Proyecto: Detección de árboles en imágenes aéreas

Documento de Requisitos del Sistema (DRS)

Autores: Pablo Asensio Martínez

Vanesa Lomas García

Modelo: IEEE/ANSI 830-1998

Índice

[1 Introducción 2](#__RefHeading___Toc416_3441010062)

[1.1 Propósito del documento 2](#__RefHeading___Toc418_3441010062)

[1.2 Alcance del producto 2](#__RefHeading___Toc420_3441010062)

[1.3 Definiciones, acrónimos y abreviaturas 2](#__RefHeading___Toc422_3441010062)

[1.4 Referencias 2](#__RefHeading___Toc424_3441010062)

[1.5 Descripción del resto del documento 2](#__RefHeading___Toc426_3441010062)

[2 Descripción general 3](#__RefHeading___Toc428_3441010062)

[2.1 Perspectiva del producto 3](#__RefHeading___Toc430_3441010062)

[2.2 Funciones del producto 3](#__RefHeading___Toc432_3441010062)

[2.3 Características del usuario 3](#__RefHeading___Toc434_3441010062)

[2.4 Restricciones generales 3](#__RefHeading___Toc436_3441010062)

[2.5 Suposiciones y dependencias 3](#__RefHeading___Toc438_3441010062)

[3 Requerimientos específicos 4](#__RefHeading___Toc440_3441010062)

[4 Apéndice 5](#__RefHeading___Toc442_3441010062)

[5 Índices 6](#__RefHeading___Toc1001_3441010062)

[5.1 Palabras clave 6](#__RefHeading___Toc1003_3441010062)

[5.2 Bibliografía 6](#__RefHeading___Toc1005_3441010062)

[5.3 Webs 6](#__RefHeading___Toc1007_3441010062)

[5.4 Empresas 6](#__RefHeading___Toc1009_3441010062)

# Introducción

En esta sección se proporcionará una introducción a todo el documento. Consta de varias subsecciones: propósito, ámbito del sistema, definiciones, referencias y visión general del documento.

## Propósito del documento

Documento dirigido al cliente José Vélez Serrano y a la dirección de Blind Visión. El objetivo del documento es fijar los requisitos del sistema impuestos por el cliente y explicar la manera en que se va a abordar el problema.

## Alcance del sistema

Se desea crear una aplicación que, pasándole una imagen aérea, detecte el número de árboles que hay. En adelante, se denominará a dicha aplicación como TreeDetector.

TreeDetector será capaz de realizar un conteo de los árboles presentes en una imagen aérea y de localizarlos con una bounding box. La aplicación deberá poder utilizarse en cualquier estación del año, lo que implica que se deberán reconocer los árboles en cualquiera de los estados por los que pasan a lo largo de un año (con hojas y sin hojas).

Con esta aplicación se podrán realizar estudios de la cantidad de vegetación de hay en una determinada zona, o de forma más generalizada, en una determinada ciudad. Estos estudios podrían ser muy útiles para, por ejemplo, realizar análisis de la calidad del aire.

## Definiciones, acrónimos y abreviaturas

* **Dataset.** Hace referencia al conjunto de datos con el que se trabajará, en nuestro caso las imágenes aéreas.
* **Bounding Box.** Rectángulo que viene determinado por cuatro coordenadas(x, y, anchura, altura) y que se dibujará en las imágenes para indicar la posición de los árboles.
* **Deep Learning (en español, aprendizaje automático).** Es un conjunto de algoritmos de aprendizaje automático que utilizan redes neuronales para dar solución a problemas de por ejemplo clasificación o detección.
* **Red neuronal.** Son un modelo inspirado en el funcionamiento del cerebro humano, formado por un conjunto de nodos llamados neuronas que están conectadas y trasmiten señales entre sí. Estas neuronas se organizan en capas y su objetivo es el de ‘aprender’ a realizar una tarea concreta, en este caso, detectar árboles en imágenes.
* **Data augmentation.** Técnica que permite aumentar el tamaño y diversidad de un dataset por medio de perturbaciones en los datos originales (rotaciones, traslaciones…). Esto hará que se mejorar la precisión y la generalización del modelo.

## Referencias

* Aplicación LabelImg para etiquetar las imágenes de muestra. <https://github.com/tzutalin/labelImg>

## Descripción del resto del documento

Breve resumen del resto del documento.

# 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. Esto permitirá definir con detalle los requisitos en la sección 3, haciendo que sean más fáciles de entender.

## Perspectiva del producto

El objetivo es desarrollar un programa en Python que sea capaz de detectar árboles en imágenes aéreas. Dicho programa de Python deberá poder llamarse desde Java puesto que el cliente lo que quiere es una aplicación en Java (llamada a web).

La aplicación inicialmente será diseñada para detectar árboles, pero en un futuro podría seguir desarrollándose para que fuese capaz de detectar arbustos o jardines. También se podría ampliar haciendo que realice la detección sobre otro tipo de imágenes a mayor escala, como pueden ser imágenes tomadas por un dron.

## Funciones del producto

La aplicación recibirá una imagen aérea, como la mostrada en la figura X, y sobre ella se realizará la detección de los árboles.

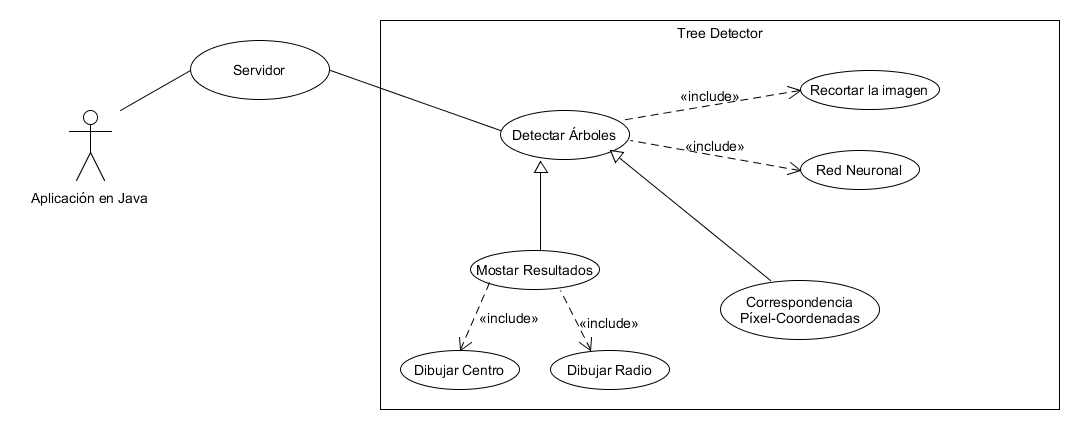
D:\Máster-Visión Artificial\2O-CUATRI\1-Aplicaciones Industriales y Comerciales\Proyecto_Deteccion_Arboles\AIVA_2021-imagenes_aereas\images\5000\austin1.tif

Cabe mencionar que las imágenes aéreas con las que se trabajará tiene una alta dimensionalidad por lo que será necesario dividirlas en imágenes más pequeñas sobre las que se realice la detección de árboles.

La salida que se obtendrá será la imagen recibida sobre la que se habrán señalado la posición de los árboles detectados con una circunferencia y su correspondiente centro. Un ejemplo del resultado que se obtendrá sobre una pequeña región de la imagen original se muestra en la Figura X.

D:\Máster-Visión Artificial\2O-CUATRI\1-Aplicaciones Industriales y Comerciales\Proyecto_Deteccion_Arboles\AIVA_2021-imagenes_aereas\images\peticion\austin24750_4750.tif

A continuación se muestra un diagrama de casos de uso que muestra de manera simplificada el funcionamiento de la aplicación que se va a diseñar:



## Características del usuario

No se nos ha indicado el perfil del usuario??

## Restricciones generales

Descripción de alto nivel de los requisitos nos funcionales.

* Lenguaje de programación. Se utilizará Python como lenguaje de programación ya que cuenta con librerías muy potentes para trabajar en tareas de visión artificial como es OpenCV y con otras librerías como TensorFlow y Keras que permiten desarrollar modelos de Deep Learning de forma sencilla.
* Limitaciones del hardware. Se cuenta con ordenadores equipados con tarjetas gráficas que son suficientes para desarrollar la aplicación.
* Protocolos de comunicación. Llamadas al programa de Python desde Java.
* Criticidad de la aplicación. Se tendrá que tener en cuenta que a veces reconocer árboles en imágenes aéreas no es tarea fácil ni siquiera para el ojo humano. Dos casos que se deben tener cuenta y en los que la detección puede no ser tan precisa como debería son el caso de árboles de tronco fino y sin hojas, y el caso en el que haya que etiquetar como diferentes dos árboles que están muy juntos. En esos casos la detección se complica.

## Suposiciones y dependencias

* Para el desarrollo de la aplicación se cuenta con 2 imágenes de dimensiones 5000x5000 proporcionadas por el cliente. Adicionalmente será necesario ampliar el *dataset* con imágenes que se recopilarán de la vista aérea de Google Maps. Otra forma de aumentar la cantidad de muestra es utilizar *data augmentation.*
* Las imágenes con las que se trabajará deberán tener una resolución suficiente para que los árboles puedan ser identificados por el ojo humano a la hora de hacerles zoom y etiquetarlas.
* La aplicación se desarrollará con la versión de Python 3.7, la versión para GPU de TensorFlow 2.2.0, Keras 2.3.1, CUDA 10.1, cuDNN 7.6 y la última versión de OpenCV.

# Requisitos específicos

Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que los satisfaga y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos.

## Funciones

Punto principal que detalla requisitos funcionales y no funcionales.

Numerar los requisitos.

Decir de dónde se deduce el requisito.

No poner requisitos que no reflejen necesidades reales.

Hacer que los requisitos sean:

* Consistentes, no deben contradecirse.
* Completos, conteniendo todo lo propuesto por el usuario que sea válido, y justificando lo que no lo sea.
* Realistas, verificando que se podrá construir.
* Verificable, el cumplimiento de los requisitos debe poderse medir.

## Rendimiento

* Carga de trabajo que se espera.
* Frecuencia de acceso a la red o a bases de datos
* Tasa de acierto/error que espera/acepta el cliente.
* Velocidad de respuesta que espera/acepta el cliente.

## Restricciones de diseño

* Necesidad de apoyo de expertos para realizar el desarrollo
* Necesidad de formación de los desarrolladores
* Limitaciones tecnológicas
* Restricciones de tiempo de desarrollo
* Restricciones de presupuesto
* Restricciones sobre el uso de paradigmas de programación

## Atributos del sistema

* Fiabilidad frente a caídas del sistema.
* Portabilidad del desarrollo a otros entornos hardware o software.
* Seguridad del sistema frente a ataques malintencionados.
* Mantenibilidad del sistema con el paso del tiempo.

## Otros requisitos

Cualquier requisito que no esté especificado antes.

# Apéndice

Detalla elementos adiciones y sus características: hardware, bases de datos…

Formatos de entrada/salida de datos, por pantalla o en listados.