

Proyecto CarDensityAI

Documento de Requisitos del Sistema (DRS)

Autor: Francisco Vázquez Donaire

Autora: Blanca Rodríguez González

Modelo: IEEE/ANSI 830-1998

Índice

1. Introducción	2
1.1. Propósito del documento	2
1.2. Alcance del sistema	2
1.3. Definiciones, acrónimos y abreviaturas	3
1.4. Referencias	3
1.5. Descripción del resto del documento	4
2. Descripción General	5
2.1. Perspectiva del Producto	5
2.2. Funciones del Producto	5
2.3. Características de los Usuarios	6
2.4. Restricciones generales	6
2.5. Suposiciones y Dependencias	7
2.6. Requisitos futuros	8
3. Requisitos Específicos	9
3.1. Funciones	9
3.2. Requisitos de Rendimiento	9
3.3. Restricciones de Diseño	10
3.4. Atributos del Sistema	10
3.5. Otros Requisitos	10
4. Apéndices	11

1. Introducción

En esta sección se realizará una primera introducción al documento de requisitos del sistema. En concreto, se introducirá el problema que el cliente quiere resolver. Para la garantizar la compresión de la solución propuesta, se explicarán algunas definiciones, acrónimos y abreviaturas de los conceptos que se tratarán en el resto del documento; así como se darán algunas referencias.

Finalmente, se definirá la estructura del resto del documento. El documento se desarrollará siguiendo las instrucciones plasmadas en el estándar de especificación de requisitos Software IEEE/ANSI 830-1998.

1.1. Propósito del documento

El presente documento de requisitos esta directamente dirigido a José Vélez Serrano, miembro del Excmo. Ayuntamiento contratador; así como a la directiva de la empresa que proporciona el servicio, *AI solutions*.

El escrito tiene como objetivo recoger los requisitos necesarios para la correcta implementación del proyecto. Se hará una breve introducción al sistema de visión artificial propuesto, ofreciendo, además un presupuesto estimado del mismo en la documentación adjunta.

1.2. Alcance del sistema

Con el fin de estimar la densidad de tráfico de un núcleo urbano específico, se plantea la implementación de un sistema de visión artificial, capaz de realizar un recuento automático de coches a partir de imágenes aéreas de la ciudad.

De forma simplificada, se sugiere como solución el diseño de una biblioteca específica basada en técnicas clásicas de procesamiento de imagen y aprendizaje profundo. La biblioteca desarrollada será capaz de analizar dichas imágenes, obteniendo como resultado el número total de automóviles presentes, así como sus coordenadas. A partir de este punto nos referiremos a esta biblioteca como **CarDensityAI**

Inicialmente, **CarDensityAI** estará únicamente diseñada para la detección y conteo de vehículos de cuatro ruedas, incluyendo, coches, furgonetas o camiones. Versiones futuras de la aplicación plantearán la detección de otros automóviles, incluyendo motocicletas o bicicletas.

Para optimizar su uso en el día a día de los trabajadores del Ayuntamiento, **CarDensityAI** se incrustará en una aplicación web en la cual el cliente podrá subir una imagen de este tipo y ver el resultado.

El uso final de **CarDensityAI** tiene como cometido acelerar los tiempos de procesado de las imágenes con respecto al procesamiento manual realizado por trabajadores del ayuntamiento. Además, el correcto funcionamiento del sistema propuesto supondrá un paso adelante para el desarrollo de proyectos futuros, como la localización de coches en carreteras.

1.3. Definiciones, acrónimos y abreviaturas

Para la comprensión de secciones futuras, se definen y acotan términos que se usarán a lo largo del documento.

- **Vehículo o automóvil:** En el ámbito de este proyecto, se denominará vehículo a todo aparato de cuatro ruedas empleado para el transporte de mercancías o personas. En este grupo entrarán coches, furgonetas, camiones o autobuses, entre otros. No se considera vehículo a las motocicletas, bicicletas u otros semejantes.
- **Aprendizaje profundo o Deep Learning (DL):** el DL pertenece a la familia de algoritmos de aprendizaje máquina, cuyo objetivo reside en la búsqueda de relaciones complejas que permitan a una máquina aprender a partir de un conjunto de datos, sin estar expresamente programada para ello. De forma similar, el DL aprenderá de forma progresiva, intentando imitar al cerebro humano mediante el uso de estructuras en forma de red neuronales o convolucional.
- **Red Neuronal:** Función no lineal que transforma unos datos de entrada en unos datos de salida usando diferentes niveles o capas ocultas. Dependiendo de nuestro objetivo, las redes pueden ser parametrizadas para resolver problemas de regresión, clasificación e, incluso, detección.
- **Tarea de Detección:** conjunto de técnicas de DL aplicadas a visión cuyo objetivo es la localización de diferentes objetos de interés en una imagen
- **Bounding Box o bbox:** Cuadro delimitador que recoge en su interior todos los elementos del objeto detectado.
- **YOLT:** *You Only Look Twice* es una red neuronal *open-source* muy utilizada en tareas de detección de objetos. Será la que se utilice para implementar la aplicación.
- **Transfer Learning:** Técnica que permite el uso de arquitecturas de redes neuronales y convolucionales preentrenadas con un gran número de datos en problemas cuyo conjunto de entrenamiento es de tamaño reducido. Se actualizarán, únicamente, las últimas capas de la red.

1.4. Referencias

- **Conceptos Deep Learning - "Deep Learning"; I. Goodfellow, Y. Bengio, A Courville (2016)**
- **YOLT - "You Only Look Twice: Rapid Multi-Scale Object Detection in Satellite Imagery"; A. Van Etten (2018) [GitHub]**
- **SOLID - Principios para desarrollar código de calidad. [web]**
- **OpenCV - Biblioteca *open source* estándar para procesamiento de imagen. [web]**
- **LabelImg - GitHub**

1.5. Descripción del resto del documento

Las siguientes secciones del documento están destinadas a realizar una descripción general del sistema propuesto (Sección 2), así como a listar los requisitos más específicos del mismo (Sección 3). Precisamente, se detallarán todas las funciones del producto, sus dependencias, y las suposiciones tomadas para implementarlo. Además, se discutirán las restricciones del producto final así como se realizará un diagrama gráfico de como un usuario podría utilizar este sistema.

2. Descripción General

En esta sección se delinean todos los elementos a considerar para el diseño de **CarDensityAI**, así como aquellos que afecten a la definición de los requisitos. El fin de este apartado es adquirir una visión global del funcionamiento del sistema, sin entrar en detalle.

2.1. Perspectiva del Producto

El sistema de detección y conteo de vehículos a partir de imágenes se desarrollará empleando el lenguaje de programación **Python**. En este sentido, se desarrollará una biblioteca bajo el nombre **CarDensityAI**, creada explotando diferentes funcionalidades de otras bibliotecas como OpenCV o TensorFlow, también implementadas en Python.

De forma resumida, el sistema propuesto estará basado en métodos actuales de Deep Learning diseñados para la detección de objetos. Precisamente, se implementará una versión de YOLO, conocida como YOLT (You Only Look Twice), diseñada, específicamente, para la detección de objetos en imágenes aéreas.

Además, para facilitar y mejorar la experiencia a nivel usuario, la biblioteca se implantará en una aplicación web mediante el uso de HTML, CSS y JavaScript.

El sistema final estará diseñado para progresar, entrenándose con nuevos datos conforme estos se reciban.

2.2. Funciones del Producto

El sistema de visión artificial propuesto recibirá una imagen aérea de un núcleo urbano específico como la mostrada en Figura 1. Sobre ella, se realizará un recuento de vehículos, independientemente de si estos se encuentran circulando o estacionados, tanto en vía pública como privada. Además del conteo, la presente aplicación será capaz de detectar las coordenadas de los vehículos presentes en las imágenes, introduciendo la posibilidad de pintar *bounding boxes* alrededor de ellos.

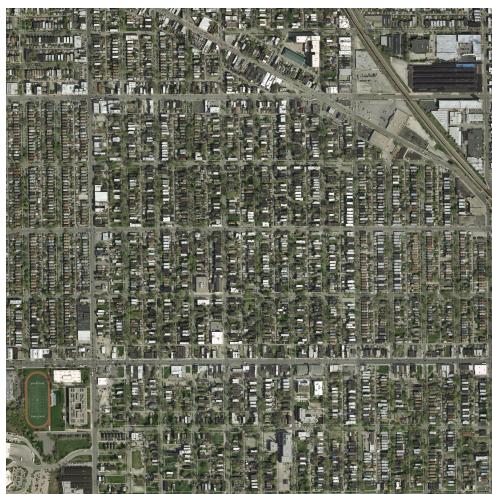


Figura 1: Vista aérea de una ciudad sobre la que se realizará el análisis.

La salida del sistema quedará guardada en un fichero *csv* que guardará el número total de vehículos que se encuentran en la imagen introducida, así como las coordenadas de los mismos. Es importante destacar que el desarrollo de este proyecto servirá de base para proyectos futuros.

De manera simplificada, el uso de **CarDensityAI** se resume en la Figura 2.

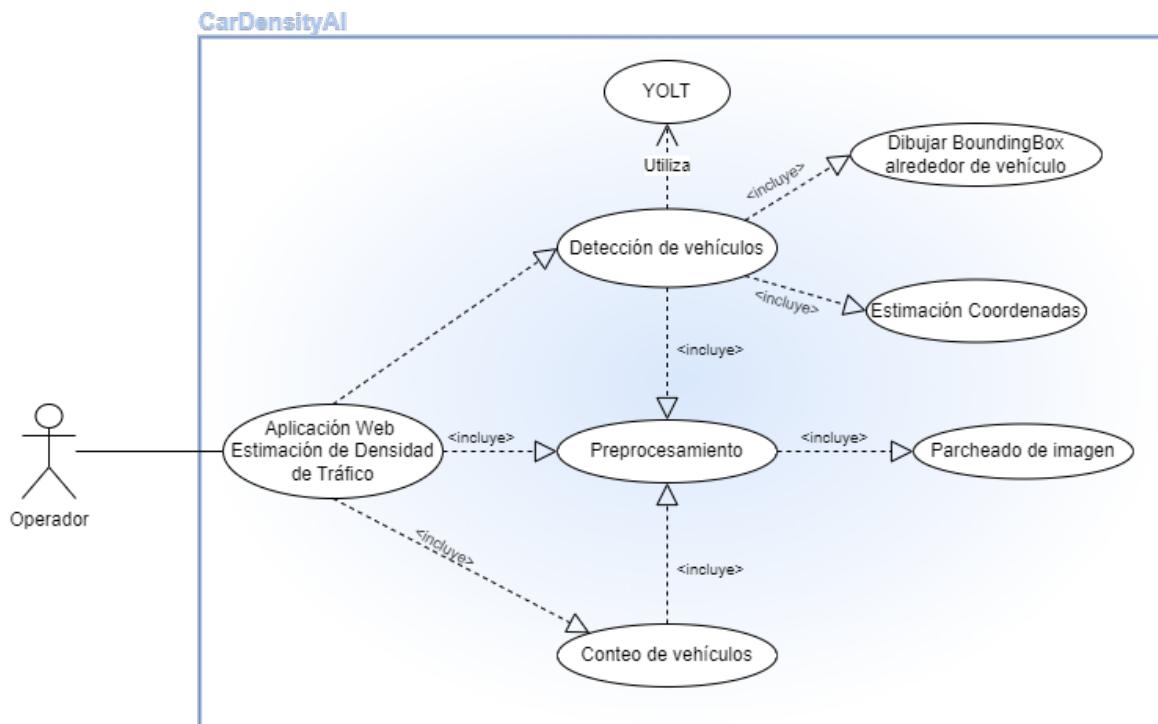


Figura 2: Casos de uso del sistema CarDensityAI

2.3. Características de los Usuarios

El sistema será manejado por empleados administrativos del Ayuntamiento (cliente), quienes introducirán las imágenes de la ciudad a estudiar y recibirán la información solicitada. Esto permitirá a los empleados realizar los estudios pertinentes entorno a la densidad del tráfico en ciudades.

Debido a la falta de conocimiento sobre la formación de los empleados del ayuntamiento, tras el desarrollo de la aplicación, los desarrolladores se comprometen a realizar breves cursos educacionales para el uso del sistema.

2.4. Restricciones generales

El sistema que se plantea ha de ser capaz de trabajar con ciertas limitaciones, especificadas a continuación:

- Limitaciones del hardware - El sistema se ejecutará en un servidor convencional de la empresa, que, en principio, no necesita de unas prestaciones muy elevadas. Con 8GB de RAM y un Intel i5 será suficiente.

- Operaciones paralelas - Durante el desarrollo del software, se utilizará GPU, lo que permitirá mayor capacidad de computo y paralelizar operaciones en el entrenamiento de la inteligencia artificial empleada. Una vez el sistema sea integrado en el hardware receptor, esto no será necesario, ya que no se necesita una gran capacidad de procesado.
- Lenguajes de programación - Se utilizará Python por su versatilidad. El uso de este lenguaje frente a otros de bajo nivel supondrá una reducción de tiempos durante el desarrollo. Además, se utilizará HTML, CSS y JavaScript para el desarrollo de la interfaz web (Frontend).
- Protocolos de comunicación - Habrá dos maneras de utilizar el software desarrollado. La primera consistirá en acceder vía *http* mediante la interfaz web. La segunda consistirá en llamar a los métodos de la biblioteca de Python desarrollada directamente desde la terminal.
- Criticidad de la aplicación- En caso de no superar un umbral de confianza en la detección del automóvil, se activará una alarma, que mostrará la escena en duda al humano, buscando su validación o rechazo.
- Consideraciones acerca de la seguridad - El sistema garantizará la confidencialidad de las imágenes empleadas.

2.5. Suposiciones y Dependencias

Para el correcto desarrollo del sistema, el cliente proporcionará a los desarrolladores 10 imágenes áreas de prueba con las que se prevé entrenar el sistema planteado. Las imágenes se entregarán en formato *.tif* y han de tener la suficiente calidad y tamaño como para ser divididas en imágenes de menor tamaño; lo que nos permitirá incrementar el tamaño de nuestra base de datos, ayudándonos a entrenar el modelo con suficiente precisión y acelerar los tiempos de procesamiento.

Tras una inspección de las imágenes entregadas, se observa que todas ellas se adquieren con condiciones de iluminación similares, así como en condiciones meteorológicas favorables, lo que facilitará exponencialmente el entrenamiento.

Es importante mencionar que, una vez el sistema sea implantado en el Ayuntamiento, las imágenes a estudiar han de cumplir las mismas características que las empleadas en el desarrollo. En caso contrario, no se asegura un correcto funcionamiento del sistema.

Las imágenes entregadas por el cliente no se encuentran etiquetadas, es decir, no existe una *verdad* de referencia que indique las posiciones y número de coches en cada una de las imágenes. Disponer de esta información es esencial para el entrenamiento del sistema. Se plantean, por tanto dos opciones:

- Etiquetado manual empleando alguna herramienta de código abierto, como LabelImg, o desarrollando un código propio.
- Búsqueda y uso de una base de datos similar a la entregada con las imágenes etiquetadas.

El sistema original se desarrollará sobre una versión de Python 3.1 o superior, empleando librerías como OpenCV (en versión 4.7) y TensorFlow.

2.6. Requisitos futuros

Para la mejora de rendimiento del sistema, se necesitarán más imágenes de prueba, que se podrán ir recolectando conforme la aplicación reciba nuevas imágenes. Esto permitirá introducir otras funciones a petición del cliente.

3. Requisitos Específicos

En esta sección se exponen los requisitos que se han de cumplir para el desarrollo satisfactorio del sistema CarDensityAI. La descripción de las especificaciones se hará de manera detallada, asegurando la consistencia y precisión de los mismos. Estos requerimientos ayudarán a los desarrolladores en la planificación del diseño de la aplicación, así como en el diseño de las pruebas unitarias a las que este se someterá.

3.1. Funciones

El software que se propone desarrollar ha de cumplir una serie de funciones acordadas con el cliente. Para garantizar la simplicidad del documento, se listan, a continuación, las funciones ha realizar atendiendo al **objetivo** de las mismas:

- **Reconocimiento y conteo de vehículos:** A petición del cliente, el sistema a desarrollar, ha de ser capaz, dado una imagen del núcleo urbano que se desee estudiar, de devolver el número de vehículos que se encuentran en ella.
Para cumplir con esta intención, el cliente entregará 10 imágenes de prueba en formato *tif* a color (3 canales) y de dimensiones 5000x5000. Con el fin de aumentar la muestra y motivados por el gran tamaño de las imágenes, estas se parchearán, consiguiendo por tanto un número mayor de imágenes para entrenar y probar el sistema.
- **Detección de vehículos:** Siguiendo la linea de la especificación anterior, además de devolver el número de coches, el cliente pide que se realice una detección de los mismos, devolviendo las coordenadas en las que se encuentren os vehículos detectados. Esto ayudará a conocer lo que detecta el sistema como vehículo y servirá, además, como base para otros proyectos.
Para la puesta a punto de esta función, se parten de las mismas imágenes que las mencionadas anteriormente.

3.2. Requisitos de Rendimiento

Uno de los intereses mas notorios del cliente es la mejora de tiempos de procesamiento de una ciudad respecto al procesamiento manual realizado por los operarios. De media, se estima que un operario medio procese una imagen cada 3 h. Este ritmo supone que una ciudad completa, formada por 400 imágenes, se procese en 1 mes. Se pretende, por tanto, desarrollar un sistema que mejore exponencialmente los tiempos. No se aceptará un sistema con tiempos peores.

Actualmente, se estima que la tasa de error medio de la linea de procesamiento de las imágenes de la ciudad cometida por los humanos se encuentra en torno al 5 %. El objetivo no es mejorarlo; no obstante, se tratará de reducir la tasa.

Al trabajar con sistemas de inteligencia artificial, estamos expuestos a que la máquina no devuelva un resultado con suficiente confianza. Al exponer esta situación al cliente, este indica que, en caso de que el valor de confianza del sistema se encuentre por debajo de un umbral establecido (se fijará atendiendo a los resultados), se activará una alarma, buscando validación humana.

No será necesario que el sistema funcione en tiempo real, puesto que el objetivo es el estudio de densidad de tráfico a posteriori.

3.3. Restricciones de Diseño

- En esta primera versión, el sistema será capaz de detectar únicamente vehículos de cuatro ruedas. En posteriores versiones se incluirán otros tipos de vehículos como bicicletas, motocicletas, etc.
- Se empleará Programación Orientada a Objetos para el diseño de la biblioteca.
- Como en todos los proyectos de esta compañía, se seguirá una metodología de desarrollo ágil; esto significa que, es posible aceptar nuevos requisitos conforme avanza el proyecto. Se aplicarán los principios *SOLID* para el desarrollo del mismo..
- El sistema funcionará con imágenes de 5000x5000 píxeles de tamaño, y tardará algunos minutos en procesarlas por completo. Si la imagen no tiene suficiente calidad, el sistema no analizará la imagen.
- El sistema estará *dockerizado* y será accesible vía *http*. Además, se entregará al cliente el sistema en tipo biblioteca de Python para que pueda testearlo de manera sencilla.

3.4. Atributos del Sistema

CarDensityAI cumplirá garantías de accesibilidad, seguridad, fiabilidad, mantenibilidad y portabilidad. Todas ellas se detallan a continuación:

- El sistema será accesible desde cualquier entorno, ya que consistirá en una aplicación web corriendo en un servidor, que será accesible incluso para sistemas iOS y Android.
- En su primera versión, el sistema no contará con sistema de autenticación, bastará con tener acceso de usuario al ordenador donde este se integre. No obstante, es una característica a estudiar para incluir en versiones futuras en caso de que el cliente así lo estime.
- El sistema tendrá una fiabilidad superior al 90 %, y reducirá de manera drástica los tiempos de procesamiento de las imágenes con respecto a la metodología anterior.
- El sistema será mantenable debido a que se desarrollará siguiendo los principios *SOLID* de desarrollo de software.
- El sistema pasará por servicios de análisis de código estático (*SonarCloud*) que asegurarán la calidad del código.

3.5. Otros Requisitos

El sistema será diseñado cumpliendo con la normativa vigente a nivel europeo y nacional de desarrollo de proyectos software.

4. Apéndices

El cliente podrá visualizar el progreso del desarrollo del sistema así como acceder a la información necesaria en el repositorio de GitHub creado por los autores con este fin.