

Proyecto AfluenceCounter

Documento de Requisitos del Sistema (DRS)

Versión del documento 0.1.0

Autor: Israel Peñalver,

David Valladares,

Alex Darío Cevallos

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.....	2
1.4	Referencias.....	3
1.5	Descripción del resto del documento.....	3
2	Descripción general.....	4
2.1	Perspectiva del producto.....	4
2.2	Funciones del producto.....	4
2.3	Características del usuario.....	5
2.4	Restricciones generales.....	5
2.5	Suposiciones y dependencias.....	5
3	Requisitos específicos.....	7
3.1	Funciones.....	7
3.2	Rendimiento.....	7
3.3	Restricciones de diseño.....	7
3.4	Atributos del sistema.....	8

1 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.

1.1 Propósito del documento

Documento dirigido al centros comerciales y comercios y a la dirección de SolVision. El objetivo es acotar el riesgo del proyecto analizando los requisitos.

1.2 Alcance del sistema

Se quiere crear una aplicación que contabilice el número de personas que pasan por delante de una tienda y cuantas de ellas acaban entrando. El nombre de la aplicación será AfluenteCounter.

AfluenteCounter podrá añadir nuevas funcionalidades en un futuro, pero inicialmente solo será capaz de contabilizar el número de personas que pasan por delante de la tienda y el número de personas que entran a la tienda, analizando las imágenes de la cámara que apunta frontalmente a la puerta y escaparate de la tienda.



Figura 1.- Ejemplo de un frame de un vídeo de muestra

1.3 Definiciones, acrónimos y abreviaturas

OpenCV.- Es una biblioteca libre de visión artificial originalmente desarrollada por Intel. OpenCV significa Open Computer Vision (Visión Artificial Abierta).

TensorFlow.- Es una plataforma de código abierto de extremo a extremo para el aprendizaje automático. Cuenta con un ecosistema integral y flexible de herramientas, bibliotecas y recursos de la comunidad que les permite a los investigadores impulsar un aprendizaje automático innovador y, a los desarrolladores, compilar e implementar con facilidad aplicaciones con tecnología de AA.

Keras.- Es una biblioteca de Redes Neuronales de Código Abierto escrita en Python. Es capaz de ejecutarse sobre TensorFlow. Está especialmente diseñada para posibilitar la experimentación en más o menos poco tiempo con redes de Aprendizaje Profundo. Sus fuertes se centran en ser amigable para el usuario, modular y extensible.

1.4 Referencias

- Aplicación VGG para etiquetar muestra [[web](#)]
- Aprendizaje Profundo [[web](#)]

1.5 Descripción del resto del documento

En la siguiente sección se describen todos aquellos factores que afectan al producto y a sus requisitos. Finalmente, en la sección 3 se describen los requisitos con más detalle.

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

2.1 Perspectiva del producto

Se pretende desarrollar en Python una aplicación que permita hacer un balance entre las personas que entran en la tienda y aquellas que pasan de largo.

Esta aplicación utilizará la biblioteca de OpenCV, que permitirá leer los videos y pre-procesarlos, y bibliotecas de aprendizaje profundo (Keras, TensorFlow), para realizar la inferencia de las personas que entran en la tienda o no.

2.2 Funciones del producto

La aplicación recibirá un vídeo en color o en escala de grises, en el cual aparecerá un flujo de personas.

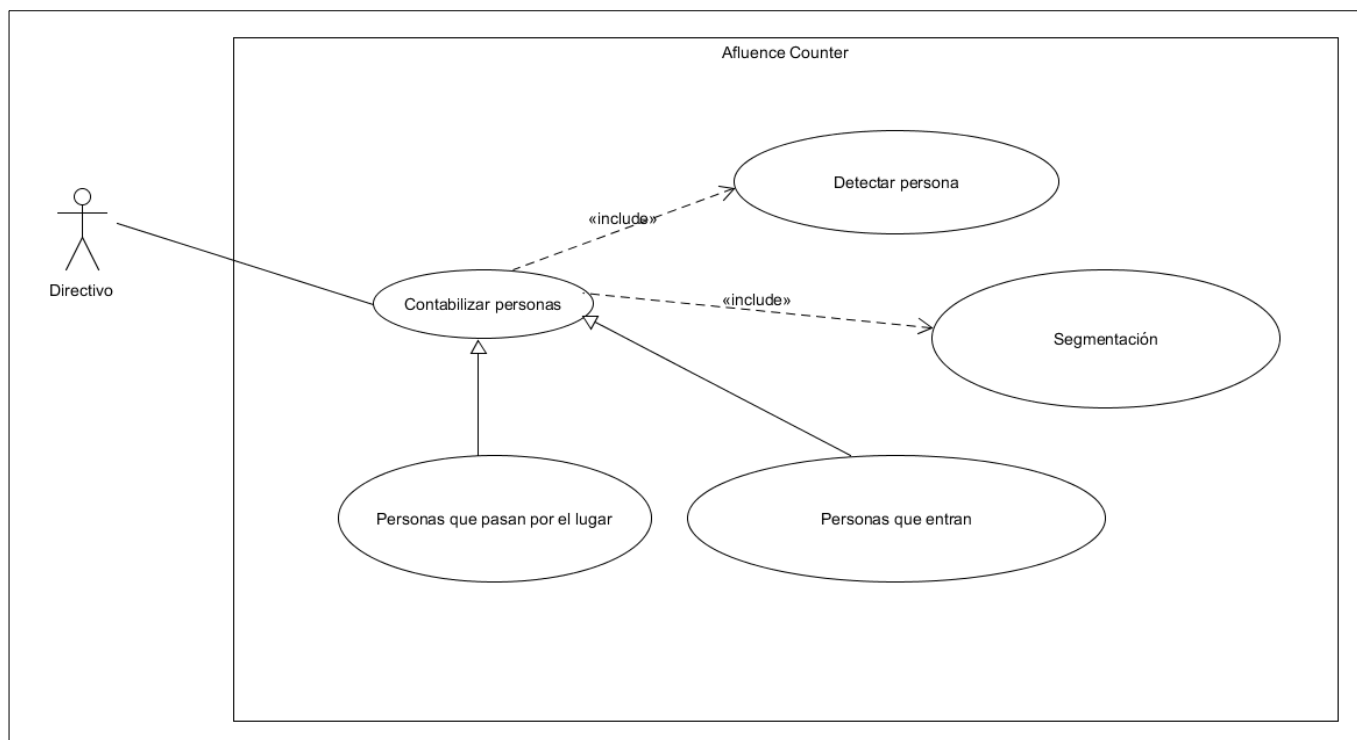


Figura 2.- Diagrama Casos de Uso.

2.3 Características del usuario

Un usuario utilizando la aplicación selecciona el vídeo a analizar y obtiene como resultado las medidas.

2.4 Restricciones generales

Descripción de alto nivel de los requisitos no funcionales.

- Lenguaje de programación.-Python. Dispone de bibliotecas que facilitarán la labor a desempeñar.
- Limitaciones del hardware. En principio una estación de trabajo de propósito general.
- Protocolos de comunicación. Llamadas a funciones desde Python
- Criticidad de la aplicación. Si no se reconocieran a personas entrando o pasando de largo se tendrá que añadir a mano en el documento generado por la aplicación.

2.5 Suposiciones y dependencias

- Para el desarrollo se dispondrán de unos 100 vídeos de prueba.
- Los vídeos deberán tener una resolución suficiente para permitir reconocer a personas y si están entrando en la tienda o no. Hay que tener en cuenta que no haya problemas de iluminación y que esta sea constante.
- Se desarrollará sobre las últimas versiones de OpenCV, Keras, TensorFlow y Python3.8.

3 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.

3.1 Funciones

1. Personas que entran en la tienda.

Origen: Desde una cámara de vídeo vigilancia el cliente pide detectar las personas que entran en la tienda.

Para desarrollarlo y testarlo se precisan 100 vídeos reales que contengan personas entrando en la tienda.

2. Personas que no entran en la tienda

Origen: Desde una cámara de vídeo vigilancia el cliente pide detectar a las personas que no entran a la tienda.

Para desarrollarlo y testarlo se precisan 100 vídeos reales que contengan personas que pasen por la tienda.

3. Contabilización.

Origen: Desde una cámara de vídeo vigilancia se contabilizan las personas que han sido detectadas entrando y pasando de largo.

Para desarrollarlo y testarlo se precisan 100 vídeos reales que contengan personas que pasen o no por la tienda.

3.2 Rendimiento

El sistema de detección de actividad de la tienda debe ejecutarse en menos de un segundo sobre una imagen utilizando un i5.

El cliente no aceptaría una tasa de error superior a 1 por cada 1000.

3.3 Restricciones de diseño

- Necesidad de formación al cliente sobre el uso de la aplicación.
- Se va a usar Programación Orientada a Objetos como en otras bibliotecas desarrolladas por EngenVision.
- El sistema no será capaz de reconocer aquellas personas cuya posición se encuentre fuera

del rango de visión en algún punto.

- El sistema no funcionará bien para vídeos de una resolución deficiente y en condiciones de mucha variabilidad de iluminación.

3.4 Atributos del sistema

- Portabilidad.- El desarrollo sobre Python asegurará su portabilidad a sistemas MAC/Windows/Linux.
- Mantenibilidad.- EL código desarrollado cumplirá las normas de estilo de la empresa (nomenclatura de variables y funciones, tamaño de las funciones, comentarios...).