

Documento de Especificación de Requisitos

23 de marzo de 2022

Grupo E

Alumnos:

David Correas Oliver
Elena Giraldo del Viejo
Juan Manuel Molina Maza

Índice

1. Introducción	3
1.1. Propósito	3
1.2. Ámbito del sistema	3
1.3. Definiciones, acrónimos y abreviaturas	4
1.4. Descripción	4
1.5. Personal involucrado	4
2. Descripción general	5
2.1. Perspectiva del proyecto	5
2.2. Funciones del producto	5
2.3. Características de los usuarios	5
2.4. Restricciones	6
2.5. Suposiciones y dependencias	7
3. Requisitos específicos	8
3.1. Requisitos funcionales	8
3.2. Requisitos no funcionales	8

1. Introducción

Primeramente, en esta sección se aborda una introducción al presente documento de especificación de requisitos. Se estructura en varios apartados: propósito, ámbito del sistema, definiciones y siglas, descripción y personal involucrado.

1.1. Propósito

El objetivo fundamental de esta memoria es proporcionar al organismo o empresa solicitante un análisis de viabilidad y riesgos sobre el problema en cuestión en base a los requisitos formulados. Se pretende proporcionar al cliente una herramienta software que se ha denominado *TrafficDetector*. Esta herramienta debe ser capaz, en una primera versión, de detectar vehículos a partir de imágenes aéreas de satélite. En posteriores versiones de la herramienta se profundizará en otros aspectos adicionales a la propia detección de vehículos.

1.2. Ámbito del sistema

Como se ha comentado en la sección anterior, se aspira a desarrollar un sistema de visión artificial encargado de la detección o la localización de vehículos de cualquier tamaño o color que se encuentren presentes en las imágenes adquiridas.

Para ello, se aplicarán técnicas clásicas de procesamiento de imagen que pueden constituir el modelo completo de detección de vehículos o, en su defecto, pueden considerarse como una manera de preprocesar las imágenes para aplicar posteriormente técnicas de inteligencia artificial, en concreto algoritmos de Deep Learning. Uno de los requisitos imprescindibles para el futuro modelo es su robustez, es decir, que sea capaz de reconocer los vehículos sin importar la iluminación, el tamaño, el color o la orientación en cada imagen.

La capacidad de detección en cualquier entorno de la futura aplicación dependerá del tamaño y la variabilidad de la base de datos. Si se utilizan únicamente técnicas clásicas, cuanto más representativa sea la muestra de imágenes con la que se pruebe el algoritmo, será menos probable que la detección sea errónea con una imagen nueva. Además, si finalmente se usan técnicas de Deep Learning, es fundamental este requisito para evitar que el modelo se adapte demasiado a los datos y permitirle generalizar para cualquier tipo de imagen de entrada.

En principio, no es necesaria una supervisión manual de la herramienta software por lo que no se desarrollará una interfaz muy sofisticada para la aplicación. Simplemente el cliente seleccionará un conjunto original de imágenes sobre el que desee aplicar la detección y la herramienta generará el conjunto de imágenes con los vehículos ya detectados.

Con la puesta en funcionamiento del TrafficDetector, se conseguirá disminuir las horas de trabajo mecánico y repetitivo que se requieren para realizar la detección manual de los vehículos. La

única acción manual estará enfocada hacia la supervisión de la herramienta. Esta reducción de horas en la detección permitirá al cliente destinar y reorientar recursos económicos y de mano de obra hacia otros sectores y áreas, ayudándole así a incrementar la productividad y el beneficio económico asociado.

1.3. Definiciones, acrónimos y abreviaturas

- *Deep Learning*: conjunto de técnicas que se enmarcan dentro del aprendizaje automático y que se inspiran en el funcionamiento de las redes neuronales cerebrales para contruir un modelo artificial que sea capaz de realizar predicciones con el mayor éxito posible.
- *Detección de objetos*: teconología computacional relacionada con la visión artificial y el procesamiento de imagen que trata de reconocer y localizar objetos en imágenes o en vídeos.
- *Dataset*: Colección de datos necesarios para diseñar y evaluar el algoritmo/modelo de procesamiento de datos, en este caso de procesamiento de imágenes.
- *Data augmentation*: Conjunto de técnicas utilizadas en el ámbito de análisis de datos que aumentan el tamaño de la base de datos a través de la modificación de los datos existentes convirtiendo estas transformaciones en nuevos datos que se añaden al conjunto original.

1.4. Descripción

Se ha habilitado un repositorio en github donde se mostrarán toda la documentación y resultados de TrafficDetector. A este repositorio se puede acceder haciendo doble click en el siguiente link [Repositorio Github](#).

1.5. Personal involucrado

Para la consecución de los objetivos propuestos se llevará a cabo la contratación de tres ingenieros cuyo ámbito de especialización sea la visión artificial. Cada uno de ellos se encargará de planificar los flujos de trabajo y de desarrollar el código necesario para la aplicación. Después evaluarán el modelo y colaborarán junto con el cliente en la evaluación de la aplicación y la propuesta de nuevas mejoras.

2. Descripción general

En esta sección se describen todos aquellos factores que afectan al producto y a sus requisitos. No se describen los requisitos, sino su contexto, ya que éstos se describen en la Sección 3.

2.1. Perspectiva del proyecto

El objetivo de este producto es desarrollar un sistema que sea capaz de detectar vehículos en imágenes. El cliente quiere que, dada una calle o una imagen, el sistema determine el número de vehículos que se encuentran en esa imagen o calle respectivamente.

Además del número de vehículos, se podrá ver la densidad de vehículos por tramo de carretera para poder estudiar el flujo de tráfico en distintas zonas de la ciudad.

Para abordar el sistema se utilizarán técnicas de Deep Learning, aunque este método puede cambiar más adelante. Se estudiará también la posibilidad de, en una versión posterior, diferenciar entre tipos de vehículos.

2.2. Funciones del producto

El sistema permitirá realizar las siguientes funciones:

- Introducir el nombre de una calle para ver su tráfico.
- Introducir una foto aérea como la mostrada en la Figura 1 para detectar los vehículos.
- Mostrar en la imagen un mapa de calor con el flujo del tráfico.
- Contar el número de vehículos que aparecen en esa calle.

En la Figura 2 se puede observar de manera esquematizada los posibles casos de uso de la aplicación.

2.3. Características de los usuarios

Este sistema será utilizado por ayuntamientos de distintas ciudades para poder realizar un estudio sobre el flujo de tráfico en distintos tramos de la ciudad y así poder buscar soluciones a problemas de movilidad existentes con mayor precisión. No se menciona si los usuarios del ayuntamiento precisan de conocimientos informáticos.

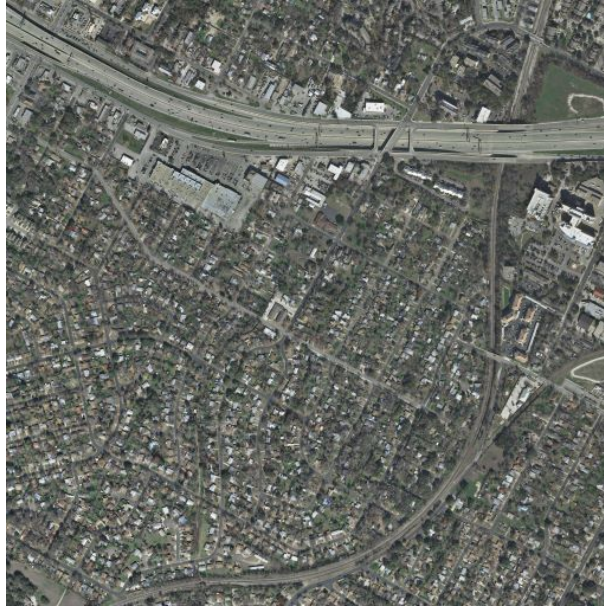


Figura 1: Ejemplo de foto en la que se hará la detección

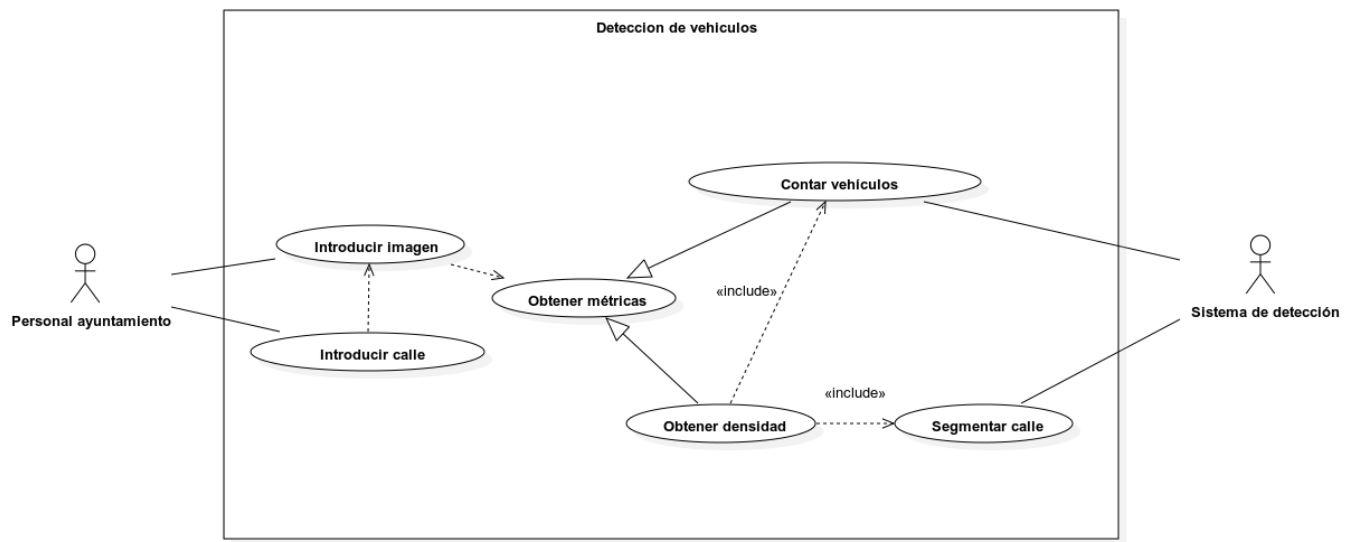


Figura 2: Caso de uso

2.4. Restricciones

- **Políticas reguladoras.** El programa debe de poder ser ejecutado desde un ordenador convencional.
- **Limitaciones de hardware.** Para ejecutar el programa se necesitará un servidor con un

procesador de unos 16 hilos y que cuente con tarjeta gráfica.

- **Lenguajes de programación.** Se utilizará Python como lenguaje de programación del sistema ya que cuenta con numerosas librerías de visión artificial y Deep Learning, como Tensorflow o Keras.
- **Requisitos de fiabilidad.** La información introducida en los formularios tiene que concordar el tipo de dato esperado en un determinado campo con el tipo de dato introducido.

2.5. Suposiciones y dependencias

- Se cuenta con un pequeño *dataset* proporcionado por el cliente de 16 imágenes de 5000x5000 píxeles de la ciudad de Austin. El número de imágenes es insuficiente, por lo que se ampliará el *dataset* con otras imágenes aéreas o utilizando *data augmentation*.
- La ubicación de las imágenes es conocida, y también la altura a la que fueron tomadas.
- Las imágenes se tomarán siempre en una misma franja horaria, sobre las 11 de la mañana.
- Se utilizará alguna herramienta como Open Street Map para identificar las calles que aparecen en una imagen.

3. Requisitos específicos

3.1. Requisitos funcionales

Código	Nombre	Fecha	Grado necesidad	
RF1	Seleccionar calle	10/03/2022	Media	
Descripción	Seleccionar el nombre de una calle y devolver la región de la calle en la fotografía			
Entradas	Fuente	Salida	Destino	Restricciones
Imagen aerea	Sistema de ficheros local	Región de la imagen	Sistema de ficheros local	Los mapas usados para obtener la región de interes deben de ser gratuitos y de uso libre
Nombre de la calle	Campo de texto			
Proceso	El usuario selecciona la imagen guardadas en el sistema de ficheros local y escribe el nombre de la calle. El sistema segmenta la región de la calle y guarda la imagen en el sistema de ficheros local.			
Efecto colateral	No aplica			

Código	Nombre	Fecha	Grado necesidad	
RF2	Calcular densidad de coches	10/03/2022	Alta	
Descripción	Dada la región de una calle, detectar y devolver el número y la densidad de coches.			
Entradas	Fuente	Salida	Destino	Restricciones
Segmento de imagen aérea	Sistema de ficheros local	Número de coches y densidad en coches/píxeles	Escritura del resultado dentro de la imagen	No aplica
Proceso	Se selecciona la región de una calle. El sistema detecta el número de coches en cada región. Se escribe dentro de la imagen el número de coches detectados.			
Efecto colateral	No aplica			

3.2. Requisitos no funcionales

Código	Nombre	Fecha	Grado necesidad	
RNF1	Máquina con python	10/03/2022	Alta	
Descripción	La aplicación debe de ejecutarse en una máquina con Python instalado. Dicha máquina debe de contar con la posibilidad de instalar paquetes de Python adicionales			
Entradas	Fuente	Salida	Destino	Restricciones
No aplica	No aplica	No aplica	No aplica	No aplica
Proceso	La instalación de la aplicación en la máquina puede ser hecha por una persona del equipo de desarrollo o bien por una persona escogida por el cliente con conocimientos técnicos de Python.			
Efecto colateral	No aplica			

Código	Nombre	Fecha	Grado necesidad	
RNF2	Ejecución offline	10/03/2022	Alta	
Descripción	La aplicación procesa las imágenes y la densidad de coches sin requisitos de tiempo real. No habrá restricciones de tiempo de ejecución máximo.			
Entradas	Fuente	Salida	Destino	Restricciones
No aplica	No aplica	No aplica	No aplica	No aplica
Proceso	No aplica			
Efecto colateral	No aplica			

Código	Nombre	Fecha	Grado necesidad	
RNF3	Mapas de uso libre	10/03/2022	Alta	
Descripción	Los mapas con los que trabajará el sistema son de uso libre y gratuito			
Entradas	Fuente	Salida	Destino	Restricciones
No aplica	OpenStreetMaps	No aplica	No aplica	No aplica
Proceso	No aplica			
Efecto colateral	No aplica			

Código	Nombre	Fecha	Grado necesidad	
RNF4	Imágenes aéreas	10/03/2022	Alta	
Descripción	Cada imagen aérea cuenta con la información de: Latitud, Longitud, Hora del día, Tamaño			
Entradas	Fuente	Salida	Destino	Restricciones
No aplica	Cliente	No aplica	Sistema de ficheros local a la aplicación	Para la primera versión solo se tratan imágenes de Austin.
Proceso	No aplica			
Efecto colateral	No aplica			