et al., 2010	space-driven	image segmentation
Al-Absi et al. 2010	car-driven	spot patch
Choeychuen, 2012	car-driven	image segmentation
Jermurawong, 2012	space-driven	spot patch
Blumer et al., 2012	space-driven	image segmentation
Yusnita et al. 2012	space-driven	image segmentation
Al-Kharusi & Al-Bahadly, 2014a	space-driven	image segmentation
Al-Kharusi & Al-Bahadly, 2014b	space-driven	image segmentation
Almeida et al., 2013	space-driven	spot patch
Almeida et al., 2015	space-driven	spot patch
Delibaltov et al., 2013	space-driven	spot patch
Lui et al., 2013	space-driven	image segmentation
Tschentscher et al., 2013	space-driven	spot patch
Tschentscher et al., 2015	space-driven	spot patch
Masmoudi et al., 2014a	car-driven	image segmentation
Masmoudi et al., 2014b	car-driven	image segmentation
Sevillano et al., 2014	space-driven	spot patch
Gálvez del Postigo et al., 2015	space-driven	image segmentation
Mateus et al., 2015	space-driven	spot patch
Wang et al., 2015	car-driven	image segmentation
Xie et al., 2015	car-driven	object detection
Amato et al., 2017	car-driven	object detection
Acharya et al., 2018	space-driven	object detection
Nieto et al., 2018	car-driven	object detection

2.3. Comparación entre los estudios analizados y el sistema propuesto

Puesto que no se dispone de conjuntos de imágenes y datos comunes con los trabajos analizados, no es posible establecer una comparativa cuantitativa. No obstante, como las características que presenta el sistema detallado en este trabajo son novedosas, y muestran nuevas maneras de enfocar la detección de la ocupación de las plazas de aparcamiento con respecto a las existentes, se procede a un análisis cualitativo de las mismas.

Una de las ventajas más importantes es la utilización de un enfoque *orientado a vehículos* para detectar los vehículos presentes dentro del aparcamiento. Uno de los problemas detectados en los sistemas basados en el enfoque *orientado a espacios*, es que se analiza cada una de las plazas de aparcamiento para detectar si está vacía u ocupada, lo que genera una dependencia de la posición de un vehículo dentro de la plaza, es decir, si en este tipo de sistemas un vehículo aparca entre dos plazas u ocupando parte de dos, se presenta un problema al considerar cada plaza como un espacio individual. Con el enfoque seleccionado se consigue solventar esta problemática.

En línea a lo comentado, el sistema deja de lado los planteamientos binarios en el que se decide si una plaza está ocupada o vacía, y utiliza un planteamiento en el que se conoce el *porcentaje de espacio (área) ocupado* en cada una. Con esto se consigue eliminar la dependencia externa que produce una colocación errónea del vehículo dentro de las plazas, si este es colocado, por ejemplo, entre dos de ellas.

Otra ventaja que pocos sistemas aportan, es la *detección y mapeo de las coordenadas de las plazas de aparcamiento de manera automática* para obtener un modelo computacional que pueda ser utilizado por el sistema. La mayoría de trabajos plantean esto como un paso manual a realizar en la fase de configuración o incluso lo obvian. En el sistema de este trabajo es uno de los puntos principales, pensado para hacer el sistema automatizado y escalable, además de ser una ventaja y valor añadido en aparcamientos grandes, evitando el etiquetado manual de todas las plazas.

Además, el sistema utiliza la *segmentación de instancias*, un método más avanzado a la detección de objetos, como método de detección de vehículos haciendo uso de una R-RNC. De esta manera, el sistema es capaz de detectar y generar los contornos de los vehículos en tiempo real para su posterior ubicación en el aparcamiento. La

ventaja que aporta este método, con respecto a los sistemas basados en la segmentación de imágenes, es la eliminación de los problemas que causan los cambios de condiciones lumínicas, climáticos y ocasionados por las sombras que se producen en ellos. Con respecto a los sistemas que utilizan imágenes con *recortes* de plazas, para entrenar clasificadores que infieran si una plaza está vacía u ocupada, la ventaja es la mejora de los datos necesarios para su entrenamiento. En lugar de necesitar imágenes de los aparcamientos para entrenar el sistema, se generaliza utilizando imágenes de vehículos para entrenar la red neuronal.

Finalmente, la ventaja más relevante es la implementación de un *algoritmo de búsqueda* dedicado a encontrar la *plaza óptima para el vehículo entrante* al aparcamiento en función de sus características. Esto nunca ha sido considerado en ninguno de los trabajos existentes hasta la fecha. Además, esta ventaja mejora la experiencia del usuario con el sistema, al indicarle la plaza en la que estacionar al entrar al aparcamiento, evitando así que tenga que emplear tiempo en buscar dónde aparcar.

Capítulo 3

Sistema propuesto

3.1. Descripción general

Este trabajo fin de máster, se basa en la creación de un sistema enfocado a la gestión de la ocupación de las plazas de aparcamiento en entornos al aire libre a través del análisis de imágenes. Para ello, se propone dotar de *inteligencia* al sistema por medio de *visión artificial* y *aprendizaje automático* —haciendo uso de redes neuronales— permitiendo que sea capaz de procesar las imágenes con las que reconocer el entorno, identificar las líneas que definen las plazas de aparcamiento, reconocer los vehículos dentro de las plazas y evaluar el estado final de ocupación.

Con el fin de optimizar la distribución de las plazas de aparcamiento en base al estado general de ocupación, el sistema es capaz de reconocer el tamaño, tanto de los vehículos estacionados, como de los vehículos entrantes. Gracias a esta información y a algoritmos de búsqueda y decisión, el sistema tiene la capacidad de asignar la mejor plaza a cada uno de los vehículos entrantes en función de sus características y necesidades, evitando tiempos de espera y retenciones dentro de los aparcamientos.

La posición de estacionamiento de los vehículos dentro de las plazas es uno de los factores determinantes en los sistemas de detección de ocupación. En los sistemas basados en sensores es uno de los mayores problemas que lleva a falsos positivos detectando plazas con un estado erróneo. En los sistemas revisados en la literatura que hacen uso de VA utilizan un estado binario —ocupado o vacío— para definir la ocupación de las plazas de aparcamiento. En el sistema propuesto se busca solucionar este problema determinando el estado de ocupación en base al área ocupada por el vehículo en cada plaza de aparcamiento, con la restricción de utilizar imágenes

tomadas desde planos cenitales para evitar oclusiones entre vehículos y problemas ocasionados por la perspectiva, y poder determinar el contorno exacto del vehículo. Con esto se consiguen solucionar los problemas de vehículos mal estacionados dentro de las plazas, llevando el conocimiento del estado del aparcamiento y la optimización de la distribución del espacio un paso más allá. En la [figura 3.1](#) se muestra un ejemplo de la casuística mencionada, en la que se ve que en la fila superior no caben más vehículos, pero un sistema basado en sensores determinaría, al menos, dos espacios vacíos, y un sistema basado en VA con un estado binario de ocupación detectaría, al menos, un espacio vacío.

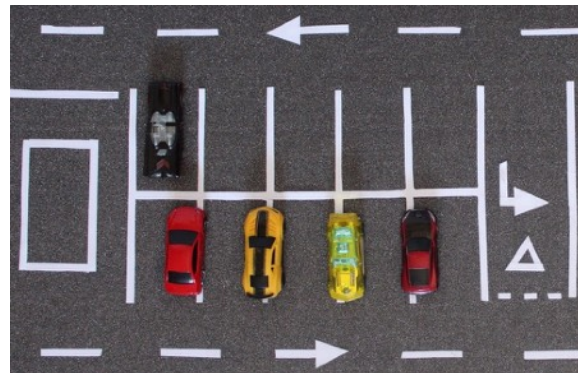


Figura 3.1. Vehículos estacionados de manera incorrecta en plazas de aparcamiento

El tamaño del aparcamiento es un aspecto que queda fuera del alcance de este sistema, puesto que los espacios de aparcamiento difieren mucho en tamaño, encontrando recintos con 100 plazas hasta otros que pueden alcanzar más de 1000. Por ello, el sistema se ha planteado de manera distribuida para hacer frente a las necesidades cambiantes del entorno, disponiendo de distintos nodos de computación, cada uno de ellos encargados de una tarea. El sistema dispone de nodos dedicados a monitorizar uno o varios grupos de plazas de aparcamiento para determinar el estado de ocupación de cada una, y nodos dedicados a evaluar las características del vehículo entrante. Ambos nodos utilizarán cámaras para obtener las imágenes. Por último, un nodo central o servidor será el encargado de mantener y unificar la información del aparcamiento completo.

Además, el sistema va más allá de una detección de estado de ocupación de las plazas de aparcamiento, siendo capaz de: *detectar* y *reconocer* vehículos, diferenciándolos del resto de elementos que pueden coexistir dentro de un aparcamiento; y de determinar el área que ocupan dichos vehículos en cada plaza de aparcamiento. Para detectar cambios en el estado de las plazas y actualizar el estado general, el sistema incluye algoritmos de detección de movimiento para saber cuando existen cambios dentro del

aparcamiento, y se realiza una evaluación automática periódica y *adaptativa*, en función de la afluencia de tráfico.

Para describir el sistema se plantea una división cronológica de los pasos que se han dado en el proyecto. En el [apartado 3.2](#), se muestra el modelado de un entorno de pruebas basado en una maqueta de un aparcamiento a escala, con la que simular distintas situaciones y obtener imágenes analizables en el sistema. En el [apartado 3.3](#), se incluyen la teoría y algoritmos utilizados para dar soporte a los módulos del sistema que hacen uso de la visión artificial y el aprendizaje automático en su implementación. En el [apartado 3.4](#), se plantea el diseño software que da soporte al sistema. En el [apartado 3.5](#), se define la arquitectura sobre la que se despliega el sistema definiendo los distintos nodos de computación y enunciando las características hardware. La decisión de incluir el [apartado 3.3](#) antes del diseño y la arquitectura, es evaluar la posibilidad de implementación de cada uno de los requisitos del sistema mediante pruebas de concepto con *visión artificial* y *aprendizaje automático*.

3.2. Modelado del entorno de pruebas

Puesto que el sistema se basa en el análisis de imágenes, para cubrir el mayor número de casuísticas posibles se ha modelado una maqueta a escala. Mediante la maqueta se simulan diferentes condiciones de colocación de los vehículos, y escenarios con condiciones de iluminación cambiantes.

En la [figura 3.2](#) (a) se muestra la maqueta final empleada en el desarrollo del proyecto para la implementación del sistema y las pruebas de concepto, junto con el etiquetado de las distintas zonas (b). La maqueta consta de: (i) una entrada y una salida para vehículos, (ii) seis bloques de espacios de aparcamiento con diferentes topologías, (iii) un espacio habilitado para la detección del vehículo entrante donde se determinará el tamaño del vehículo que accede al aparcamiento para su posterior emplazamiento, y (iv) dos puntos de acceso peatonales que pueden ser, por ejemplo, las diferentes entradas de un centro comercial. En la [figura 3.3](#) se simula un mismo escenario en diferentes momentos del día: diurno (a), con luz natural; y nocturno (b), con alumbrado artificial.

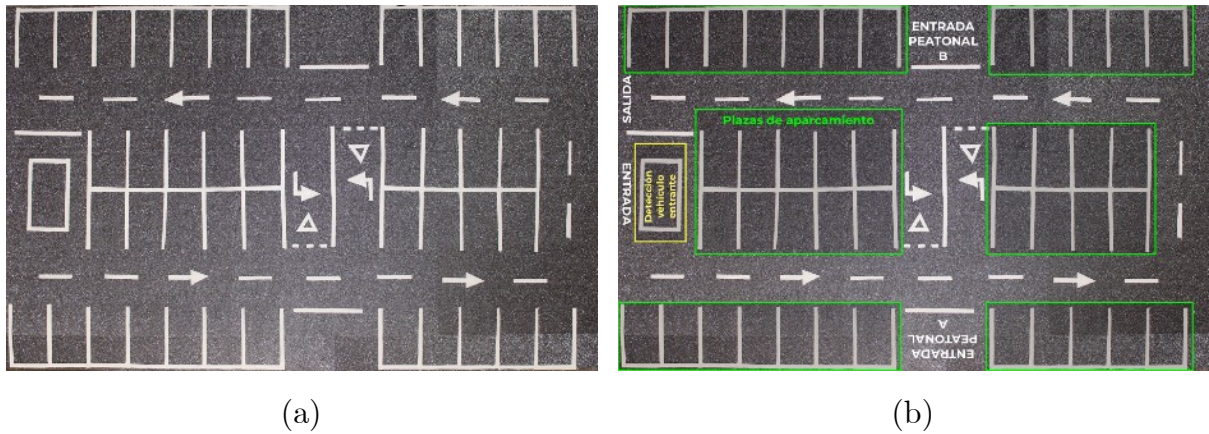


Figura 3.2. (a) Maqueta del aparcamiento. (a) Maqueta del aparcamiento con zonas etiquetadas

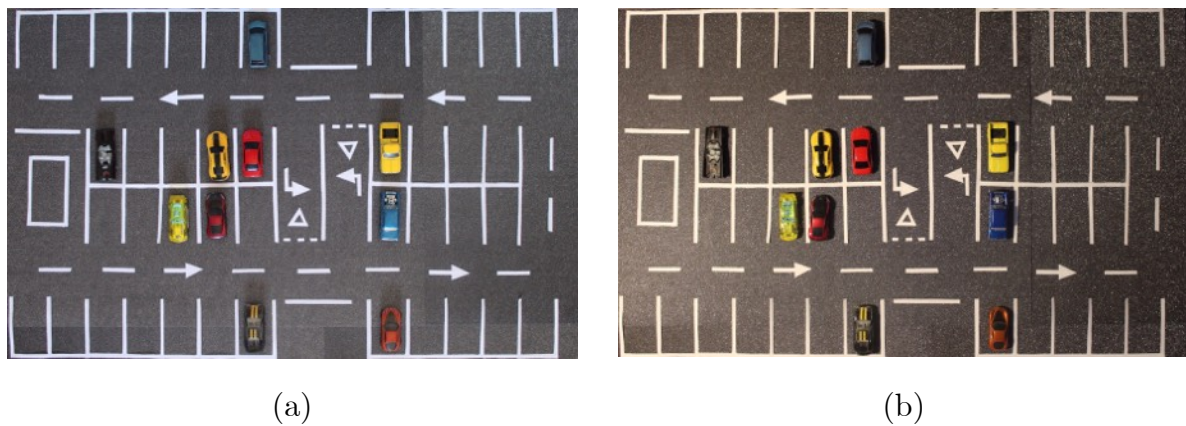


Figura 3.3. Modelado de escena de día (a) y noche (b)

3.3. Visión artificial y aprendizaje automático

En este apartado se detallan los módulos que emplean *visión artificial* y *aprendizaje automático* para la segmentación de imágenes y extracción de información del entorno. El objetivo final de estos módulos es la obtención de una representación computacional del aparcamiento y los objetos ubicados dentro de él. Cada uno de los módulos proporcionan la representación de las entidades físicas contenidas en las imágenes, como objetos finales que se manejan en el sistema para la gestión de la ocupación. En el [apartado 3.4](#), se especifica en detalle las entidades contenidas en el sistema.

“La segmentación es el proceso por el cual se extrae de la imagen cierta información subyacente para su posterior uso. La segmentación está basada en dos principios fundamentales de discontinuidad y similitud.

Cabe pues enfocar la segmentación orientada en bordes (discontinuidad) y orienta en regiones (similitud). Así una región es, en líneas generales, un área de la imagen en la que sus píxeles poseen propiedades similares (intensidad, color, etc.), mientras que un borde es una línea que separa dos regiones, por tanto de diferentes propiedades (discontinuidad).” (Pajares y De la Cruz, 2007, p. 8)

Como nota introductoria, las funcionalidades que se detallan a continuación se han desarrollado utilizando Python (Python, 2019), como lenguaje de programación; y OpenCV (OpenCV, 2019) como librería de visión artificial. En el [apartado 3.4](#) se describen en detalle las características software del sistema.

3.3.1. Detección de plazas de aparcamiento

Para la *detección de las plazas de aparcamiento*, se hace un procesado de la imagen del aparcamiento vacío, se detectan las líneas que definen cada una de las plazas, y se obtiene un mapa computacional de ellas. Cada plaza de aparcamiento se define en el sistema mediante cuatro coordenadas que forman un paralelogramo.

Puesto que en un aparcamiento existen múltiples líneas —como las marcas viales— las plazas se analizan en bloques. Un bloque está formado por un conjunto de plazas contiguas y adyacentes. En la [figura 3.4](#) se muestran dos de los bloques existentes en la maqueta de aparcamiento diseñada.

En apartados posteriores se muestra que esta división, además de favorecer la detección y definición de las plazas de aparcamiento, es un factor clave para la escalabilidad del sistema y el procesamiento en paralelo con el uso de diferentes nodos de computación.

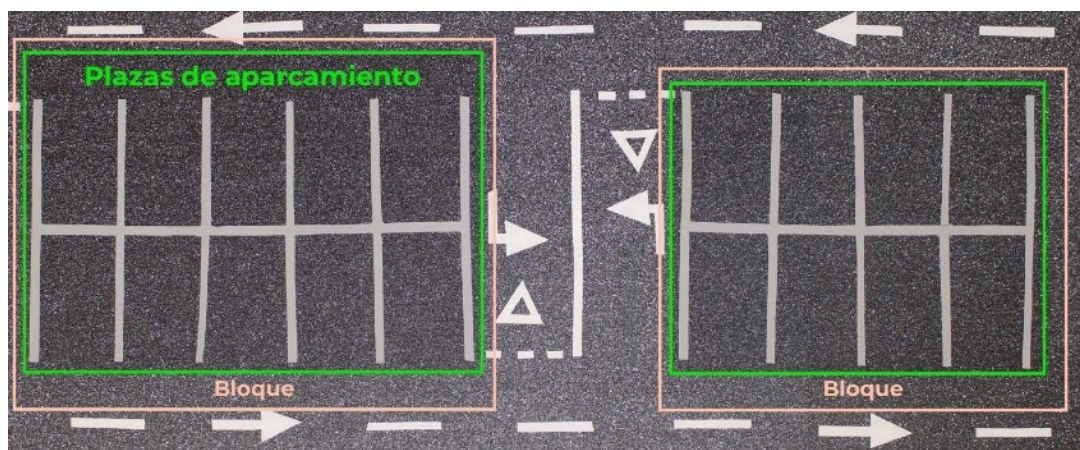


Figura 3.4. Bloques de aparcamiento

En la [figura 3.5](#) se muestran las etapas de la segmentación de las plazas de aparcamiento para obtener objetos manejables computacionalmente por el sistema. A continuación, se detallan y especifican los algoritmos y métodos empleados, así como el proceso de transformación de una imagen al pasar por cada una de las etapas.

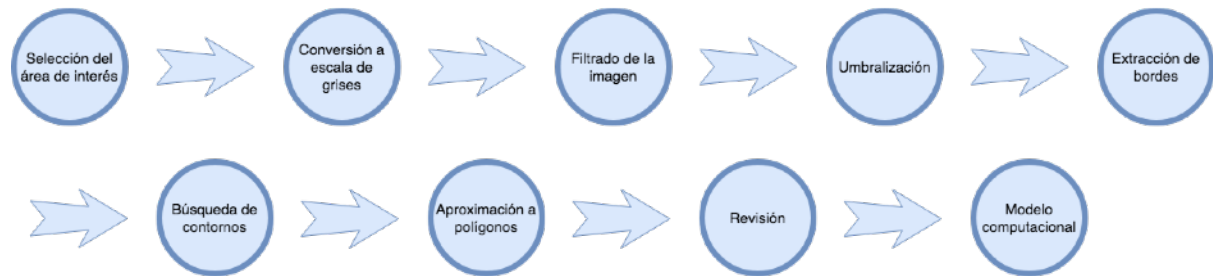


Figura 3.5. Etapas para la detección de plazas de aparcamiento

Selección de la región de interés

Mediante la selección de las regiones de interés (*Region of Interest, RoI*) de una imagen, se analiza cada uno de los bloques de aparcamiento de manera aislada. De esta manera se busca que cada RoI contenga únicamente las líneas correspondientes a las plazas de aparcamiento de un bloque. En la [figura 3.6](#) (a) se muestra la selección de la RoI de la imagen original.

“Una región de interés (RoI) es una parte de una imagen en la que se desea realizar alguna operación. Se define una RoI mediante la creación de una máscara binaria, que es una imagen binaria que tiene el mismo tamaño que la imagen que se desea procesar. En la imagen de máscara, los píxeles que definen la RoI se establecen en 1 y todos los demás píxeles se establecen en 0.” (Procesamiento basado en ROI, 2018).

Conversión a escala de grises

La *escala de grises* es un sistema que se utiliza para convertir los grados de luminosidad de los colores en una equivalencia en escala de grises. Esta escala está compuesta por tres colores: blanco, negro y gris. En función del nivel de detalle que se necesite se utilizarán más o menos niveles de gris distintos para conseguir efectos como la profundidad.

La transformación en escala de grises es una operación que se realiza antes del procesamiento de la imagen. Los motivos para la utilización de imágenes en escala de grises son: (i) el procesamiento de la imagen es menos costoso computacionalmente, pasamos de utilizar tres canales de color con el modelo *RGB* (*red, green and blue*) a

un único canal; y (ii) favorece la extracción de líneas de aparcamiento al trabajar con imágenes en las que existe un fondo oscuro —suelo— y un primer plano claro —líneas de aparcamiento—. Las imágenes tratadas contienen 256 niveles de gris que van desde el nivel 0 —negro— al 255 —blanco—. En la [figura 3.6](#) (b) se muestra la imagen en escala de grises sobre la que se aplican los siguientes pasos para determinar las plazas.

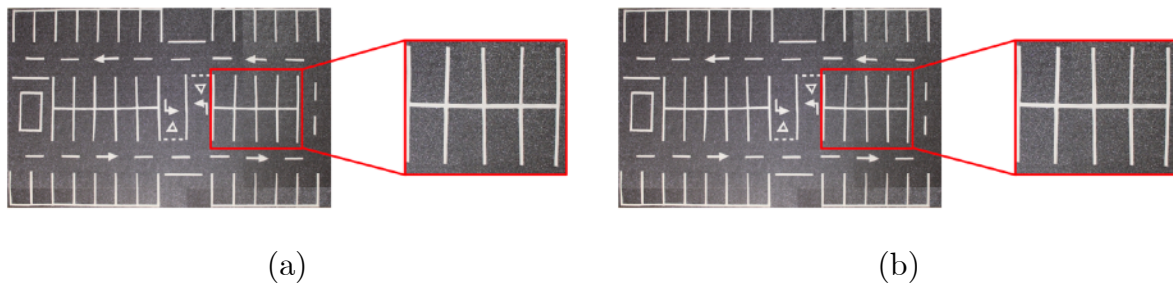


Figura 3.6. (a) Selección de ROI para un bloque de aparcamientos. (b) RDI en escala de grises

Filtrado de imágenes

El filtrado de imágenes es un proceso que se utiliza para resaltar, mejorar, suavizar o suprimir algunas características de una imagen. Los filtros se utilizan tanto para detectar elementos o bordes, como para eliminar elementos como el ruido en una imagen. En ambos casos, se detectan cambios bruscos producidos por las componentes de alta frecuencia. Los *filtros de paso alto* se utilizan para el realce de estos componentes (detección de bordes), y los *filtros de paso bajo* para el suavizado de los mismos (eliminación de ruido).

Los *filtros lineales* modifican la matriz de la imagen original aplicando en cada píxel una matriz de convolución —núcleo o *kernel*—, que afecta al píxel central y los colindantes. Estas matrices son cuadradas y suelen tener dimensiones de: 3x3, 5x5, etc. Las dimensiones son impares para que el elemento central de la matriz coincida con el píxel central de la ventana de la imagen sobre la que se aplica. Se pueden imaginar estas matrices como una ventana que se va deslizando y modificando la imagen.

“En el procesamiento de la imagen los términos: núcleo, kernel, matriz de convolución o máscara hacen referencia a una matriz pequeña que se utiliza para desenfocar, enfocar, realce, detección de bordes y más. Esto se logra realizando una convolución entre un núcleo y una imagen.” (Kernel (image processing), 2019).

Los *filtros no lineales* se basan en el mismo concepto de ventana deslizante que los anteriores, pero se diferencian en que, además de utilizar una matriz de convolución, realizan cálculos como la media —*filtro de la media*— sobre los píxeles de la ventana.

Aparentemente la imagen original del aparcamiento vacío —[figura 3.2](#) (a)— no muestra ruido. Pero si se analiza la imagen en detalle —[figura 3.7](#)—, mostrando los valores de gris de una submatriz de la imagen, se aprecian los cambios de nivel —ruido— que, de no suavizarlos, complican el procesado en las siguientes etapas. Por ello, en el sistema se utiliza un primer filtrado para reducir el ruido que pueda presentar la imagen. En este caso, al tratarse de un proceso de suavizado se utilizan *filtros de paso bajo*.

En OpenCV, entre los filtros destinados al suavizado de imágenes se dispone de los siguientes: *filtro gaussiano*, *filtro de la mediana* y *filtro bilateral*. En la [figura 3.8](#) se muestra el resultado de la aplicación de los filtros anteriores a la imagen original del aparcamiento —[figura 3.2](#) (a)— para eliminar el ruido existente. Los mejores suavizados se obtienen con el filtro *lineal gaussiano* y el filtro *no lineal de la mediana*.

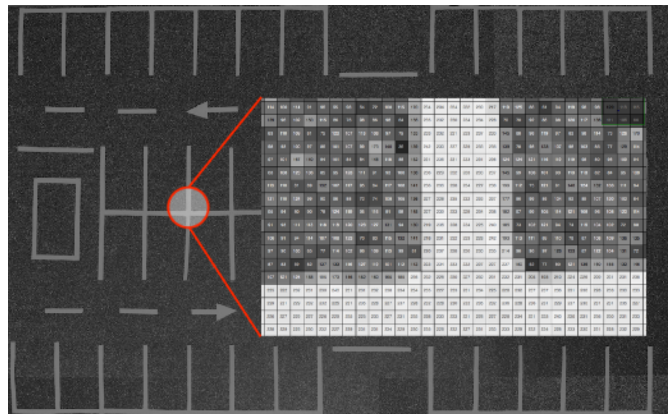


Figura 3.7. Valores de niveles de gris de una porción de la imagen del aparcamiento vacío

En este trabajo, tanto el filtro gaussiano como el filtro de la mediana se utilizan con una dimensión de ventana de convolución de 5x5. Esto quiere decir que el suavizado se aplica sobre submatrices de la imagen de 25 píxeles, y se modifica el valor del píxel central. Empíricamente, se ha comprobado que esta ventana produce mejores resultados que una ventana de dimensión 3x3, al suavizar y eliminar más ruidos en las zonas de fondo, preservando los bordes correspondientes a las líneas.

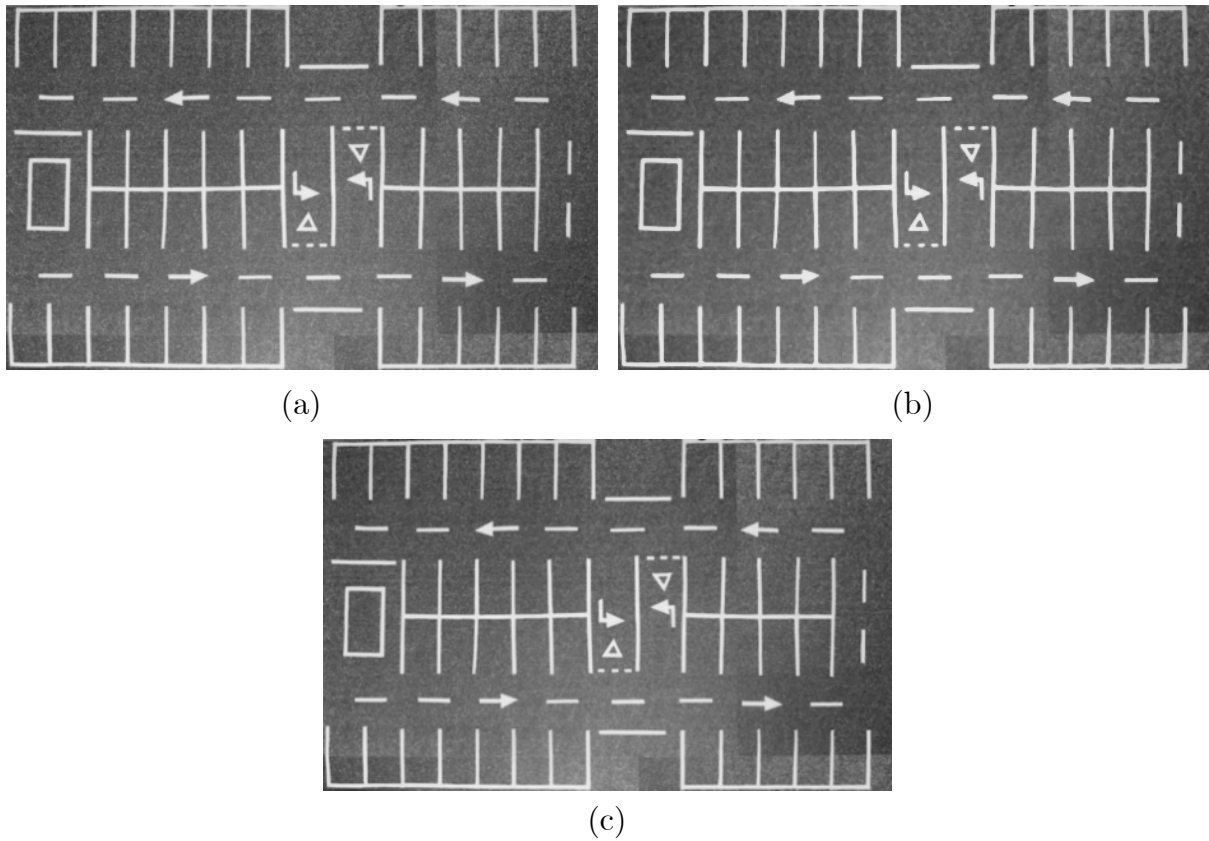
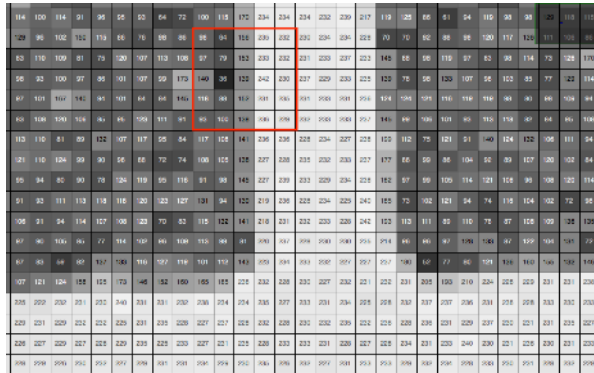


Figura 3.8. Resultados obtenidos a partir de la imagen del aparcamiento vacío mostrada en la [figura 3.2](#). (a). (a) Suavizado de ruido en la imagen mediante el *filtro gaussiano*; (b) Suavizado de ruido en la imagen mediante el *filtro de la mediana*; (c) Suavizado de ruido en la imagen mediante el *filtro bilateral*.

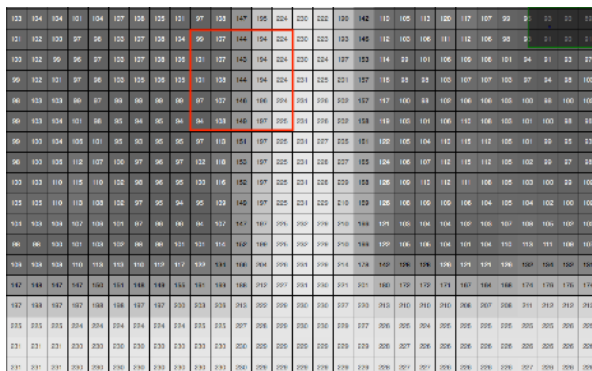
El *filtro gaussiano* es un filtro lineal que utiliza una matriz de convolución para calcular los valores de la ventana. En este caso se trabaja sobre regiones de dimensión 5x5 píxeles. El valor máximo le corresponde al elemento central de la matriz y decrece hacia los extremos.

El *filtro de la mediana* es un filtro no lineal que asigna al píxel central de la ventana de trabajo el valor de la mediana del conjunto de los píxeles de dicha ventana. Esto hace que no sea necesario calcular una matriz de convolución, obteniendo un valor final presente en la imagen y no uno calculado en base a la ventana.

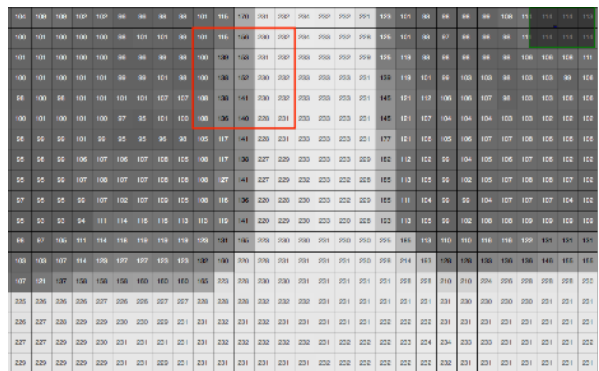
El *filtro de la mediana* es un filtro no lineal que asigna al píxel central de la ventana de trabajo el valor de la mediana del conjunto de los píxeles de dicha ventana. Esto hace que no sea necesario calcular una matriz de convolución, obteniendo un valor final presente en la imagen y no uno calculado en base a la ventana.



(a)



(b)



(c)

Figura 3.9. Valores de niveles de gris de una submatriz de la imagen. (a) Imagen original; (b) Imagen suavizada mediante filtro gaussiano; (c) imagen suavizada mediante filtro de la mediana.

En la [figura 3.9](#) se muestra en detalle el resultado de la eliminación de ruido realizado con cada uno de los filtros. En rojo, la ventana deslizante de dimensión 5×5 que va suavizando el ruido. En (a) se ve el nivel de ruido de la imagen original, en (b) y (c) se demuestra la reducción de ruido conseguida con los filtros gaussiano y de la mediana, respectivamente. Con los filtros se suavizan los valores de niveles de gris obteniendo un resultado más homogéneo en los píxeles de fondo —suelo—. A simple vista, parece que el filtro gaussiano genera unos bordes más limpios entre el suelo y las líneas de las plazas, pero si analizamos los valores de gris en detalle, se observa que el filtro de la mediana genera unos saltos mayores, acentuando así los bordes. Por ejemplo, si nos situamos en el píxel central de la ventana delimitada en rojo, correspondiente a un borde, y analizamos los píxeles a la derecha, el filtro gaussiano genera un cambio más suave entre los valores de suelo y línea (144, 194 y 224), mientras que el filtro de la mediana acentúa el cambio entre dichos valores (152, 230, 232).

Por lo tanto, el *filtro de la mediana* es el que mejor suavizado produce del fondo con un mayor preservación de los bordes resaltando así las líneas de las plazas de aparcamiento. En (Median filter, 2019) se afirma:

“Para niveles pequeños o moderados de ruido gaussiano, el filtro de mediana es demostrablemente mejor que el desenfoque gaussiano para eliminar el ruido, conservando al mismo tiempo los bordes para un tamaño de ventana determinado y fijo. Sin embargo, su rendimiento no es mucho mejor que el desenfoque gaussiano para niveles altos de ruido, mientras que, para el ruido de moteado y el ruido de sal y pimienta (ruido impulsivo), es particularmente efectivo. Debido a esto, el filtrado de mediana es muy utilizado en el procesamiento de imágenes digitales.” (*Edge preservation properties*).

Umbralización de imágenes

La *umbralización* o métodos del valor umbral son algoritmos que nos permiten segmentar imágenes para la extracción de objetos de interés. La umbralización se basa en la determinación de un valor de umbral que se utiliza para determinar si el píxel analizado forma parte del fondo o del objeto de interés. La imagen a segmentar debe estar en escala de grises puesto que el valor umbral se establece con un nivel de gris.

Dentro de los métodos de umbralización, tenemos métodos en los que el valor umbral se determina de manera empírica, y otros que se encargan de calcular automáticamente el valor umbral basándose en el histograma de la imagen a segmentar.

“Un histograma de color es una representación de la distribución del color en una imagen. En las imágenes digitales, un histograma de color representa el número de píxeles que tienen colores en cada una de las listas fijas de rangos de colores, que se extienden sobre el espacio de color de la imagen, es decir, el conjunto de todos los posibles colores.” (Color histogram, 2019)

En OpenCV, se pueden utilizar ambos tipos de umbralización. En los métodos de umbralización manual, se encuentran algoritmos simples como la umbralización binaria, en la que si el píxel supera el valor umbral se le asigna el valor de 255 (blanco) y 0 (negro) en caso contrario; o la umbralización a cero, en la que si el píxel supera el valor umbral mantiene su valor y 0 (negro) en caso contrario. En la

umbralización automática, se dispone de métodos que calculan el valor de umbral en base al histograma de la imagen. Entre ellos, tenemos la umbralización adaptativa, en la que el valor umbral se calcula de manera *local* en cada submatriz de la imagen; o el método de Otsu (Otsu, 1979), que calcula de manera automática el valor umbral *global* y segmenta la imagen en base a dicho valor.

Una de las restricciones para que el método de Otsu funcione correctamente es que el histograma de la imagen a segmentar sea bimodal. Un *histograma bimodal* es el que presenta un distribución caracterizada por dos picos separados por un valle.

Para determinar el método de umbralización óptimo para la búsqueda de las plazas de aparcamiento en el sistema, se analizan los histogramas de varias imágenes correspondientes a aparcamientos vacíos. En la [figura 3.10](#) se muestran los histogramas de tres imágenes en los que se ve que, en mayor o menor medida, las distribuciones son bimodales. Por lo general, los aparcamientos vacíos presentan este tipo de distribución en la que los valores a la izquierda corresponden al fondo (suelo), y los que están a la derecha a los objetos a segmentar (líneas de aparcamiento). Por lo tanto, se concluye que, al presentar este tipo de distribución, el método óptimo para la búsqueda del valor umbral es el *método de Otsu*.

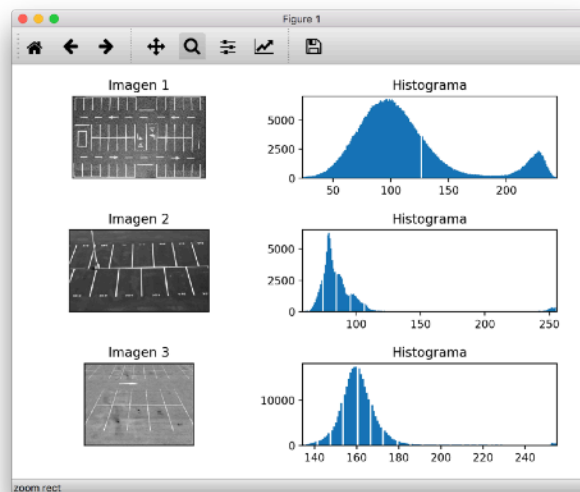


Figura 3.10. Histogramas de niveles de gris en aparcamientos vacíos

En la [figura 3.11](#) se muestra el resultado que se obtiene con la umbralización mediante Otsu de la imagen filtrada con el filtro de la mediana —[figura 3.8](#) (b)—, devolviendo un valor de umbral de 153. Este valor se utiliza en la siguientes etapas del algoritmo para delimitar los píxeles pertenecientes al fondo —suelo—, con un nivel de gris menor que 153; y los píxeles pertenecientes a posibles bordes —líneas de

aparcamiento—, con un nivel de gris mayor a 153. En la [figura 3.10](#) se observa en el histograma de la imagen 1 que el valor de 153 divide la gráfica en dos zonas: la de la izquierda, correspondiente a los píxeles más oscuros que representan el suelo; y la de la derecha, a los píxeles claros que se corresponden con las líneas.

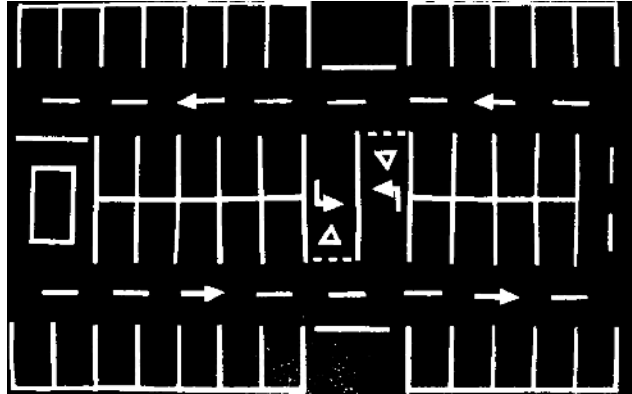


Figura 3.11. Imagen umbralizada mediante Otsu obteniendo un valor umbral de 153

Extracción de bordes

La extracción de bordes se utiliza para identificar los cambios bruscos de intensidad existentes entre píxeles adyacentes. Se corresponden con los contornos de las líneas de las plazas de aparcamiento.

“Los puntos de borde, o simplemente bordes, son píxeles alrededor de los cuales la imagen presenta una brusca variación en los niveles de gris. El objetivo consiste en dada una imagen, que puede o no estar corrompida por ruido, localizar los bordes más probables generados por elementos de la escena y no por ruido.” (Pajares y De la Cruz, 2007, p. 133).

En la [figura 3.12](#) se muestra el histograma con el perfil de intensidad de la iluminación correspondiente a la imagen de la [figura 3.2](#) (a) pasada a escala de grises. En histograma se observa que los bordes —cambios en la gráfica— se presentan como un cambio progresivo en el nivel de intensidad en lugar de presentar cambios bruscos.

Los métodos clásicos para la detección de bordes se basan en el cálculo de la primera o segunda derivada en los puntos locales teniendo en cuenta la intensidad vista en la [figura 3.12](#). La primera derivada es cero en todos los tramos en los que la intensidad se mantiene constante, y cambia a un valor constante en los tramos en los que se produce un cambio. La segunda derivada funciona al contrario, siendo cero en todos los tramos, excepto en el inicio y final de un cambio de intensidad.

Se dispone de distintos operadores con los que detectar bordes según se utilice la primera o segunda derivada. Así, entre los operadores de primera derivada encontramos: el de Sobel, el de Prewitt o el de Roberts, y entre los de segunda derivada los operadores Laplacianos.

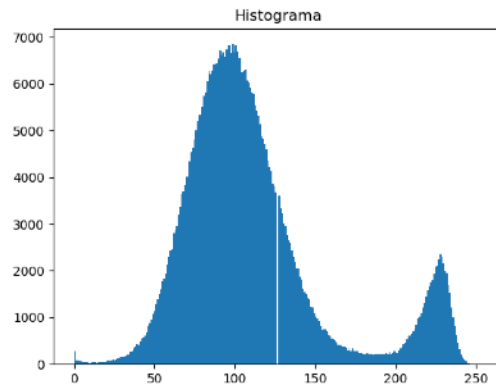


Figura 3.12. Histograma de intensidad de la imagen de la [figura 3.2](#) pasada a escala de grises

En el sistema se utiliza el algoritmo de Canny basado en la teoría de los operadores de la primera derivada. La razón principal de su utilización es que, además de detectar los bordes en la imagen, es capaz de detectar y diferenciar los contornos. Esta característica es perfecta para la detección de plazas de aparcamiento puesto que favorece la extracción de las mismas como contornos en la imagen.

A continuación, se analiza brevemente el algoritmo de Canny y se muestran los resultados de su utilización en el sistema.

Algoritmo de Canny

El algoritmo de Canny se utiliza para detección de bordes en una imagen y se basa en tres etapas principales:

1. Obtención del gradiente.

El gradiente es un vector en el que sus componentes miden la rapidez de cambio que se produce entre los valores de los píxeles, teniendo en cuenta la distancia y las direcciones x e y . (Penedo, 2009). Cada píxel de la imagen tiene asociado un gradiente y es un vector perpendicular al borde. Para obtener el gradiente: (i) se aplica un filtro gaussiano para suavizar la imagen, y (ii) en cada píxel de la ventana de convolución filtrada se calcula la magnitud —módulo— y dirección del gradiente en ese punto siguiendo un proceso similar al operador de *Sobel*.

2. *Supresión de píxeles que no pertenecen al borde o supresión no máxima.*

Con la supresión no máxima se busca adelgazar los bordes reduciendo al máximo los píxeles para obtener las líneas más finas posibles. Para ello, partiendo del *módulo* y *dirección* de gradiente calculado previamente, en cada píxel:

1. Se calcula la dirección d que mejor aproximación ofrece a la dirección del gradiente en ese punto que es perpendicular al borde.
2. Si el módulo del gradiente en el punto (x, y) es menor que uno de los píxeles contiguos en la dirección d se le asigna el valor 0, en caso contrario, se le asigna el valor de $módulo(x, y)$.

3. *Aplicación de umbral por histéresis a la imagen de la supresión no máxima.*

Puesto que la imagen resultante de la supresión no máxima puede contener ruido que genere falsos bordes, se aplica una umbralización por histéresis para reducir la probabilidad de aparición. Para ello, se fijan dos valores de umbral ($t1$ y $t2$), donde $t1$ es menor que $t2$, y se aplica la siguiente lógica en el gradiente (G) de cada punto:

1. Si el gradiente es mayor que el umbral superior ($G > t2$), entonces el píxel se considera un borde.
2. Si el gradiente es menor que el umbral inferior ($G > t1$), entonces el píxel se considera fondo y se descarta.
3. Si el gradiente se encuentra entre el umbral superior e inferior ($t1 < G < t2$), entonces el píxel se considera borde si tiene un píxel vecino que supere el umbral superior.

Como nota final, se recomienda la utilización de umbrales cuya relación sea de 2:1 o 3:1 entre el umbral superior e inferior. (Canny Edge Detector, 2014).

En el sistema se utiliza el algoritmo de Canny para obtener una segmentación de las líneas de las plazas de aparcamiento de manera que se definan mediante el contorno, eliminando el color interior de las líneas. Este adelgazamiento y contorneado hace que la definición de las coordenadas de las plazas sea más sencilla en fases posteriores del algoritmo.

En la [figura 3.13](#) se muestra el resultado que genera la aplicación del algoritmo de Canny en el sistema partiendo de la imagen previamente filtrada mediante el filtro de la mediana —[figura 3.8](#) (b)—. Los umbrales utilizados para la histéresis en el algoritmo se basan en el obtenido de manera automática mediante Otsu —[figura 3.11](#)—, cuyo valor es de 153. Siguiendo el principio de proporcionalidad recomendado para dichos umbrales, se aplica un proporción de 2:1 obteniendo un valor de umbral superior de 306 y de umbral inferior de 153.

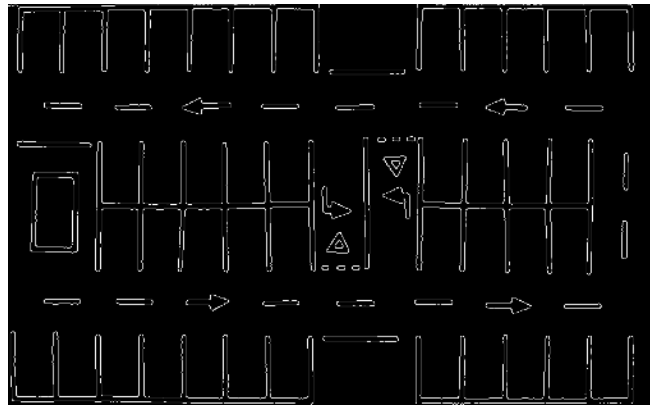


Figura 3.13. Extracción de bordes mediante el algoritmo de Canny

Una vez segmentada la imagen, eliminado el ruido y definidos los contornos de todas las líneas, se explican los métodos utilizados para la obtención final de las coordenadas de cada una de las plazas de aparcamiento.

Detección de contornos y aproximación a polígonos

La *detección de contornos* se utiliza en el sistema para obtener las coordenadas finales de las plazas de aparcamiento. Mediante este método, cada bloque de aparcamiento se identifica y diferencia como un contorno que es tratado de manera independiente con respecto al resto de bloques de aparcamiento y elementos de la imagen. En la [figura 3.14](#) (a) se muestran los diferentes contornos detectados en la imagen del aparcamiento.

“Los contornos pueden explicarse simplemente como una curva que une todos los puntos continuos (a lo largo del límite), con el mismo color o intensidad. Los contornos son una herramienta útil para el análisis de formas y la detección y reconocimiento de objetos.” (Mordvintsev & Abid, 2013)

OpenCV proporciona métodos tanto para la detección de contornos —*findContours*—, como para el dibujado de los mismos —*drawContours*—. En el sistema se utiliza el método de búsqueda de contornos que recibe:

1. La *imagen* resultante de la detección de bordes mediante el algoritmo de Canny.
2. El *método de aproximación al contorno* para determinar el número de puntos que se generan en la definición del mismo. Si pensamos en la detección del contorno de un línea recta, con el *flag CHAIN_APPROX_NONE* se genera un punto por cada píxel, con *CHAIN_APPROX_SIMPLE* únicamente se generan los puntos extremos que definen la recta.
3. El *modo de recuperación de contorno* que permite especificar si se quieren recuperar los contornos devolviendo la jerarquía de los mismos y el modo en el que se establece dicha jerarquía. Este parámetro es interesante si, por ejemplo, se quieren recuperar los objetos dentro de una plaza conociendo la relación jerárquica entre el contorno de la plaza continente —*padre*— y el objeto contenido —*hijo*—. Se proporcionan los siguientes modos de recuperación: (i) *RETR_LIST*, devuelve todos los contornos sin establecer ninguna relación jerárquica entre ellos; (ii) *RETR_EXTERNAL*, devuelve los contornos exteriores de cada grupo de contornos que mantengan una relación de jerarquía; (iii) *RETR_CCOMP*, devuelve los contornos jerarquizados utilizando únicamente 2 niveles (*padre-hijo*); y (iv) *RETR_TREE*, devuelve la jerarquía de contornos completa sin límite de niveles de profundidad.

Para la detección de las plazas de aparcamiento, al tratarse de líneas rectas, se utiliza el método de aproximación *CHAIN_APPROX_SIMPLE* reduciendo la necesidad de memoria al generar menos puntos por contorno, y el modo de recuperación *RETR_LIST* para obtener todos los contornos ni necesidad de jerarquía puesto que todos están en el mismo nivel. En la [figura 3.14](#) (a) se muestra el resultado de dibujar los contornos obtenidos mediante la búsqueda anterior en el aparcamiento.

Si se analiza más en detalle la [figura 3.14](#) (a) se observa que, aún utilizando el método de aproximación de contornos que genera el menor número de puntos posibles, las líneas no son completamente rectas y presentan pequeñas variaciones que hacen que cada línea esté definida internamente por multitud de puntos. En la [figura 3.14](#) (b) se muestran todos los puntos resultantes que definen los contornos. Por ello, para reducir el número de puntos y buscar la representación más simple del aparcamiento para definir las coordenadas de las plazas, se utiliza la aproximación a polígonos.

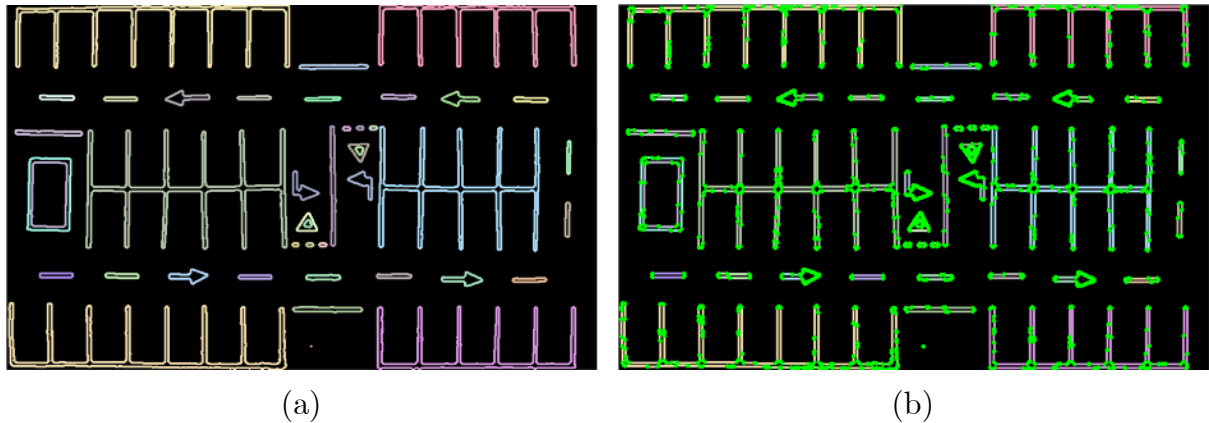


Figura 3.14. Líneas del aparcamiento detectadas mediante la búsqueda de contornos. (a) Imagen con contornos detectados (cada uno con un color); (b) Imagen con contornos detectados con los puntos (en verde) que definen cada contorno

OpenCV proporciona un método de aproximación a polígonos —*approxPolyDP*— con el que se pueden reducir el número de puntos finales. El método se basa en la utilización del *algoritmo de Ramer–Douglas–Peucker*.

“El propósito del algoritmo de *Ramer–Douglas–Peucker* es, dada una curva compuesta de segmentos de línea (también llamada polilínea en algunos contextos), encontrar una curva similar con menos puntos. El algoritmo define una "diferencia" basada en la distancia máxima entre la curva original y la curva simplificada (es decir, la distancia Hausdorff entre las curvas). La curva simplificada consiste en un subconjunto de los puntos que definieron la curva original.” (Ramer–Douglas–Peucker algorithm, 2019).

El algoritmo utiliza un parámetro (ϵ) que determina la precisión de la curva aproximada, definiendo así la distancia máxima válida entre la curva original y su aproximación. El algoritmo es un proceso recursivo cuyo funcionamiento es el siguiente (Ramer–Douglas–Peucker algorithm, 2019):

1. Se toma como solución inicial el segmento resultante de la unión de los puntos extremo del contorno y se busca el punto más alejado del segmento.
 1. Si la distancia del punto más alejado es menor que ϵ el resto de puntos más cercanos son descartados.

2. Si la distancia del punto más alejado es mayor que ϵ el punto se mantiene en la aproximación final y se busca de manera recursiva hasta obtener el mismo resultado de (a).

Con el resultado del algoritmo se pueden generar los mismos contornos de la [figura 3.14](#) (a) utilizando menos punto para su definición siendo esto lo que se busca.

Para calcular el valor de ϵ se utiliza el perímetro de cada uno de los contornos analizados —[figura 3.14](#) (a)— correspondientes a bloques de aparcamiento. Se ha comprobado empíricamente que el valor óptimo de ϵ es igual al 0.5% del perímetro del contorno original. Con este valor se consigue el menor número de puntos sin eliminar los que forman parte de las coordenadas finales de las plazas de aparcamiento.

En la [figura 3.15](#) se muestra el resultado final de aplicar el *algoritmo de Ramer–Douglas–Peucker*. En la imagen (a) se muestran los contornos obtenidos mediante la *detección de contornos* junto con los puntos detectados para su definición (verde), y los puntos resultantes de la aproximación por polígonos (rojo). En la imagen (b) se muestran los contornos dibujados en base a los puntos que se obtienen mediante la aplicación del algoritmo de *Ramer–Douglas–Peucker*. Se observa que se reducen los puntos preservando los necesarios para definir las estructuras de los contornos.

No obstante, para la detección de las plazas de aparcamiento todavía existen puntos redundante en las zonas de cruce entre líneas verticales y horizontales en los bloques de aparcamiento. En la [figura 3.15](#) (b) se aprecia este problema obteniendo en algunos cruces hasta 4 puntos diferentes en la separación central de un bloque. Para reducir al mínimo el número de puntos, se procesan todos los que forman un contorno (bloque de aparcamiento) agrupando los puntos en los que su distancia euclídea (Euclidean distance, 2019) es menor que un valor umbral, y calculando el punto medio del grupo generando así una única coordenada en los cruces de líneas verticales y horizontales —[figura 3.16](#) (a)—. Los extremos de las líneas verticales de los bloques de aparcamiento se reducen correctamente utilizando un punto único en cada extremo.

Siguiendo con la [figura 3.15](#), se observa que se detectan tanto las líneas de las plazas de aparcamiento como el resto de marcas viales. Esta casuística queda cubierta en el sistema al procesar cada uno de los bloques de aparcamiento por separado, seleccionando el RoI de las plazas a segmentar, dejando fuera las marcas externas al bloque. No obstante, para paliar un posible error de selección, se implementa una solución que tiene en cuenta el área mínima que deben de tener los contornos

detectados para ser considerados bloques de aparcamiento u objetos de interés. En la [figura 3.16](#) se observa el resultado de la aplicación de esta solución con los contornos dibujados sobre fondo negro con los puntos calculados en la reducción a los mínimos puntos necesarios (a) y sobre la imagen del aparcamiento original (b).

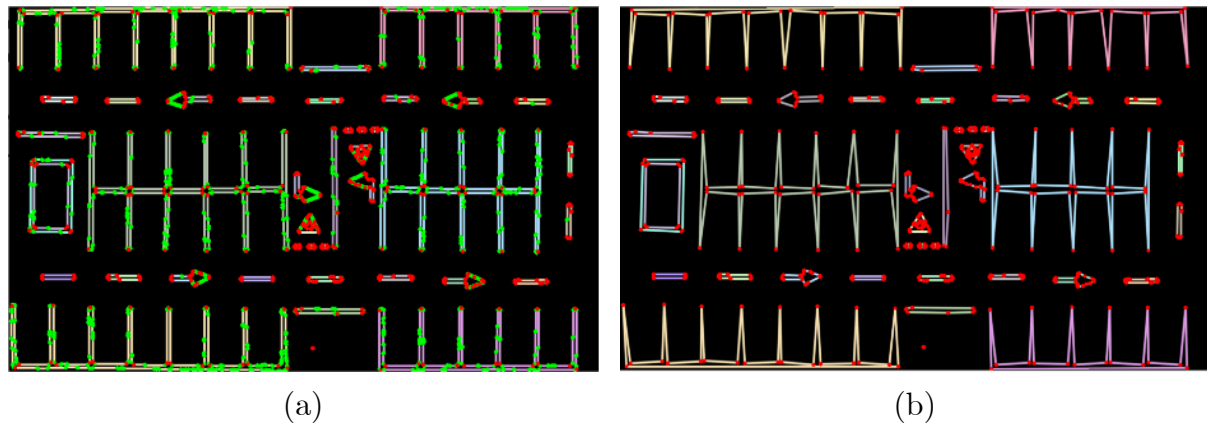


Figura 3.15. (a) Imagen con puntos resultantes de la detección de contornos (verde) y puntos resultantes de la aproximación a polígonos (rojo); (b) Imagen con contornos dibujados con los puntos resultantes de la aproximación a polígonos (rojo)

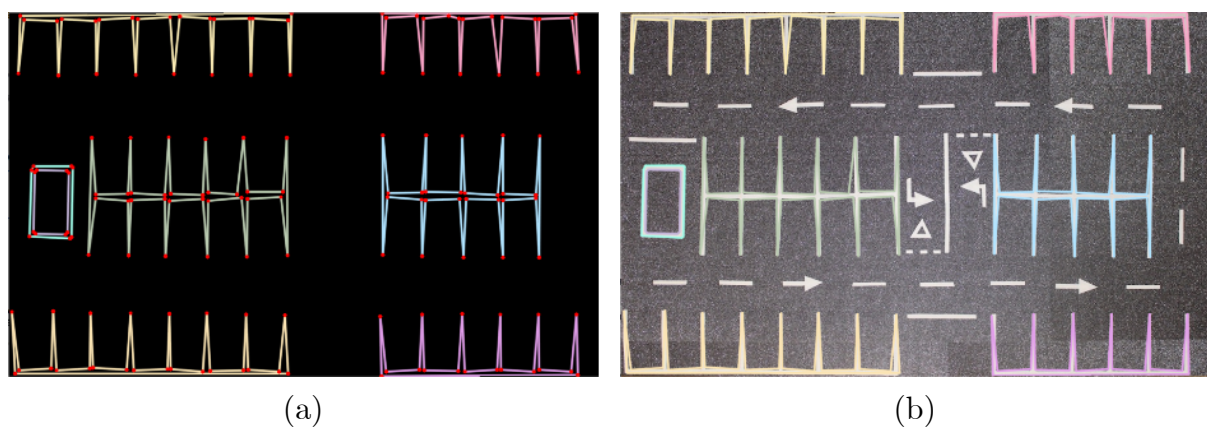


Figura 3.16. Contornos obtenidos al establecer el área mínima de un contorno para ser considerado objeto de interés. (a) Imagen con contornos aislados por área y puntos cercanos agrupados en un central (verde); (b) Imagen con contornos sobre aparcamiento original

En la [figura 3.17](#) se muestra el resultado final con el dibujado de las entidades detectadas como plazas en el sistema.

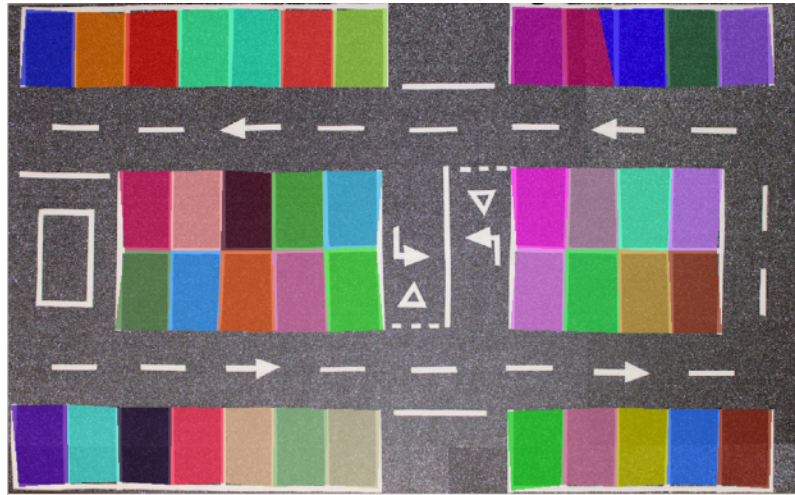


Figura 3.17. Plazas representadas en el sistema como entidades definidas por coordenadas

Ajuste manual

Como se observa en la [figura 3.17](#), en el modelo computacional obtenido algunas plazas de aparcamiento se definen de manera incorrecta. Por ejemplo, en la fila superior las plaza 9 y 10 se han detectado incorrectamente por obtener una coordenada errónea.

Puesto que la detección de las plazas de aparcamiento es un punto crítico ya que una mala definición haría que la detección de la ocupación no fuese correcta, se propone una última etapa, únicamente necesaria en la fase de configuración del sistema, mediante la que se hace una comprobación manual de las coordenadas de las plazas de aparcamiento obtenidas. En caso de existir una detección incorrecta en algún bloque de plazas, el usuario encargado de la configuración del sistema puede corregir las coordenadas sobre la imagen en una interfaz gráfica habilitada.

Generación del modelo computacional de las plazas

El resultado del proceso de detección automático descrito y la comprobación y corrección manual, en caso de ser necesaria, generan en el sistema las coordenadas de todas las plazas de aparcamiento. En este paso final se agrupan las coordenadas y se generan los objetos correspondientes a cada plaza con los que se trabaja en el sistema.

3.3.2. Detección de vehículos en el aparcamiento

Una tarea tan simple para un humano como identificar los vehículos en una imagen es un proceso muy complicado para una máquina. Las máquinas *ven* una imagen como un conjunto de bits. Abstrayendo al más alto nivel se puede decir que son capaces de *ver* la imagen como una matriz en la que cada elemento representa un color, pero no de decidir dónde existe un vehículo o el número de ellos de manera simple.

Selección de tecnología para la detección

Para dotar al sistema de esta capacidad de identificación y categorización de los vehículos en una imagen, obviando el resto de objetos que no sean de interés —como pueden ser peatones o carros de la compra—, se plantea el uso de técnicas de *aprendizaje automático*. Concretamente, se hace uso de *redes neuronales* para dotar al sistema de inteligencia haciéndolo capaz de reconocer patrones. Siendo las *redes neuronales convolucionales* (Redes neuronales convolucionales, 2019) las que aportan una percepción semejante a la de la corteza visual.

Las redes neuronales proporcionan un conjunto de algoritmos potentes que permiten al sistema resolver problemas triviales para un humano, pero con una complejidad muy elevada para un computador, aportando al sistema capacidad de percepción sensorial.

El enfoque planteado en este trabajo necesita: (i) detectar y diferenciar los vehículos dentro de una imagen, y (ii) calcular el área que ocupa cada vehículo con un método con un buen grado de precisión. Teniendo en cuenta estos requisitos se analizan las alternativas existentes para encontrar la mejor opción a implementar en el sistema.

Selección de método de detección

Los métodos de extracción de información de imágenes parten de un conjunto de categorías de objetos a detectar en la imagen analizada. Estos métodos se pueden clasificar en cuatro categorías según la información que son capaces de extraer de la imagen y el modo en que lo hacen. En la [figura 3.18](#) se ilustra las diferencias entre los métodos a continuación descritos.

- La *clasificación de imágenes* predice para una imagen una única categoría correspondiente al objeto predominante. En la [figura 3.18](#) se observa que se detectan los coches como categoría principal sin hacer diferenciación entre ellos.

- La *detección de objetos* categoriza y localiza la posición de múltiples objetos dentro de una imagen con un cuadro delimitador. Este método se diferencia de la *clasificación de imágenes* en que, en lugar de categorizar un único objeto, añade tantas cajas delimitadoras como objetos se reconozcan en la imagen. En la [figura 3.18](#) se observa que se detectan, localizan y separan cada uno de los coches que aparecen
- La *segmentación semántica* identifica la categoría de cada píxel de una imagen lo que permite categorizar los diferentes tipos de objetos, pero sin diferenciar entre objetos de la misma clase. En la [figura 3.18](#) se observa que se predicen dos categorías —coche y asfalto— sin diferenciar cada coche como una entidad independiente.
- La *segmentación de instancias* identifica la categoría de cada píxel de la imagen diferenciando entre píxeles de la misma categoría pertenecientes a diferentes objetos de la misma clase. Emplea técnicas de detección de objetos y segmentación semántica. Predice la localización y clase de los objetos reconocidos en una imagen —como hace la detección de objetos—, pero no dibuja una caja delimitadora alrededor de cada uno. En su lugar, predice la máscara de esos objetos —como hace la segmentación semántica—. En la [figura 3.18](#) se observa que se predicen dos categorías —coche y asfalto— y se diferencia cada uno de los coches con una máscara.

Por lo tanto, la mejor técnica a emplear es la ***segmentación de instancias*** puesto que el sistema necesita: (i) detectar todos los vehículos existentes dentro de una imagen y (ii) conocer el área de cada uno de los vehículos. Es cierto que la *segmentación de instancias* es computacionalmente más costosa que la *detección de objetos*. No obstante, en caso de aplicar detección de objetos, sería necesario un análisis posterior del cuadro del objeto detectado para obtener los píxeles del vehículo, elevando el coste y complejidad computacional.

Especificación del algoritmo y redes neuronales empleadas

Las *redes neuronales convolucionales basadas en regiones* (R-RNC) son un tipo de RNC enfocadas a la detección de objetos. Las R-RNC dividen la imagen en un conjunto de regiones propuestos en base a una búsqueda selectiva, y luego procesan cada una de las regiones en una RNC para predecir la categoría de los objetos y delimitarlos con una caja.

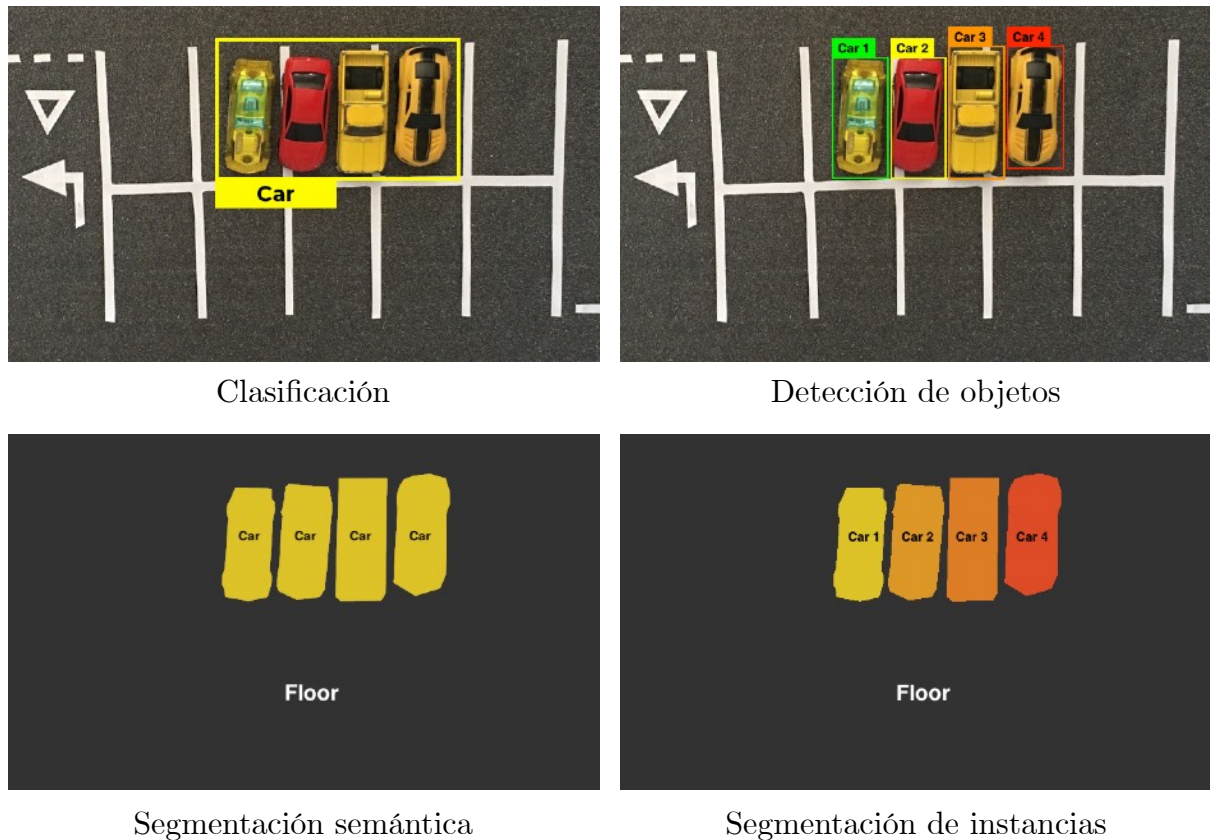


Figura 3.18. Métodos de extracción de objetos de imágenes

En (Girshick et al., 2013) se encuentra la primera aproximación a este tipo de algoritmos capaces de detectar objetos, pero no de *segmentar instancias*. Sobre el algoritmo original se han hecho mejoras. Girshick plantea una mejora en *Fast R-RNN* (Girshick, 2015) haciendo el algoritmo original —*R-CNN*— más rápido al procesar todas las regiones propuestas juntas en su RNC con un *pool* de regiones de interés. Run et al. plantean *Faster R-RNN* (Ren et al., 2015) mejorando el algoritmo anterior con la utilización de una RNC para seleccionar las propuestas de regiones —red de propuestas de región—. En la última versión del algoritmo, He et al. proponen *Mask R-RNN* (He et al., 2017) una mejora con respecto a *Faster R-RNN* que incluye la *segmentación de instancias* en la imagen.

Por tanto, en base a los requisitos del sistema y el análisis realizado se concluye que el algoritmo de máscara R-RNC —*mask R-RNN*— basado en R-RNC es una buena opción a utilizar en este trabajo.

“La máscara R-RNC (red neuronal convolucional regional) es un marco de dos etapas: la primera etapa escanea la imagen y genera propuestas (áreas que pueden contener un objeto). Y la segunda etapa clasifica las propuestas y genera cuadros delimitadores y máscaras.” (Abdulla, 2018)

Para la inclusión del algoritmo en el sistema se hace uso de la librería de código abierto *Mask_RCNN* (Mask_RCNN, 2018). La librería se basa en los siguientes conceptos detallados en profundidad en (Abdulla, 2018).

- Una red neuronal troncal basada en las arquitecturas *ResNet101* y red piramidal de característica (*Feature Pyramid Network, FPN*). *ResNet101* se utiliza como extractor de características con varias capas. Las primeras capas detectan características de bajo nivel —como bordes—, y las siguientes, características de alto nivel definiendo la categoría del objeto. Como mejora sobre la arquitectura *ResNet101* se incluye *FPN* añadiendo una segunda pirámide de capas que utiliza las características de alto nivel de las capas de *ResNet101* para pasarlas a las capas inferiores. Con esto se permite que las funciones de todos los niveles tengan acceso a funciones de nivel inferior y superior.
- Una red neuronal liviana para determinar las regiones propuestas —red de regiones propuestas (RRP)— mediante el escaneo de la imagen con la utilización de una ventana deslizante para encontrar áreas con objetos.
- Una etapa de análisis de las regiones de interés (RoI) propuestas por la RRP que genera dos salidas para cada RoI: (i) la clase del objeto dentro de la región en caso de ser detectado y (ii) un ajuste de la caja delimitadora del objeto para refinar la encapsulación.
- Una etapa final que genera las máscaras de segmentación de cada objeto detectado en base a las RoI de la etapa anterior.

Entrenamiento de la red neuronal

La fase entrenamiento de la red neuronal es el proceso en el que el sistema aprende a detectar y segmentar los vehículos, separándolos del fondo y del resto de objetos. Para entrenar un modelo de aprendizaje automático desde cero y obtener una tasa de predicción mínimamente aceptable, se deben utilizar conjuntos de datos de al menos 300 imágenes de cada objeto a predecir.

En este trabajo, en lugar de utilizar un conjunto de datos enorme, partimos de un modelo previamente entrenado sobre el conjunto de datos de COCO (COCO, 2018). Este modelo es capaz de reconocer vehículos capturados con un plano normal —ya sean vistos de frente o de lateral—, pero no es capaz de reconocer vehículos capturados desde un plano cenital. Lo que aporta la utilización de un modelo

previamente entrenado es que tiene aprendidas muchas características comunes de imágenes del mundo real.

Por lo tanto, partiendo del modelo entrenado en COCO, se genera un nuevo conjunto de 66 imágenes de vehículos tomadas desde un plano cenital para reentrenar la red. El conjunto se divide en dos grupos: 48 imágenes para el entrenamiento y 18 imágenes para la validación. El conjunto de entrenamiento está compuesto por imágenes con distintos tipos de vehículos, colocados en varias posiciones, sobre distintas superficies y con diferentes condiciones de iluminación. Con esto se consigue un buen entrenamiento de la red para resolver problemas relacionados con cambios lumínicos, sombras, diferentes colores de superficie en el aparcamiento, etc. En la [figura 3.19](#) se muestra un ejemplo de imágenes utilizadas. En (a) se colocan varios vehículos en distintas posiciones del aparcamiento. En (b) se utiliza un vehículo con un ángulo y perspectiva distinto y sobre una superficie diferente.

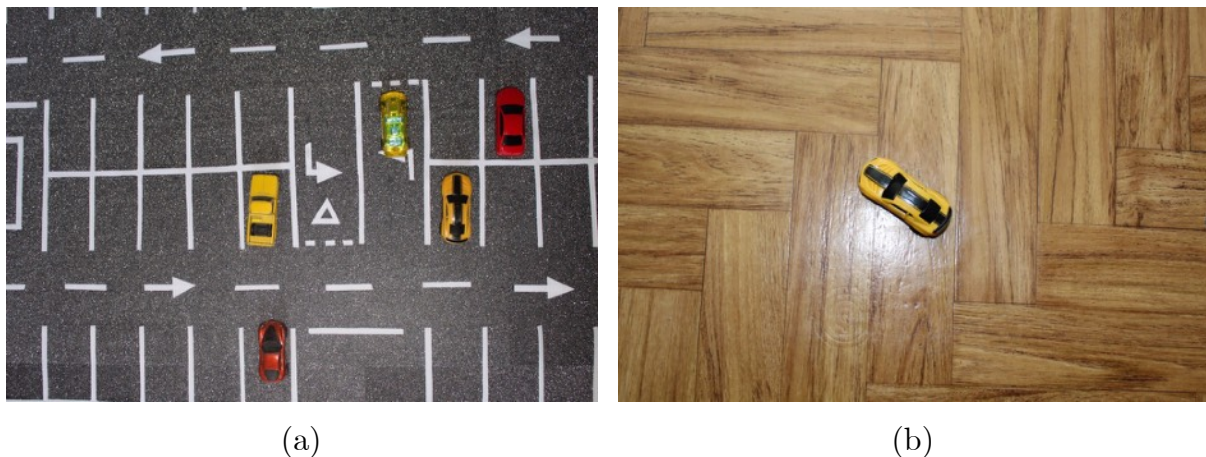


Figura 3.19. Ejemplo de imágenes utilizadas para el entrenamiento de la red neuronal. (a) Imagen del aparcamiento con varios vehículos; (b) Imagen con un vehículo sobre un fondo diferente

Una vez generado el conjunto de imágenes, se anotan y clasifican en cada una de ellas los objetos de interés. En el sistema propuesto se anotan en cada imagen los vehículos que aparecen. En la [figura 3.20](#) se muestra el resultado de la anotación de las imagen de la [figura 3.19](#) (a).

Prueba de la red neuronal

Una vez entrenado el modelo se utiliza un conjunto de imágenes de prueba para validar la efectividad de la detección de vehículos en el sistema. En la [figura 3.21](#) se muestra la detección de vehículos. En (a) se muestra la imagen original. En (b) a la

imagen original se le aplica un efecto de negativo a los colores, dejando las zonas en las que se han detectados vehículos con sus colores originales.

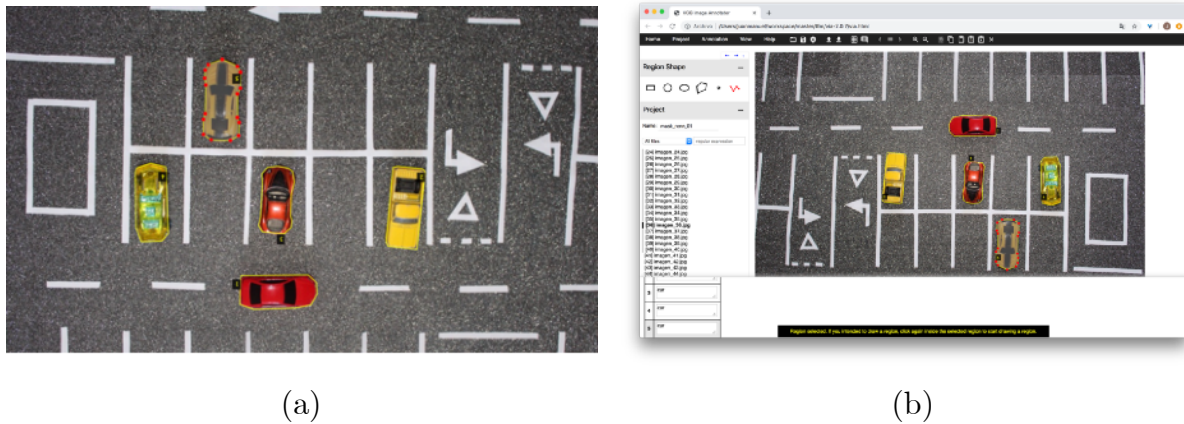


Figura 3.20. Anotaciones de vehículos presentes en la imagen de la [figura 3.19](#)

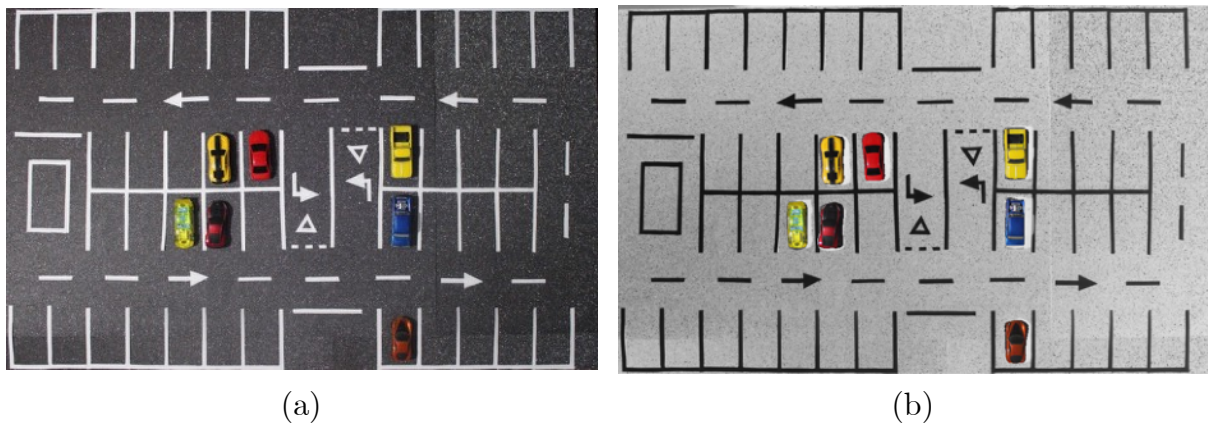


Figura 3.21. Detección de vehículos en el aparcamiento. (a) Imagen original; (b) Imagen en negativo con máscaras de vehículos detectados en color original

3.3.3. Medición sobre la escena

La medición sobre la escena es el proceso mediante el cual el sistema es capaz de conocer el tamaño de los diferentes objetos que coexisten en el aparcamientos. Estas mediciones son necesarias para obtener: (i) el área de las plazas de aparcamiento, (ii) el área de los vehículos estacionados y (iii) el área de los vehículos que acceden al aparcamiento. Gracias a (i) y (ii) el sistema es capaz de determinar el área de ocupación de cada plaza de aparcamiento, y mediante (iii) el sistema obtiene la información necesaria para posteriormente determinar la plaza óptima para el vehículo entrante. En la [figura 3.22](#) se muestra el *área de detección de vehículo* en la que se obtiene el tamaño de los vehículos entrantes al aparcamiento para su posterior ubicación.

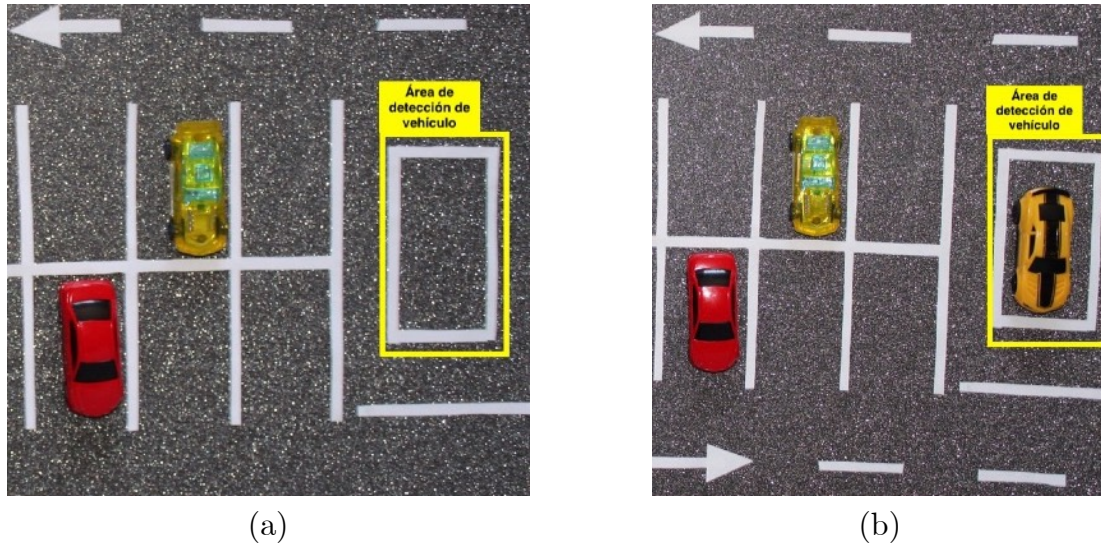


Figura 3.22. Área destinada a la detección del vehículo entrante

Para hacer correctamente las mediciones, en la configuración del sistema se determina un factor multiplicador con el que pasar de píxeles a metros. Este paso es necesario al tratarse de un sistema escalable que utiliza múltiples cámaras para monitorizar el aparcamiento. Así se consigue eliminar la necesidad de colocar todas las cámaras a la misma altura utilizando para cada una la escala necesaria mediante un *factor multiplicador* —relación píxel/metro—, obteniendo medidas genéricas y uniformes en el sistema general. Estas medidas genéricas son las que permiten, por ejemplo, determinar si el vehículo entrante cabe en una plaza o, por el contrario, es demasiado pequeña para dicho vehículo.

Estas mediciones y la determinación del estado de una plaza en base al área ocupada son las razones por las que se plantea la utilización del *plano cenital*. La utilización del plano cenital en el sistema permite: (i) reducir al máximo las distorsiones causadas por la perspectiva, (ii) la eliminación de las oclusiones producidas entre vehículos y (iii) plantear el aparcamiento como un modelo en 2 dimensiones.

“Plano cenital es aquel en el que el punto de vista de una cámara se encuentra perpendicular respecto del suelo y la imagen obtenida ofrece un campo de visión orientado de arriba abajo.” (Plano cenital, 2019)

3.3.4. Detección de movimiento

Un sistema dedicado a la detección de ocupación en un aparcamiento en el que se establezcan *intervalos constantes* para la actualización del estado de las plazas, no es razonable desde el punto de vista de la eficiencia. De la misma manera que no es óptimo establecer un intervalo de actualización del sistema demasiado rápido en

condiciones de poca afluencia de vehículos, tampoco es óptimo establecer intervalos demasiados espaciados en condiciones de elevado tráfico. Presentando, esta segunda situación, un problema de consistencia entre la información del aparcamiento que tiene el sistema y la situación real al no actualizar el estado cuando es necesario.

Por ello, en el sistema se integra una funcionalidad dedicada a la detección del movimiento del aparcamiento. En base a la afluencia de tráfico y la detección del movimiento se aumenta o disminuye el intervalo de actualización obteniendo un *sistema adaptativo* a la condiciones cambiantes.

Por lo tanto, la actualización del sistema se hace bajo diferentes situaciones: (i) cuando se detecte movimiento dentro del aparcamiento, (ii) en el momento que entra o sale un vehículo del recinto y (iii) cuando expira un *intervalo adaptativo* de tiempo en función de la afluencia de tráfico. La situación (iii) es un caso especial enfocado a paliar posibles errores de detección de movimiento puesto que, en los supuestos (i) y (ii), influyen muchas situaciones externas al sistema. Con estas tres situaciones se asegura la consistencia del sistema, como mínimo, cada *intervalo adaptativo* de tiempo.

OpenCV proporciona los siguientes tres algoritmos para la sustracción del fondo (Background subtraction, 2017):

- BackgroundSubtractorMOG es un algoritmo de segmentación de fondo y primer plano basado en la mezcla *gaussiana* introducido en (KaewTraKulPong & Bowden, 2001).
- BackgroundSubtractorMOG2 es el otro algoritmo de segmentación de fondo y primer plano basado en la mezcla *gaussiana* descrito en (Zivkovic, 2004) y (Zivkovic & Heijden, 2006). Proporciona una mejor adaptación a condiciones lumínicas cambiantes.
- BackgroundSubtractorGMG es un algoritmo que combina la estimación estadística de la imagen de fondo y la segmentación bayesiana por píxel introducido en (Godbehare et al., 2012). Emplea algoritmos basados en inferencia bayesiana para la identificación de objetos y segmentación del primer plano. Para proporcionar mejor adaptación a los cambios de iluminación los fotogramas más recientes tienen más peso que los antiguos.

Los algoritmos que se han enumerado son métodos potentes y precisos para la detección del movimiento, pero son computacionalmente costosos. Puesto que no es necesaria una precisión óptima en este aspecto, se hace uso de un método

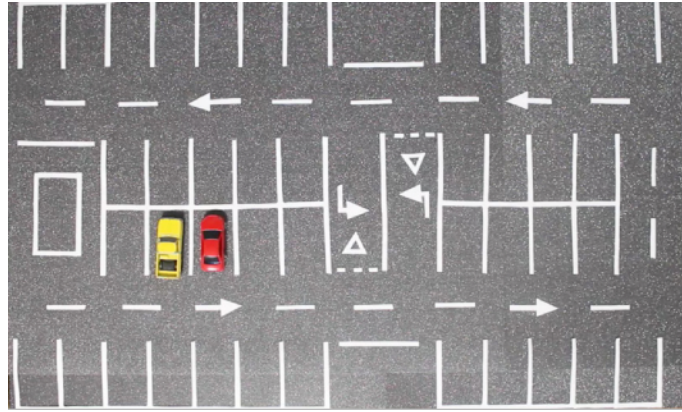
computacionalmente más simple, pero efectivo para la finalidad que nos ocupa. Además, como se detalla en el apartado de arquitectura (3.5), el software se ejecuta en dispositivos hardware simples para favorecer la escalabilidad del sistema. Por lo tanto, cuanto más simple y ajustado a la finalidad deseada sea el software, mejor.

El detector de movimiento implementado en el sistema se basa en la técnica de la sustracción de fondo para la segmentar las dos partes de la imagen —fondo y primer plano—. Para ello se comparan imágenes consecutivas que ocurren en los instantes temporales $t1$ y $t2$ siendo $t2 > t1$. La imagen que ocurre en $t1$ se utiliza como fondo por ser anterior a $t2$. El análisis en el instante $t2$ se hace sustrayendo a la imagen de $t2$ la de $t1$. Los objetos en movimiento aparecen en el resultado de la resta como primer plano, mientras que las partes estáticas se eliminan asociándose como parte del fondo.

Realmente, el sistema sustrae a $t2$ una imagen generada en base a una media ponderada de los fotogramas previos a $t2$. Con esto se consigue eliminar falsos positivos que pueden ser ocasionados, por ejemplo, por las diferentes condiciones de luz que se dan a lo largo del día.

Puesto que en un aparcamiento coexisten vehículos con otros objetos como bicicletas o incluso peatones, se establece un área mínima que deben tener los objetos del primer plano para ser considerados *objetos de interés*. En esta funcionalidad no se utiliza la red neuronal para la detección de los vehículos para reducir la complejidad computacional.

Con todo esto, se obtiene un método *simple y efectivo* computacionalmente capaz de adaptarse a las condiciones cambiantes evitando falsos positivos. En la [figura 3.23](#) se muestra el resultado de la detección de movimiento. Se parte de un estado inicial (a) dado en el instante $t0$, en el instante $t1$ (b) se detecta un objeto de interés en el área de detección, en los instantes $t2$ (c) y $t3$ (d) se detecta el movimiento de ese mismo objeto. En las imágenes (b), (c), (d) se muestra, en la fila superior, el objeto detectado enmarcado; y, en la inferior, la sustracción de fondo entre imágenes consecutivas obteniendo así los objetos en movimiento.



(a)

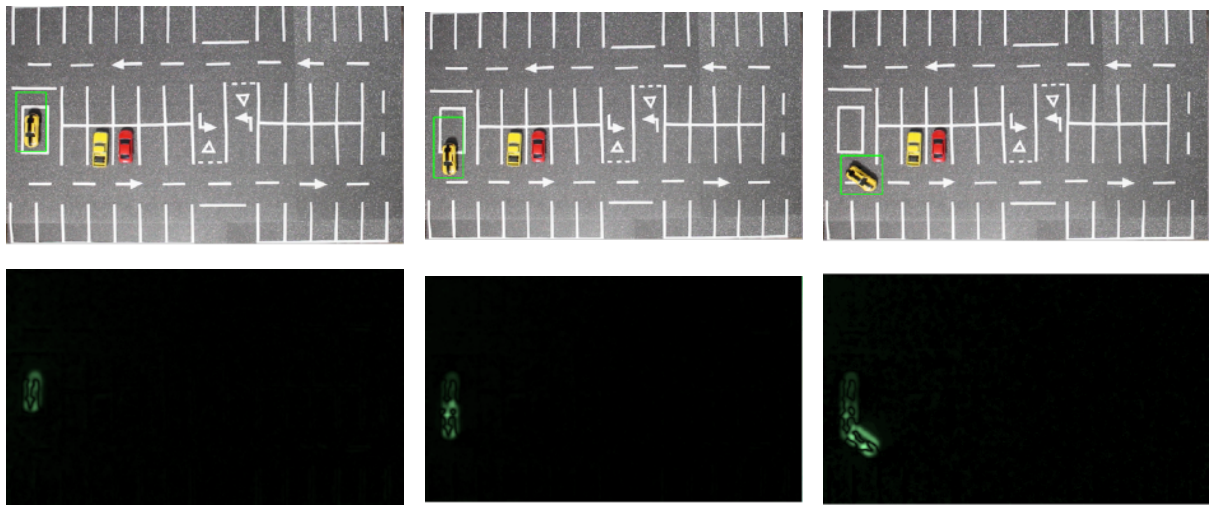


Figura 3.23. Imagen original. (a) (b) y (c) Objetos (arriba) y máscaras (abajo) detectadas

3.4. Diseño

En este apartado se detalla el diseño planteado para el sistema propuesto definiendo las clases y el software que se emplea en el desarrollo del trabajo.

3.4.1. Clases

Las *clases* o *modelos* son los objetos que representan las entidades que existen en el sistema. En la [figura 3.24](#) se muestra el diagrama de las clases utilizadas para modelar el sistema y se especifican a continuación.

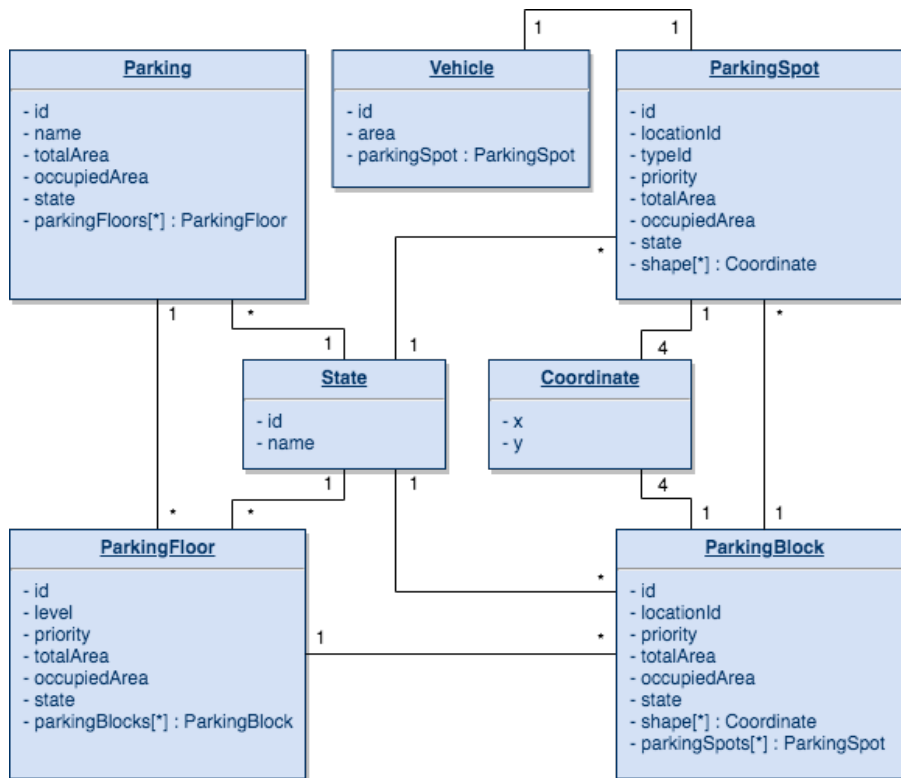


Figura 3.24. Diagrama de clases del sistema

Aparcamiento

El *aparcamiento* (*Parking*) define un espacio de estacionamiento específico. Todos los objetos existentes en el sistema deben pertenecer a una entidad de este tipo puesto que estarán ubicados en uno de ellos. Cada aparcamiento contiene al menos una *planta de aparcamiento*. Los atributos que contiene un aparcamiento son:

- Un identificador.
- Un nombre de aparcamiento.
- El área total destinada a plazas de aparcamiento determinada en base a las áreas de las plantas de aparcamiento.
- El área ocupada en las plazas de aparcamiento determinada en base a las áreas ocupadas de las plantas de aparcamiento.
- Un conjunto de plantas de aparcamiento.
- El estado del aparcamiento.

Planta de aparcamiento

Las *plantas de aparcamiento* (*ParkingFloor*) definen los niveles que pueden existir dentro de un aparcamiento. Las plantas de aparcamiento pertenecen a una entidad de aparcamiento y contienen un conjunto de bloques de aparcamiento. Los atributos que contiene una planta de aparcamiento son:

- Un identificador.
- El nivel que ocupa en el aparcamiento.
- La prioridad de la planta para determinar el orden de llenado preferente para la búsqueda de huecos libres.
- El área total de la planta de aparcamiento determinada en base a las áreas de los bloques del nivel.
- El área ocupada de la planta de aparcamiento determinada en base a las áreas ocupadas en los bloques del nivel.
- El estado de la planta de aparcamiento.
- El conjunto de bloques de aparcamiento.

Bloques de aparcamiento

Los *bloques de aparcamiento* (*ParkingBlock*) definen grupos de plazas contiguas. Los bloques de aparcamiento pertenecen a una planta de aparcamiento y contienen un conjunto de plazas de aparcamiento. Los atributos que contiene un bloque de aparcamiento son:

- Un identificador.
- Un identificador asociado a la ubicación real del bloque en el aparcamiento. Por ejemplo, existen aparcamientos que utilizan colores, letras o números para identificar cada zona.
- Las coordenadas del polígono que cubre todas las plazas que contiene el bloque.

- El área total del bloque de aparcamiento determinada en base a las áreas de las plazas del bloque.
- El área ocupada del bloque de aparcamiento determinada en base a las áreas ocupadas de las plazas del bloque.
- El estado del bloque de aparcamiento.
- El conjunto de plazas de aparcamiento.

Plazas de aparcamiento

Las *plazas de aparcamiento* (*ParkingSpot*) son la entidad mínima existente en la jerarquía del aparcamiento. Las plazas pertenecen a un bloque de aparcamiento. Los atributos que contiene una plaza de aparcamiento son:

- Un identificador.
- Una identificar asociado a la ubicación real de la plaza en el aparcamiento. Por ejemplo, existen aparcamientos que utilizan letras o números para identificar las plazas.
- Las coordenadas del polígono que define la plaza.
- El área total de la plaza.
- El área ocupada de la plaza determinada en base a la existencia o no de un vehículo estacionado.
- El estado de la plaza de aparcamiento.

Estados

Los estados (*State*) en las entidades de aparcamiento descritas definen la tasa de ocupación mediante un identificador. Todas las entidades tienen en común los siguientes estados en función de su área de ocupación:

- Vacío (*empty*) si el área de ocupación es inferior al 10%
- Medio vacío (*half_empty*) si el área de ocupación está entre el 10% y el 50%.
- Medio lleno (*half_full*) si el área de ocupación está entre el 50% y el 90%.

- Lleno (*full*) si el área de ocupación es superior al 90%.

Las plazas de aparcamiento disponen de dos estados adicionales: reservado (“*reserved*”) y desactivado (“*disabled*”). Definidos para situaciones en los que se pueda, por ejemplo, reservar una plaza a través de una aplicación o la plaza esté siendo restaurada. En el sistema son tratados como lleno.

Coordenadas

Las *coordenadas* (*Coordinate*) contienen dos atributos que hacen referencia a un par de puntos — x e y —. Son utilizadas para definir los vértices de los polígonos que forman las entidades del sistema.

Vehículo

Los vehículos estacionados en cada uno de los aparcamientos se representan con la clase *vehículo* (*Car*). A cada vehículo se le asigna una plaza de aparcamiento en el momento que entra en el parking. Los atributos que contiene un vehículo son:

- Un identificador.
- Un área correspondiente al área del polígono que define el vehículo.
- La plaza de aparcamiento asignada.

3.4.2. Software

Para dar soporte a los requisitos del sistema y funcionalidades descritas en el [apartado 3.3](#) se detallan los requisitos software planteados en el diseño y utilizados para la implementación. Se definen los siguientes puntos: lenguaje de programación, librerías y sistema de base de datos.

3.4.2.1. Lenguaje de programación

Como *lenguaje de programación* se decide utilizar *Python* —versión 3.5— puesto que es considerado uno de los lenguajes por excelencia para sistemas orientados a inteligencia artificial. En (DewangNautiyal, 2019) lo colocan el primero en el ranking, por delante de R y Java, gracias a su simplicidad y versatilidad afirmando que:

“Python está considerado en primer lugar en la lista de todos los lenguajes de desarrollo de la IA debido a su simplicidad. Las sintaxis de python son

muy simples y se pueden aprender fácilmente. Por lo tanto, muchos algoritmos de IA pueden ser fácilmente implementados en él. Python necesita poco tiempo de desarrollo en comparación con otros lenguajes como Java, C++ o Ruby. Python soporta estilos de programación orientados a objetos, funcionales y orientados a procedimientos. Hay muchas librerías en python, lo que hace que nuestras tareas sean más fáciles.”

Otros artículos que buscan el mejor lenguaje de programación en el área de nuestro trabajo posicionan también a *Python* en la primera posición —(Fabisiak, 2019) y (Zola, 2019)—.

3.4.2.2. Sistema de base datos

MongoDB es el sistema de base de datos seleccionado para persistir los datos del sistema. *MongoDB* es una base de datos NoSQL orientada a documentos. Persiste los datos en documentos con formato BSON —representación binaria de JSON (*JavaScript Object Notation*, “notación de objeto de JavaScript”)—.

Una base de datos *NoSQL*, a diferencia de las bases de datos relacionales, nos permite un esquema flexible y dinámico en la estructura de los datos. Esto permite disponer de documentos con estructuras diferentes dentro de la misma colección. Además, *MongoDB* está pensado y diseñado para ser escalable proporcionando *replicación* y *fragmentación* de manera muy sencilla.

Estos dos motivos son las razones principales para la utilización de *MongoDB* en este sistema porque permite: (i) definir un esquema dinámico, algo que aporta versatilidad en el momento que se quieren incluir datos adicionales en cada aparcamiento en particular; y (ii) el planteamiento de un sistema escalable de manera simple.

3.4.2.3. Librerías

En la etapa de diseño se definen las principales librerías para implementar los módulos descritos en el apartado 3.3. Se utiliza *OpenCV* —versión 4.0.1— para las tareas de visión artificial y *Tensorflow* —versión 1.13.1— para incluir aprendizaje automático en el sistema proporcionando la capacidad de reconocer los vehículos. La motivación de esta elección es similar a la del lenguaje de programación puesto que ambas librerías son referencias en sus áreas, tienen una gran documentación y comunidad y son software libre.

OpenCV (Open Source Computer Vision Library) es una de las librerías más utilizadas para aplicaciones de visión artificial. Está desarrollada en C y C++ lo que hace posible el desarrollo de software de visión artificial que necesita un alto grado de eficiencia y con una complejidad computacional elevada. Además, dispone de una librería en python —*OpenCV-Python*— que hace de API con el núcleo integrando completamente OpenCV en software desarrollado en Python.

TensorFlow es una de las librerías más utilizadas en aprendizaje automático y es de código abierto. Está enfocada al desarrollo de sistemas que requieren construir y entrenar redes neuronales, con el objetivo de dotar al software de capacidades similares al aprendizaje y razonamiento humano. Con las redes neuronales el sistema es capaz de analizar y detectar patrones previamente aprendidos.

Cabe destacar las siguientes librerías que tienen un peso importante en el desarrollo del sistema. Para obtener una mejor capacidad de cálculo y procesamiento de las imágenes se han utilizado principalmente: *numpy*, *scikit-learn*, *scipy* y *matplotlib*. Para la tareas de aprendizaje automático y la integración de redes neuronales se ha utilizado la librería *Mask_RCNN* de Matterport (*Mask_RCNN*, 2018) y *keras*. En la integración del sistema de base de datos se utiliza la librería *mongoengine* que hace de driver de MongoDB en Python.

3.5. Arquitectura

En este apartado se detalla la arquitectura propuesta para el despliegue de este sistema en un entorno real. Se elabora una descripción detallada de la arquitectura propuesta, los componentes hardware necesarios, y los protocolos y formas de comunicación utilizados para comunicar los distintos elementos.

3.5.1. Descripción general

La arquitectura del sistema se diseña teniendo en cuenta que los aparcamientos pueden presentar distribuciones y tamaños variados. Se pueden encontrar desde aparcamientos con un número de plazas reducido, a aparcamiento enormes con varias plantas.

Por ello, se plantea una arquitectura distribuida que tenga en cuenta estas restricciones y que permita el despliegue de la infraestructura del sistema en cualquier superficie independientemente de sus características. La arquitectura se compone de distintos nodos de computación con diferentes características y que realizan distintas tareas del sistema de manera modular. En la [figura 3.25](#) se

muestran los distintos nodos de computación que componen la arquitectura general del sistema descritos en el [apartado 3.5.2](#).

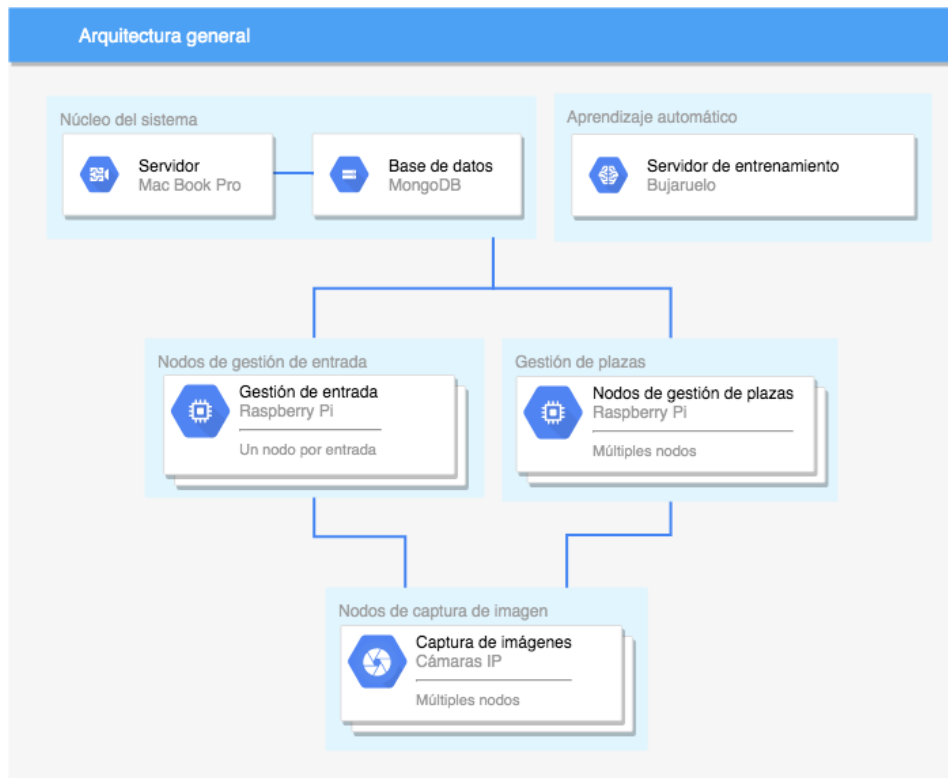


Figura 3.25. Arquitectura del sistema

3.5.2. Nodos de computación

En este apartado se describen los *nodos de computación* existentes en el sistema junto con las especificaciones hardware y la funcionalidad que se ejecuta en cada uno de ellos.

Cámara IP WiFi

Las *cámaras IP WiFi* son los dispositivos encargados de la captura de las imágenes sobre las que se evalúa el estado de ocupación de las plazas. Cada cámara captura un conjunto de plazas correspondiente a uno o varios bloques de aparcamiento. Se plantea la utilización de cámaras *WiFi* para evitar la instalación del cableado de red. Para la alimentación se pueden acoplar las cámaras, por ejemplo, en postes de luz del aparcamiento para disponer de toma de corriente y una ubicación con altura para colocar la cámara. En la [figura 3.26](#) se muestra el emplazamiento propuesto ejemplificado.

Para este trabajo y las pruebas de concepto se utiliza una cámara Canon EOS 1300D con la que se capturan las imágenes de la maqueta que modelan diferentes situaciones de un entorno real.



Figura 3.26. Ejemplo de emplazamiento de cámara IP

Nodos de gestión de plazas

Los *nodos de gestión de plazas* hacen de interfaz entre el servidor y las cámaras IP. Se conectan a una o varias cámaras IP en función de las capacidades de computación que permita el dispositivo utilizado. Si son capaces de procesar una imagen se conectan a una cámara y, en caso de poder procesar varias en paralelo, se conectan a varias. Esta decisión se toma teniendo en cuenta que en ningún caso se debe comprometer el rendimiento del sistema general por un cuello de botella en el procesamiento de estos nodos.

El envío masivo de imágenes y el procesamiento de todas ellas en el servidor no es eficiente ni escalable a menos que se aumenten las prestaciones de dicho servidor. Esto hace que no sea viable, ni desde el punto de vista económico, ni operativo. Por ello, cada nodo de gestión de plazas procesa localmente las imágenes de la/s cámara/s que controla y únicamente envía al servidor la información de ocupación actualizada. Los módulos de detección de vehículos y movimiento son incluidos en estos nodos. Las comunicaciones se describen en detalle en el [apartado 3.5.3](#).

Para este trabajo y las pruebas de concepto se utiliza una Raspberry Pi modelo 3 B+ como nodo de gestión de plazas.

Nodos de gestión de entrada

Los *nodos de gestión de entrada* son los encargados de procesar las imágenes de los vehículos entrantes para detectar las medidas. La decisión de utilizar este tipo de nodos viene motivada por las mismas razones que llevan a incluir los nodos de gestión de plazas.

Estos nodos se conectan con el servidor en el momento que un nuevo coche accede al aparcamiento indicando las características del vehículo. El servidor devuelve la ubicación óptima en base a los parámetros proporcionados. Los módulos de medición de vehículos y movimiento son incluidos en estos nodos.

Para este trabajo y las pruebas de concepto se utiliza una Raspberry Pi modelo 3 B+ como nodo de gestión de entrada.

Servidor

El servidor es el encargado de centralizar la información intercambiando datos con el resto de nodos. El servidor incluye: (i) el software que contiene la capa de negocio de la aplicación junto con los modelos de las clases descritas en el [apartado 3.4.1](#), (ii) el sistema de gestión de base de datos para persistir la información del sistema, y (iii) un servidor web utilizado por los nodos para la comunicación. El servidor proporciona una API Rest con la que interactúan el resto de nodos para obtener la información del sistema general. Para la implementación del API Rest se utiliza *Flask*, como framework sobre Python, que ya integra un servidor web. Los algoritmos de búsqueda de plazas de aparcamiento se implementan en el servidor.

La máquina empleada para hacer la función de servidor se puede plantear de manera física o en la nube. Un servidor físico es una máquina instalada *ad hoc* en cada aparcamiento para la gestión del mismo. Un servidor en la nube proporciona la posibilidad de un sistema online capaz de controlar múltiples aparcamientos simultáneamente, teniendo como requisito extra la necesidad de una conexión a internet. Además, presenta ventajas añadidas al servidor físico como la escalabilidad del sistema sin necesidad de mejorar una máquina física, y las actualizaciones periódicas y niveles de servicio que ofrecen los proveedores de infraestructura en la nube.

Para este trabajo y las pruebas de concepto se utiliza como servidor un Mac Book Pro con un procesador Intel Core i7 a 2,7 GHz y una memoria DDR3 de 6 GB 1600 MHz.

Servidor de entrenamiento de red neuronal

El entrenamiento de las redes neuronales es un proceso off-line que se hace antes de utilizar el sistema. La salida del entrenamiento se utiliza en el sistema en producción para detectar los vehículos presentes en cada imagen. Para este entrenamiento se utiliza un servidor proporcionado por la UCM que, gracias a sus prestaciones, permite optimizar la fase de aprendizaje del sistema. Las características del servidor *bujaruelo.dacya.ucm.es* son las siguientes:

- Linux bujaruelo 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64 GNU/Linux
- Intel(R) Xeon(R) CPU E5-2695 v3 @ 2.30GHz, 28 *cores*

3.5.3. Comunicaciones

Las comunicaciones entre los distintos nodos del sistema se hacen a través de una red WLAN mediante el estándar IEEE 802.11. La información se intercambia entre los nodos del sistema utilizando el formato JSON.

El modelo propuesto se basa en una arquitectura *cliente-servidor*. En el sistema se dispone de un servidor central que contiene la información del sistema y de múltiples clientes —nodos de gestión de plazas y gestión de entrada— que consultan y actualizan la información asociada a las plazas que controlan.

Capítulo 4

Resultados

Para analizar el sistema propuesto en este trabajo se plantean dos enfoques. Primero, se prueba la precisión del sistema para determinar el porcentaje de acierto que se obtiene en la detección de vehículos dentro del aparcamiento. Segundo, se analiza en términos de rendimiento los tiempos de ejecución del software en los nodos de detección de vehículos.

4.1. Precisión del sistema

En el análisis de la precisión del sistema se incluyen las siguientes medidas:

1. Precisión en la detección del número de plazas generales.
2. Precisión en la detección del número de plazas ocupadas.
3. Precisión en la determinación del área de las plazas en el sistema. Se analiza el error absoluto y relativo que se produce entre el área real de una plaza y la obtenida en el sistema.
4. Precisión en la determinación del área de los vehículos. Se analiza el error absoluto y relativo que se produce entre el área real de una plaza y la calculada en el sistema.
5. Precisión en la detección del área ocupada por vehículos en las plazas. Se analiza el error absoluto y relativo que se produce entre el área real ocupada por un vehículo en una plaza y la calculada en el sistema.

Con el análisis de los datos resultantes de la precisión en la detección de plazas se obtiene que:

- La precisión del sistema en el cálculo del número de plazas es del 100% detectando 42 plazas que equivalen a las totales existentes en el aparcamiento.
- La precisión en la determinación del área de las plazas es del 95,90%. En el [anexo 1](#), la [tabla A.1.1](#) muestra los resultados obtenidos en cada una de las plaza.

En la [tabla 4.1](#) se muestran los resultados de la precisión obtenidos en la determinación del área real que ocupan los vehículos calculada en el sistema. De este análisis se obtiene que la precisión es 92,12%.

Tabla 4.1. Resultados en la precisión en el cálculo del área real

Vehículo (#)	Área detectada (mm ²)	Área real (mm ²)	Error absoluto	Error relativo	Porcentaje de error (%)
Coche rojo	2063,36	2100,00	36,64	0,02	1,74 %
Coche amarillo	2360,62	2100,00	260,62	0,12	12,41 %
Coche naranja	1980,72	1680,00	300,72	0,18	17,90 %
Coche granate	1832,12	1720,00	112,12	0,07	6,52 %
Coche azul	2052,23	1998,00	54,23	0,03	2,71 %
Camioneta amarilla	2420,75	2280,00	140,75	0,06	6,17 %
Deportivo amarillo	1960,38	1820,00	140,38	0,08	7,71 %
			130,68	0,08	7,88 %
				0,92	92,12 %

En las [tabla 4.2](#) y [4.3](#) se analiza el desempeño del sistema en la determinación de ocupación del aparcamiento. En la [tabla 4.4](#) se observa que la precisión en la detección de ocupación de forma binaria —vacío y ocupado— es del 91%, mientras que la precisión en base a la detección del área ocupada por los vehículos en las plazas de aparcamiento es del 93,39%.

Tabla 4.2. Resultados de la precisión en la detección de la ocupación del aparcamiento en binario —ocupado y vacío—

Prueba #	Plazas ocupadas reales	Plazas ocupadas detectadas	Error absoluto	Error relativo	Porcentaje de error (%)
0	2,00	2,00	0,00	0,00	0,00 %
1	5,00	4,00	1,00	0,25	25,00 %
2	3,00	3,00	0,00	0,00	0,00 %
3	6,00	5,00	1,00	0,20	20,00 %
4	6,00	6,00	0,00	0,00	0,00 %
			0,40	0,09	9 %
				0,91	91 %

Tabla 4.3. Resultados de la precisión en la detección de la ocupación del aparcamiento.

Prueba (#)	Área detectada (mm ²)	Área real (mm ²)	Error absoluto	Error relativo	Porcentaje de error (%)
0	3780,00	4044,08	264,08	0,07	6,53 %
1	9600,00	10257,33	657,33	0,06	6,41 %
2	5820,00	6213,25	393,25	0,06	6,33 %
3	11598,00	12606,82	1008,82	0,08	8,00 %
4	11598,00	12309,56	711,56	0,06	5,78 %
			607,01	0,07	6,61 %
				0,93	93,39 %

4.2. Tiempos de ejecución

Puesto que la parte computacionalmente más costosa es la detección de los vehículos, ya que utiliza la red neuronal para el análisis de imágenes, y se hace en dispositivos hardware sencillos —Raspberry Pi—, con el fin de medir la bondad del sistema, se estudian los tiempos de ejecución por ser esta parte la más crítica del sistema. Con esto se busca evaluar la idoneidad de dispositivos hardware sencillos para realizar

esta tarea, y la viabilidad del sistema en general al calcular los tiempos de ejecución de su parte más crítica.

En la [tabla 4.4](#) se observan los tiempos de ejecución obtenidos en el análisis de un conjunto de imágenes resultando una media de 19,01s para detectar, en la Raspberry, los vehículos que aparecen en una imagen.

Tabla 4.4. Resultados de la precisión en la detección de la ocupación del aparcamiento.

Prueba (#)	Tiempo (s)
0	18,50
1	18,89
2	19,70
3	18,39
4	19,56
.....	
	19,01

Capítulo 5

Conclusiones

5.1. Análisis de resultados

En este trabajo se propone un sistema de detección de la ocupación de un aparcamiento mediante técnicas de visión por computador y aprendizaje automático. Las técnicas más utilizadas en la literatura hacen uso únicamente de visión por computador y técnicas de clasificación de imágenes para determinar el estado de ocupación de una plaza. Son pocos los trabajos previos en el área que hacen uso del aprendizaje automático para identificar los vehículos presentes en un aparcamiento para determinar la ocupación. Ninguno de ellos realiza un sistema con las características de este trabajo planteando la ocupación en función al área de ocupación de cada plaza.

Los resultados obtenidos en el capítulo 4 aportan datos que, a priori, pueden ser contradictorios. La *precisión* analizada del sistema es muy buena superando en todos los casos el 90% de acierto. Si a esto se añade que los resultados se basan en el acierto del sistema en la extracción de medidas y áreas sobre una superficie, se puede concluir que las expectativas del sistema planteado se han cumplido con éxito. La parte peor se encuentra en los resultados de los tiempos de ejecución. Si bien es cierto que en un aparcamiento la actualización no tiene que ser constante para obtener un sistema que conozca en tiempo real la ocupación, los tiempos obtenidos de 19 segundos para el procesado y detección de los vehículos, pueden ocasionar problemas de coherencia en los datos del sistema.

5.2. Cumplimiento de objetivos

Respecto a los objetivos planteados en el [apartado 1.3](#) a cumplir en el desarrollo de este trabajo, se puede concluir que se han cumplido todas las expectativas previstas. Concretamente:

- Se han estudiado los trabajos previos para detectar las técnicas más utilizadas y las que se utilizan en la actualidad con el objetivo de enfocar este trabajo hacia las nuevas tecnologías.
- Se ha diseñado e implementado un sistema que cumple con los requisitos establecidos.
- Se ha implementado una arquitectura distribuida para dar soporte a un sistema en el que se cumpla este requisito.
- Se ha modelado un entorno de pruebas en el que simular situaciones y casuísticas propias de un aparcamiento.
- Se han obtenido y analizado los resultados del sistema propuesto.

5.3. Trabajo futuro

Los trabajos futuros a este sistema comienzan con la mejora de las características del mismo para obtener mejores tiempos de respuesta en la detección de imágenes. Se plantea la búsqueda de alternativas a las redes neuronales —inclusive otro tipo de enfoque de red neuronal— que permitan identificar vehículos reduciendo los tiempos de respuesta, pero manteniendo la precisión. También, se plantea el estudio de dispositivos o arquitecturas diferentes para el procesamiento de las imágenes en la detección de vehículos sin que se produzcan cuellos de botella.

A posteriori, el sistema podría ser mejorado incluyendo el procesamiento en 3 dimensiones (3D), utilizando imágenes de varias cámaras para obtener un modelo 3D del aparcamiento. Esto haría que fuese posible su implementación en aparcamientos de interior donde la colocación de las cámaras no puede ser a gran altura.

Como colofón, el sistema propuesto podría ser utilizado para la detección de aparcamientos a pie de calle obteniendo imágenes de, por ejemplo, las cámaras de los semáforos. La detección de plazas sería un proceso manual en caso de no existir líneas delimitadoras, pero el planteamiento para la detección de la ocupación realizado en este trabajo, en el que se conoce el área ocupada de cada espacio, es

ideal como punto de partida para un trabajo enfocado a la gestión de la ocupación en cualquier espacio.

Chapter 1

Introduction

1.1. Problem identification

How much time do we waste looking for parking? Who hasn't felt stressed by having to drive around in the car looking for space while seeing that you're late for an important appointment, a meeting or an exam? These are realities and problems we face in populated cities with a high level of traffic. What's more, if we ask drivers why they are most frustrated behind the wheel we will have a clear answer, the search for parking.

According to Ludvik Eide (Eide, 2018), a study carried out in North America to evaluate the frustration caused by the search for parking concludes that: "61% of drivers reported feeling stressed while looking for a place to leave the car, 42% had not been able to go to an appointment for not finding parking, 34% had abandoned a trip due to parking problems and 23% had experienced anger at the wheel.

But it's not just about health. The parking problem has a negative impact on economic and environmental issues. According to a study by INRIX Research, Americans spend a total of \$73,000,000,000 (73 billion) on parking. He concludes that they invest an average of 17 hours per year per driver in what they call "the parking pain", reaching its peak in New York, where they lose 107 hours per driver per year (Searching for Parking Costs Americans \$73 Billion a Year, 2017).

Another company study reveals that in 2017 the cost of owning a vehicle averaged \$10,288 per driver in the United States. Among the expenses of ownership establishes the existence of two types of cost. Direct costs include maintenance, insurance, fuel

and parking and toll rates. Indirect or hidden costs including parking and parking fines. Indirect costs, including lost time searching for parking, represent more than 30% of the total cost. Dr. Graham Cookson, Chief Economist at INRIX, says: "The true cost of driving was staggering, but what really surprised was the size and breakdown of hidden costs. Parking, for example, represented a third of the total cost of owning a vehicle" (New INRIX Study Finds Parking is the Largest Cost of Driving, 2018). As shown in [figure 1.1](#), the total cost of parking search was 14% of the annual cost of vehicle ownership.

At the European level, INRIX Research analyses the situation in the United Kingdom. As can be seen in [figure 1.2](#), British drivers spend 44 hours a year looking for parking, at a cost of £733 per driver, making a total of £23.3 billion in the country. As can be seen in [figure 1.2](#), the search for parking, in the case of UK, accounts for more than 16% of the cost of ownership of the vehicle —[figure 1.2](#)—.

According to a Xerox study of the 19 largest cities in Europe (El 9% de los conductores pasan más de 15 minutos al día buscando aparcamiento, según un estudio de Xerox, 2016), 9% of drivers spend an average of 96 hours a year looking for parking, which is about 15 minutes a day. This figure increases in some European capitals including 19% of drivers and assuming a search time of up to 30 minutes.

With the evaluated data, we can conclude that there is a clear problem with parking in large populations that far from improving tends to worsen.

1.2. Motivation

It's a fact that most of us have ever been late for a meeting, a date or a movie started at the cinema for wasting too much time looking for parking. What's more, in places like shopping malls where there are systems that indicate the occupancy of squares, the laps looking for the precious space are constantly taking place. Even ignoring the extra factor of the co-driver who says —“come back, come back, you've skipped one”— to turn around and find a badly parked Smart. Wasting time looking for parking is a real problem that nobody likes.

In fact, without leaving our city, in Madrid it is very common to calculate travel time and add at least 15 minutes thinking about the search for parking. In short journeys in the capital, with good traffic conditions, do not exceed 30 minutes which means that we spend a third of the time of our trip in the search for parking.

If we add to this the fact that nowadays lack of time is one of the main reasons for stress, and the hidden costs involved in finding parking are considerable; we see that we are facing a serious problem that we all take for granted and assume.

And, although it is true that public transport is becoming more and more popular and more emphasis is being placed on sustainable mobility alternatives, such as Madrid Central, on which we have to focus for the future; the mass use of personal vehicles is a reality today that presents a difficult solution in the short and medium term.

Faced with this problem, there are car parks that have sought to provide a solution with sensors that indicate with a green light if the square is free and red otherwise, and incorporate information panels in the avenues of the car park indicating the number of free spaces in each area. [Chapter 2](#) explains in detail the existing solutions in the literature. By way of introduction, the use of sensors is an ineffective solution due to the fact that: (1) does not provide the driver with a knowledge of the exact position of the free spaces causing the driver to need to "spin" to find a place; and (2) depends on the placement of the vehicle inside the parking space for the sensor to correctly detect the occupation. If we add to this that the implementation of these systems involves a high cost and complex deployment due to the need to use a sensor for each parking space, which implies a high need for devices, wiring and control and monitoring systems, we find the first problems of this type of system.

If we already observe that this solution presents problems in closed spaces where anchorage points, sockets and conduits are available where necessary, it is logical to think that its implementation has even more disadvantages in open spaces, to the point of being unviable.

In addition, with the technological advances that we experience on a daily basis, it is difficult to understand that the most cutting-edge options in the area are more than ten years old (Lee et al, 2008). If we add to this the boom we are experiencing in artificial intelligence (AI) (Russell and Norvig, 2016), reflected in statements such as that of Hanoi Morillo, AI expert at Google, who says: "The world will be AI First" (Balbi, M). It is logical to think of a system capable of improving parking management making use of the advantages that artificial intelligence can provide.

For these reasons, the idea of this work is to use artificial intelligence to design and implement an intelligent system, distributed and adaptive, capable of:

- classify objects within an outdoor car park, differentiating which are vehicles;

- determine the occupation and distribution of these vehicles within the car park in real time;
- determine the size of the vehicle that accesses the car park;
- determine the optimal location for the incoming vehicle according to its dimensions and the occupation of the car park.

1.3. Work objectives

The main objective of this work is the creation of an expert system for the search of free spaces in a car park using artificial vision (AV) and machine learning (ML). In order to design a robust, efficient, distributed and adaptive system, the following division in general objectives to be achieved in the final system is proposed.

- **Research.** A first phase of research is carried out to evaluate the different alternatives existing in the market, and identify the technologies, algorithms and tools necessary for the implementation of the system.
- **Design and implementation.** The design and implementation are the central parts of this work and can be divided into the following specific objectives to be met by the system:
 - **Identification of parking spaces.** The system must be able to detect the lines of the car park in order to obtain a computer map of the parking spaces. For this, computer vision techniques are applied to analyse and search for patterns and lines in the images.
 - **Object detection.** Since vehicles coexist with other elements (people, trees, litter bins, etc.) within a car park, the system must be able to identify which vehicles are in order to determine their position and size within the car parks. To this end, neuronal networks (NN) are used to provide learning and knowledge to the system and, together with the VA, be able to make a classification of entities.
 - **Parking and vehicle measurements.** This is one of the great improvements with respect to the current systems. In order to avoid depending on the exact placement of vehicles within a parking space, a problem that appears in systems with sensors when a vehicle parks between two parking spaces occupying both and generating false

positives, it is proposed that the system be able to establish the free and occupied size within each parking space. In order to do this, it must be able to determine the measurements of each seat and the size and position of the cars parked in them.

- **Identification and measurement of the incoming vehicle.** Since the system knows at all times the remaining space in each of the parking spaces, it is proposed to measure the vehicle that accesses the enclosure in order to determine its optimal location.
- **Search algorithm.** An algorithm is designed to determine the ideal position for the vehicle taking into account the size and various factors specified in the description of the system.
- **System architecture.** Since one of the premises is to obtain a distributed and scalable system, the hardware architecture of the system has several computing nodes. In each one of these nodes the different specific objectives stated above are implemented.
- **Proof of concept.** The final objective is to obtain an expert system with which to carry out a proof of concept in order to evaluate the results obtained. For this purpose, the system deployment is designed and, by means of a battery of images, the performance and accuracy offered are evaluated.
- **Analysis of results and conclusions.** A final analysis of the results obtained is carried out, establishing the conclusions reached in the work together with possible future work.

1.4. Structure of the document

The structure of this document is made up of 5 chapters in which the different aspects related to this work are described.

Chapter 1 of Introduction, which contains the present section, is an introduction describing: the problems encountered (1.1), the motivation involved in solving the problem (1.2), the objectives to be achieved in the work in order to provide a viable solution (1.3) and the structure of the document (1.4).

Chapter 2 of State of the art contains a review of the studies prior to the development of the system, making a division into two sections: systems not based on artificial vision (2.1), which includes systems based on sensors and robotic car parks;

and systems based on artificial vision (2.2), paying special attention to systems based on neural networks. A final section (2.3) is added comparing the characteristics of the previous approaches and the system described in this paper.

Chapter 3 of the proposed System contains the central part of this report describing: the system in detail (3.1), the modeling of the test environment used to recreate different casuistry (3.2), the modules that include functionalities with artificial vision and automatic learning, (3.3) and the design (3.4) and architecture (3.5) proposed for the system.

Chapter 4 of Results shows the measures used to validate the performance and effectiveness of the system and provides quantitative results on the work done.

Chapter 5 of Conclusions contains the analysis of the results obtained in the previous chapter and the conclusions and future work that are raised as a result of this work.

Chapter 5

Conclusions

5.1. Analysis of results

This work proposes a system for detecting the occupation of a parking using computer vision and machine learning techniques. The most commonly used techniques in literature make use only of computer vision and image classification techniques to determine the occupancy status of a parking spot. Few previous works in the area make use of machine learning to identify vehicles present in a parking to determine occupancy. None of them carries out a system with the characteristics of this work, proposing the occupation according to the occupation area of each parking spot.

The results obtained in [chapter 4](#) provide data which, a priori, may be contradictory. The precision of the system analysed is very good, in all cases surpassing 90% success. If we add to this the fact that the results are based on the success of the system in extracting measurements and areas on a surface, we can conclude that the expectations of the proposed system have been met successfully. The worst part is in the results of the execution times. Although it is true that in a car park the update does not have to be constant to obtain a system that knows the occupation in real time, the times obtained of 19 seconds for the processing and detection of the vehicles, can cause problems of coherence in the data of the system.

5.2. Objectives achieved

With regard to the objectives set out in [section 1.3](#) to be met in the development of this work, it can be concluded that all expectations have been met. Specifically:

- Previous approaches have been studied in order to detect the most commonly used techniques and those currently in use, with the aim of focusing this work on new approaches.
- A system has been designed and implemented that meets the established requirements.
- A distributed architecture has been implemented to support a system that meets this requirement.
- A test environment has been modeled to simulate parking situations and cases.
- The results of the proposed system have been obtained and analysed.

5.3. Future Work

Future work on this system begins with the improvement of its characteristics in order to obtain better response times in image detection. The search for alternatives to neural networks—including another type of neural network approach—that allow the identification of vehicles, reducing response times, but maintaining accuracy, is proposed. Also, the study of different devices or architectures for image processing in vehicle detection without bottlenecks is proposed.

A posteriori, the system could be improved by including 3-dimensional (3D) processing, using images from several cameras to obtain a 3D model of the parking. This would make it possible to implement it in indoor car parks where the placement of the cameras cannot be at great height.

As a colophon, the proposed system could be used for the detection of street-level parking spots by obtaining images of, for example, traffic light cameras. The detection of parking spaces would be a manual process in the absence of boundary lines, but the approach to the detection of occupation made in this work, in which the occupied area of each space is known, is ideal as a starting point for work focused on the management of occupation in any space.

Bibliografía

Acharya, D., Yan, W., Khoshelham, K. (2018). Real-time image-based parking occupancy detection using deep learning. In: Proceedings of the 5th Annual Research, 2087.

Al-Absi, H.R.H., Devaraj, J.D.D., Sebastian, P., Voon, Y.V. (2010). Vision- based automated parking system. In: 10th International Conference on Information Science, Signal Processing and their Applications.

Al-Kharusi, H., Al-Bahadly, I. (2014). Intelligent Parking Management System Based on Image Processing. World Journal of Engineering and Technology, 2(2), 55-67.

AlexNet. (2019). [online] Recuperado de <https://en.wikipedia.org/wiki/AlexNet> [Accedido el 25 de Abril de 2019]

Almeida, P, Oliveira, L.S., Silva, E., Britto, A., Koerich, A. (2013). Parking space detection using textural descriptors. In: 2013 IEEE International Conference on Systems, Man, and Cybernetics.

Almeida, P., Oliveira, L.S., Britto, A., Silva, E., Koerich, A. (2015). PKLot – A robust dataset for parking lot classification. Expert Systems with Applications, 42(11), 4937-4949.

Amato, G., Carrara, F., Falchi, F., Gennaro, C., Meghini, C., Vairo, C. (2017). Deep learning for decentralized parking lot occupancy detection. Expert Systems with Applications, 72, 327-334.

Background subtraction. (2017). [online] Recuperado https://docs.opencv.org/3.3.0/db/d5c/tutorial_py_bg_subtraction.html [Accedido el 13 de Abril de 2019]

Balbi, M. (2018). Presente y futuro de la inteligencia artificial: cómo va a cambiar nuestro mundo. [online] Recuperado de <https://www.infobae.com/tendencias/>

[innovacion/2018/05/20/presente-y-futuro-de-la-inteligencia-artificial-como-va-a-cambiar-nuestro-mundo/](#) [Accedido el 2 de Abril de 2019]

Blumer, K., Halaseh, H.R., Ahsan, M.U., Dong, H., Mavridis, N. (2012) Cost-Effective Single-Camera Multi-Car Parking Monitoring and Vacancy Detection towards Real-World Parking Statistics and Real-Time Reporting. In: International Conference on Neural Information Processing, 506-515.

Bong, D.B., Ting, K.C., Rajae, N. (2006). Car-park occupancy information system. Real-Time Technol. Appl. Symp., 1-4.

Bong, D.B.L., Ting, K.C., Lai, K.C. (2008). Integrated approach in the design of car park occupancy information system (COINS). International Journal of Computer Science, 35(1), 1-8.

Callao Smart Parking. (n.d.). [online] Recuperado de <https://www.parkingcallao.es/> [Accedido el 10 de Abril 2019]

Canny Edge Detector. (2014). [online] Recuperado de https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html [Accedido el 25 de Abril de 2019]

Chen, L.C., Hsieh, J.W., Lai, W.R., Wu, C.X., Chen, S.Y. (2010). Vision-based vehicle surveillance and parking lot management using multiple cameras. In: 2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 631-634.

Choeychuen, K. (2012) Available car parking space detection from webcam by using adaptive mixing features. In: Ninth International Conference on Computer Science and Software Engineering.

Color histogram (2019). [online] Recuperado de https://en.wikipedia.org/wiki/Color_histogram [Accedido el 25 de Abril de 2019]

Dan, N. (2002). Parking management system and method. US Patent App. 10/066,215. [online] Recuperado de <https://patents.google.com/patent/US20030144890A1/en> [Accedido el 10 de Abril de 2019]

Delibaltov, D., Wu, W., Loce, R.P., Bernal, E.A. (2013) Parking lot occupancy determination from lamp post camera images. In: International IEEE Conference on Intelligent Transportation Systems.

Eide, L. (2018). Reducir el coste de buscar aparcamiento. [online] Recuperado de <https://www.easyparkpartners.com/es-blog/2018/6/20/reducir-el-coste-de-buscar-aparcamiento> [Accedido el 1 de Abril de 2019]

El 9% de los conductores pasan más de 15 minutos al día buscando aparcamiento, según un estudio de Xerox. (2016). [online] Recuperado de <https://noticias.xerox.es/2016/06/29/el-9-de-los-conductores-pasan-mas-de-15-minutos-al-dia-buscando-aparcamiento-segun-un-estudio-de-xerox/> [Accedido el 1 de Abril de 2019]

Euclidean distance. (2019). [online] Recuperado de https://en.wikipedia.org/wiki/Euclidean_distance [Accedido el 13 de Mayo de 2019]

Fabian, T. (2008). An algorithm for parking lot occupation detection. In: 7th Computer Information Systems and Industrial Management Applications.

Face Detection using Haar Cascades. (2018-2019). [online] Recuperado de https://docs.opencv.org/3.4.1/d7/d8b/tutorial_py_face_detection.html [Accedido el 13 de Mayo de 2019]

Funck, S., Mohler, N., Oertel, W. (2004). Determining car-park occupancy from single images. In: Intelligent Vehicle, IEEE Symposium.

Gálvez del Postigo, C., Torres, J., Menéndez, J.M. (2015). Vacant parking area estimation through background subtraction and transience map analysis. IET Intell Transp Syst, 9(9), 835–841

Godbehere, A.B., Matsukawa, A., Goldberg, K. (2012). Visual Tracking of Human Visitors under Variable-Lighting Conditions for a Responsive Audio Art Installation. In: American Control Conference.

González, F. (2018). Carmena cierra el único aparcamiento robotizado de Madrid tras cientos de averías. Recuperado de <https://okdiario.com/espana/carmena-cierra-unico-aparcamiento-robotizado-madrid-cientos-averias-2349559> [Accedido el 10 de Abril de 2019]

Huang, C.C., Dai, Y.S., Wang, S.J. (2012). A surface-based vacant space detection for an intelligent parking lot. In: 12th International Conference on ITS Telecommunications.

Huang, C.C., Tai, Y.S., Wang, S.J. (2013). Vacant parking space detection based on plane-based Bayesian hierarchical framework. IEEE Transactions on Circuits and Systems for Video Technology, 23(9), 1598 - 1610.

Huang, C.C., Vu, H.T. (2015a). A multi-layer discriminative framework for parking space detection. In: 25th International Workshop on Machine Learning for Signal Processing.

Huang, C.C., Vu, H.T., Chen, Y. R. (2015b). A multiclass boosting approach for integrating weak classifiers in parking space detection. In: IEEE International Conference on Consumer Electronics.

Huang, C.C., Wang, S.J., Chang, Y.J., Chen, T. (2008). A Bayesian hierarchical detection framework for parking space detection. In: IEEE International Conference on Acoustics, Speech and Signal Processing.

Huang, C.C., Wang, S.J. (2010). A hierarchical Bayesian generation framework for vacant parking space detection. IEEE Transactions on Circuits and Systems for Video Technology, 20(12), 1770-1785.

Ichihashi, H., Notsu, A., Honda, K., Katada, T., Fujiyoshi, M. (2009) Vacant parking space detector for outdoor parking lot by using surveillance camera and FCM classifier. In: IEEE International Conference on Fuzzy Systems.

Ichihashi, H., Notsu, A., Honda, K., Katada, T., Fujiyoshi, M. (2010). Improvement in the performance of camera based vehicle detector for parking lot. In: International Conference on Fuzzy Systems.

Jermurawong, J., Ahsan, U., Haidar, A., Haiwei, D., Mavridis, N. (2012). Car parking vacancy detection and its application in 24-hour statistical analysis. In: 10th International Conference on Frontiers of Information Technology.

KaewTraKulPong, P., Bowden, R. (2001). An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection. In: Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems, AVBS01. VIDEO BASED SURVEILLANCE SYSTEMS: Computer Vision and Distributed Processing.

Kernel (image processing). (2019). [online] Recuperado de [https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing)) [Accedido el 27 de de Abril de 2019]

Lee, S., Yoon, D., Ghosh, A. (2008). Intelligent Parking Lot Application Using Wireless Sensor Networks. In: International Symposium on Collaborative Technologies and Systems.

Lin, S., Chen, Y.Y., Liu S.C. (2006). A Vision-Based Parking Lot Management System. In: IEEE International Conference on Systems, Man and Cybernetics.

Liu, J., Mohandes, M., Deriche, M. (2013). A multi-classifier image based vacant parking detection system. In: IEEE 20th International Conference on Electronics, Circuits, and Systems.

Masmoudi, I., Wali, A., Jamoussi, A. (2014a). Parking spaces modelling for inter spaces occlusion handling. 22nd Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision. 119-124.

Masmoudi, I., Wali, A., Jamoussi, A., Alimi, A.M. (2014b). Vision based system for vacant parking lot detection: VPLD. In: Proc. Int. Conf. Comput. Vis. Theory Appl., 2, pp. 526–533.

Masmoudi, I., Wali, A., Jamoussi, A., Alimi, A.M. (2016). Multi agent parking lots modelling for anomalies detection while parking. IET Computer Vision, 10(5).

Mateus, P.A., Maldonado, E.O., Nino, C.L. (2015). Surveillance and management of parking spaces using computer vision. In: Proceedings of symposium on signal processing, images and computer vision, 53–59.

Median filter. (2019). [online] Recuperado de https://en.wikipedia.org/wiki/Median_filter [Accedido el 3 de Mayo de 2019]

Mordvintsev, A., Abid, K. (2013). What are contours?. [online] Recuperado de https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contours_begin/py_contours_begin.html#what-are-contours [Accedido el 3 de Mayo de 2019]

New INRIX Study Finds Parking is the Largest Cost of Driving. (2017). [online] Recuperado de <http://inrix.com/press-releases/cod-us/> [Accedido el 1 de Abril de 2019]

Nieto, R.M., García-Martín, A., Hauptmann, A.G., Martínez, J.M. (2018). Automatic Vacant Parking Places Management System Using Multicamera Vehicle Detection. IEEE Transactions on Intelligent Transportation Systems, 20(3), 1069-1080.

OpenCV. (2019). [online] Recuperado de <https://opencv.org/> [Accedido el 3 de Mayo de 2019]

Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, And Cybernetics, *SMC-9*(1).

Paidi, V., Fleyeh, H., Håkansson, J., Nyberg, R. G. (2018). Smart Parking Tools Suitability for Open Parking Lots: a Review. In: Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems, 1(RESIST), 600-609.

Pajares, G., De la Cruz, J. (2007). Visión por Computador: Imágenes Digitales y Aplicaciones.

Penalva, J. (2006). Primer aparcamiento robotizado en Madrid. [online] Recuperado de <https://www.xataka.com/robotica-e-ia/primer-aparcamiento-robotizado-en-madrid> [Accedido el 10 de Abril de 2019]

Penedo, M.F. (2009). Métodos de gradiente. [online] Recuperado de http://www.varpa.org/~mgpenedo/cursos/Ip/Tema7/nodo7_2.html [Accedido el 25 de Abril de 2019]

Plano cenital. (2019). [online] Recuperado de https://es.wikipedia.org/wiki/Plano_cenital [Accedido el 5 de Mayo de 2019]

Procesamiento basado en ROI. (2018). Recuperado de <https://es.mathworks.com/help/images/roi-based-processing.html> [Accedido 13 Mayo 2019]

Python. (2019). [online] Recuperado de <https://www.python.org/> [Accedido el 5 de Mayo de 2019]

Quién es Stan, el robot Valet Parking que trabajará en el aeropuerto de Londres (2019). [online] Recuperado de <https://www.infobae.com/autos/2019/02/01/como-es-stan-el-robot-valet-parking-que-trabajara-en-el-aeropuerto-de-londres/> [Accedido el 10 de Abril de 2019]

Redes neuronales convolucionales. (2019). [online] Recuperto de https://es.wikipedia.org/wiki/Redes_neuronales_convolucionales [Accedido el 12 de Mayo 2019]

Russell, S. J., Norvig, P. (2016). Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited.

Sastre, R.L., Jiménez, G.P., Acevedo, F.J., Bascón, M.S. (2007). Computer algebra algorithms applied to computer vision in a parking management system. In: IEEE International Symposium on Industrial Electronics.

Searching for Parking Costs Americans \$73 Billion a Year. (2017). [online] Recuperado de <http://inrix.com/press-releases/parking-pain-us/> [Accedido el 1 de Abril de 2019]

Searching for Parking Costs the UK £23.3 Billion a Year. (2017). [online] Recuperado de <http://inrix.com/press-releases/parking-pain-uk/> [Accedido el 1 de Abril de 2019]

Sevillano, X., Màrmol, E., Fernandez-Arguedas, V. (2014). Towards Smart Traffic Management Systems: Vacant On-Street Parking Spot Detection Based on Video Analytics. FUSION 2014 - 17th International Conference on Information Fusion.

Sevillano, X., Màrmol, E. (2016). QuickSpot: a video analytics solution for on-street vacant parking spot detection. Multimedia Tools and Applications, 75(24), 17711–17743.

Shin, J., Jun, H. (2014). A study on smart parking guidance algorithm. Transportation Research Part C: Emerging Technologies, 44, 299-317.

Stanley Robotics (n.d.). [online] Recuperado de <https://stanley-robotics.com/> [Accedido el 10 de Abril de 2019]

True, N. (2007). Vacant parking space detection in static images. [online] Recuperado de <http://cseweb.ucsd.edu/classes/wi07/cse190-a/reports/ntrue.pdf> [Accedido el 10 de Abril de 2019]

Tschentscher, M, Neuhausen, M., Koch, C., König, M., Salmen, J., Schlipf, M. (2013) Comparing image features and machine learning algorithms for real-time parking space classification. ASCE International Workshop on Computing in Civil Engineering.

Viola, P., Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

Visión artificial. (2018). [online] Recuperado de https://es.wikipedia.org/wiki/Visi%C3%B3n_artificial [Accedido el 13 de Mayo de 2019]

- Vu, T., Huang, C.-C. (2019). Parking Space Status Inference Upon a Deep CNN and Multi-Task Contrastive Network With Spatial Transform. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(4).
- Wang, X., Hanson, A. (1998). Parking Lot Analysis and Visualization from Aerial Images. In: *Proceedings Fourth IEEE Workshop on Applications of Computer Vision. WACV'98* (Cat. No.98EX201).
- Wang, Y., Hu, Y., Hu, X., Zhao, Y. (2015). Vision-based method for parking space surveillance and parking lot management. In: *Proceedings of international conference on image and graphics*, 516–528.
- Wu, Q., Huang, C., Wang, S., Chiu, W., Chen, T. (2007) Robust parking space detection considering interspace correlation. In: *IEEE International Conference on Multimedia and Expo*.
- Wu, Q., Zhang, Y. (2006). Parking lots space detection. *Mach Learn Fall*. [online] Recuperado de https://www.cs.cmu.edu/~epxing/Class/10701-06f/project-reports/wu_zhang.pdf [Accedido el 15 de Abril de 2019]
- Yusnita, R., Norbaya, F., Basharuiddin, N. (2012). Intelligent Parking Space Detection System Based on Image Processing. *International Journal of Innovation, Management and Technology*, 3(3).
- Xie, H., Wu, Q., Chen, B., Chen, Y., Hong, S. (2015). Vehicle detection in open parks using a convolutional neural network. In: *Sixth International Conference on Intelligent Systems Design and Engineering Applications (ISDEA)*.
- Zivkovic, Z. (2004). Improved Adaptive Gaussian Mixture Model for Background Subtraction. In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004*.
- Zivkovic, Z, Heijden, F. (2006). Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7), 773-780.

Anexo 1

Resultados

Tabla A.1.1. Resultados en la precisión en la determinación del área de las plazas

Plaza (#)	Área detectada (mm ²)	Área real (mm ²)	Error absoluto	Error relativo	Porcentaje de error (%)
1	5050,30	4967,25	83,05	0,02	2,00 %
2	5042,17	4967,25	74,92	0,02	2,00 %
3	4723,51	4967,25	243,74	0,05	5,00 %
4	4643,73	4967,25	323,52	0,07	7,00 %
5	4827,19	4967,25	140,06	0,03	3,00 %
6	5099,03	4967,25	131,78	0,03	3,00 %
7	4696,76	4967,25	270,49	0,05	5,00 %
8	4831,49	4967,25	135,76	0,03	3,00 %
9	4860,63	4967,25	106,62	0,02	2,00 %
10	4670,48	4967,25	296,77	0,06	6,00 %
11	4987,71	4967,25	20,46	0,00	0,00 %
12	4838,65	4967,25	128,60	0,03	3,00 %
13	4745,01	4967,25	222,24	0,04	4,00 %
14	4971,47	4967,25	4,22	0,00	0,00 %
15	4875,92	4967,25	91,33	0,02	2,00 %

16	4876,87	4967,25	90,38	0,02	2,00 %
17	5192,19	4967,25	224,94	0,05	5,00 %
18	4810,94	4967,25	156,31	0,03	3,00 %
19	4832,92	4967,25	134,33	0,03	3,00 %
20	5274,84	4967,25	307,59	0,06	6,00 %
21	5190,28	4967,25	223,03	0,04	4,00 %
22	5388,54	4967,25	421,29	0,08	8,00 %
23	5288,69	4967,25	321,44	0,06	6,00 %
24	5027,84	4967,25	60,59	0,01	1,00 %
25	5259,55	4967,25	292,30	0,06	6,00 %
26	4916,05	4967,25	51,20	0,01	1,00 %
27	5033,57	4967,25	66,32	0,01	1,00 %
28	5465,46	4967,25	498,21	0,10	10,00 %
29	5196,01	4967,25	228,76	0,05	5,00 %
30	4790,88	4967,25	176,37	0,04	4,00 %
31	4792,31	4967,25	174,94	0,04	4,00 %
32	4098,14	4967,25	869,11	0,17	17,00 %
33	5528,05	4967,25	560,80	0,11	11,00 %
34	4928,47	4967,25	38,78	0,01	1,00 %
35	4807,12	4967,25	160,13	0,03	3,00 %
36	4761,26	4967,25	205,99	0,04	4,00 %
37	4627,01	4967,25	340,24	0,07	7,00 %
38	4735,46	4967,25	231,79	0,05	5,00 %
39	4876,39	4967,25	90,86	0,02	2,00 %
40	4753,13	4967,25	214,12	0,04	4,00 %
41	4848,21	4967,25	119,04	0,02	2,00 %
42	4987,71	4967,25	20,46	0,00	0,00 %
			203,64	0,04	4,10 %
				0,96	95,90 %

