



Detección de actividad en vídeos de tienda

Documento de diseño

Autores:

Jairo Carrillo Huélamo

Daniel Hernández Puerto

Águeda Sierra Carreto

Índice

1. Introducción	2
1.1. Introducción al desarrollo del proyecto	2
1.2. Descripción del funcionamiento de los test.....	2
1.3. Enlace al repositorio de GitHub	4
2. Diseño de la solución.....	5
2.1. Diagrama estático de clases.....	5
2.2. Finalidad y métodos de las clases	5
2.2.1. VideoDetection	5
2.2.2. FrameDetection	5
2.3. Diagrama de secuencia.....	6
2.4. Algoritmos desarrollados	7
2.4.1. Hacer la detección de las personas	8
2.4.2. Contar el número de personas que entran en la tienda.....	9
2.4.3. Contar el número de personas que pasan de largo.....	10
2.4.4. Contar el número de personas que se paran	11

1. Introducción

En este apartado se comenzará con una breve introducción al desarrollo del proyecto, seguida de unas cuantas imágenes de test en las que se ve el funcionamiento del algoritmo y por último un enlace al repositorio de GitHub.

1.1. Introducción al desarrollo del proyecto

Para el desarrollo del proyecto se ha partido de una red YOLOv5 preentrenada (<https://github.com/ultralytics/yolov5/archive/master.zip>).

Una vez descargada la red con los pesos correspondientes, se pasará a la detección de personas. Esta parte está dividida en dos, ya que el proyecto tiene dos clases: *VideoDetection* y *FrameDetection*.

La clase *FrameDetection* se encargará del reconocimiento de personas en un *frame* y devolverá una lista de *bounding boxes* que aparezcan en ese *frame*.

La clase *VideoDetection* se encargará de leer el vídeo, dibujar los *bounding boxes* de las personas y sacar los resultados.

Primero se abrirá el vídeo para leer y el vídeo de salida para generar el vídeo final con la detección. Hasta que se finalice el vídeo se irá leyendo *frame* a *frame* y se harán las siguientes operaciones. Por cada *frame* se utilizará la red YOLOv5 y se obtendrán los resultados de esa detección. Con esos resultados se pintarán en el frame los *bounding boxes* de las posibles personas que aparezcan. Una vez hecha la detección, se contará el número de personas que entran en la tienda, las que pasan de largo y las que se paran. Para ello se utilizarán varios métodos de la clase *FrameDetection* que se explicarán más adelante. También se detallará la estrategia seguida para hacer ese conteo.

Para desarrollar los test unitarios se han tenido en cuenta las siguientes posibilidades, que el vídeo se lea, que se detecten personas, cuántas personas entran, esperan y pasan de largo. Para la ejecución de los test se introducirá el vídeo de prueba, se le aplicará el algoritmo y se comprobarán esos 5 test.

1.2. Descripción del funcionamiento de los test

En este apartado se procederá a explicar el funcionamiento de los test unitarios del proyecto.

El proyecto consta de 5 test que debe pasar el vídeo, de los cuales dos son obligatorios. Estos dos son la lectura del vídeo y que haya reconocido alguna persona en el vídeo si la hay. En la Figura 1 se puede ver un ejemplo del reconocimiento de dos personas.



Figura 1: Muestra de detección de personas

El funcionamiento de los test es el siguiente, se introduce la dirección local de un vídeo que se quiera analizar. Se deberán introducir los parámetros de *groundtruth* que se espera que vaya a reconocer el vídeo. Una vez hecho esto se ejecutarán los test y se verá el resultado del algoritmo aplicado en el vídeo.

Para hacer saber cuántas personas entran, se paran o pasan de largo y compararlo con el *groundtruth* se han definido unas líneas para cada contador. En la Figura 2 se puede observar esta representación. La línea negra corresponde al contador de personas que entran, las líneas rojas para las personas que pasan de largo y las azules para las que se paran. Más adelante se explicará detenidamente cómo se hace esta detección mediante diagramas de actividad.



Figura 2: Líneas de referencia para los contadores de los distintos grupos

1.3. Enlace al repositorio de GitHub

Toda la información asociada al proyecto se encuentra en el siguiente repositorio de GitHub: https://github.com/redom69/AIVA_2022-Grupo-G

2. Diseño de la solución

En este apartado se explicará el diseño de la solución mediante una serie de diagramas que facilitarán su comprensión.

2.1. Diagrama estático de clases

En el diagrama de clases representado en la Figura 3 se representan las dos clases que hacen funcionar el sistema.

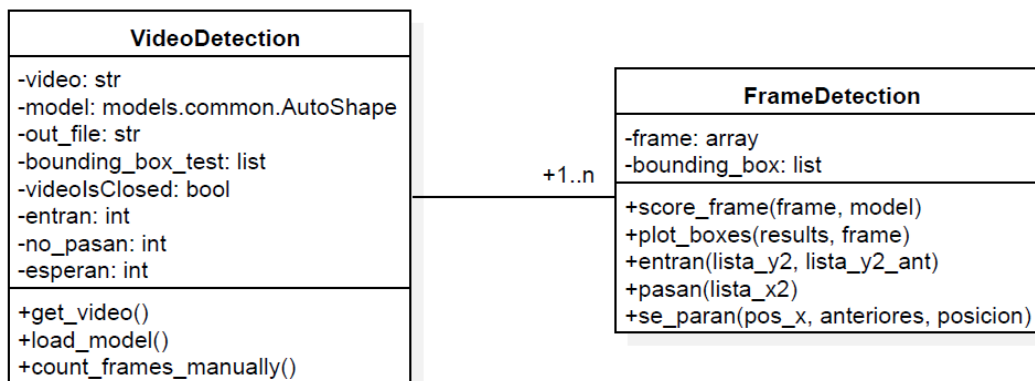


Figura 3: Diagrama estático de clases

2.2. Finalidad y métodos de las clases

En este apartado se explicarán las clases representadas en el diagrama anterior.

2.2.1. VideoDetection

La finalidad de esta clase es hacer la detección y el conteo de los diferentes grupos de personas en el vídeo completo. Para ello se utilizarán los resultados ofrecidos por la clase *FrameDetection* con el fin de dar un resultado global.

Los métodos más importantes para conseguir esta tarea son *get_video* y *load_model*. Con la primera sirve para leer el vídeo a analizar y la segunda para cargar los pesos del modelo YOLOv5 para llevar a cabo la detección.

2.2.2. FrameDetection

Esta clase sirve para hacer la detección y el conteo de los diferentes grupos de personas en un *frame*.

Los métodos más importantes son *score_frame* que utiliza el modelo YOLOv5 para hacer la detección, *plot_boxes* para representar los *bounding boxes* en el *frame*, *entran*,

pasan y *se_paran* para contar el número de personas que entran en la tienda, pasan de largo y se paran respectivamente.

2.3. Diagrama de secuencia

En la Figura 4 se representa el diagrama de secuencia del sistema.

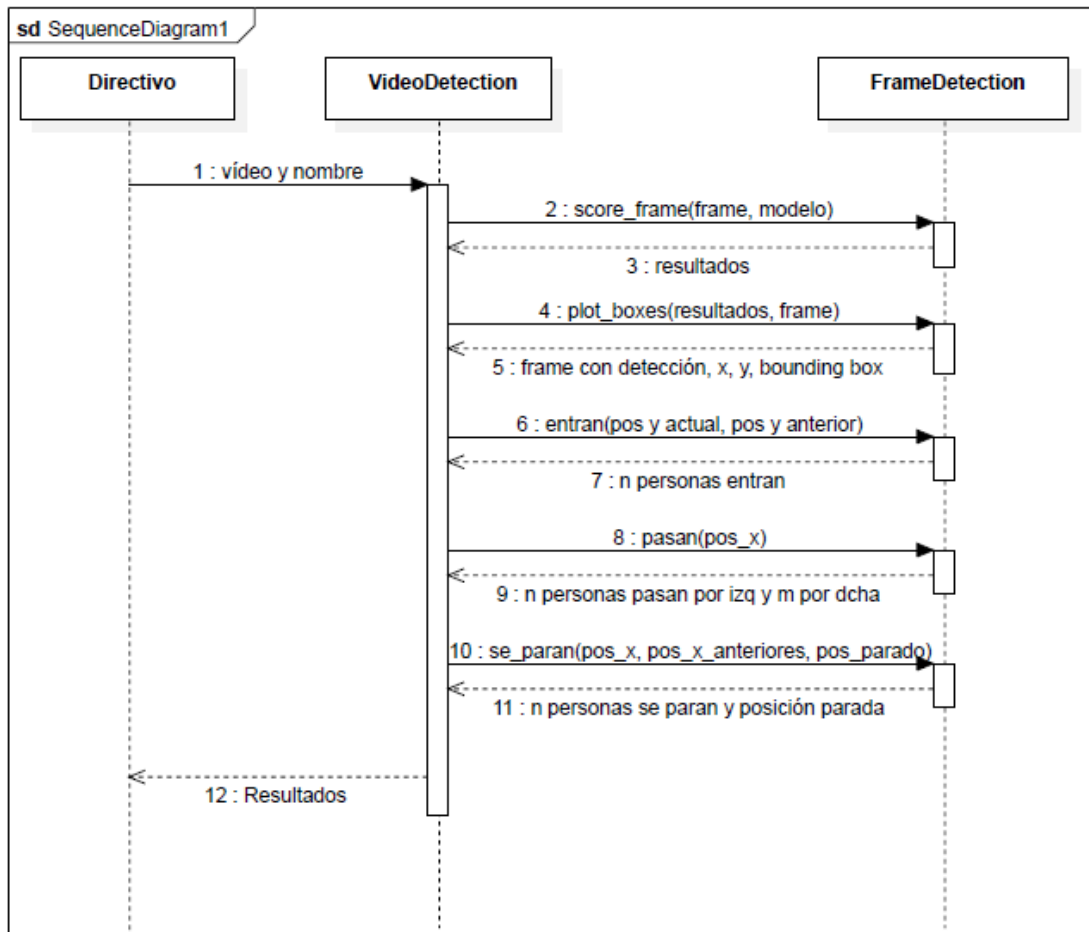


Figura 4: Diagrama de secuencia

En el diagrama de la Figura 4 hay dos clases: *VideoDetection* (VD) y *FrameDetection* (FD). Al principio del todo aparece el *Directivo* que representa al usuario del sistema, encargado de elegir los vídeos que se quieran analizar y el nombre del vídeo resultante de la detección.

Después, VD le pasa el *frame* a analizar y el modelo de YOLOv5 al método *score_frame* de FD para hacer la detección de las personas. Los resultados devueltos por este método se pasan junto con el *frame* al método *plot_boxes* de FD. De esta manera se obtiene el *frame* con la detección hecha, los *bounding boxes* y sus coordenadas de la esquina inferior derecha.

Una vez obtenida la detección, VD utiliza el método *entran* de FD al que le pasa la coordenada *y* del *bounding box* actual y el anterior. Así se obtiene el número de personas que entran en la tienda en un *frame*. Después VD emplea el método *pasan* para calcular el número de personas que pasan en un *frame* por las líneas definidas. Por último, para contabilizar el número de personas que se paran en un *frame*, VD utiliza el método de FD *se_paran* al que le pasa la coordenada *x* del *bounding box* del *frame* actual, la correspondiente a los 10 anteriores y la coordenada *x* si se ha incrementado el contador.

Con todo lo obtenido en todos los *frames* en VD se obtiene el resultado de los contadores de todo el vídeo y se devuelve al usuario.

2.4. Algoritmos desarrollados

En este apartado se comentarán los principales algoritmos del sistema. Se han considerado cuatro algoritmos principales: la detección de las personas, contar el número de personas que entran en la tienda, el número de personas que pasan de largo y el número de personas que se paran. Cada uno de estos algoritmos contará con su diagrama de secuencia correspondiente para facilitar su comprensión. La explicación se hará sobre un *frame* y para un *bounding box*.

2.4.1. Hacer la detección de las personas

En la Figura 5 se representa el diagrama asociado a esta tarea.

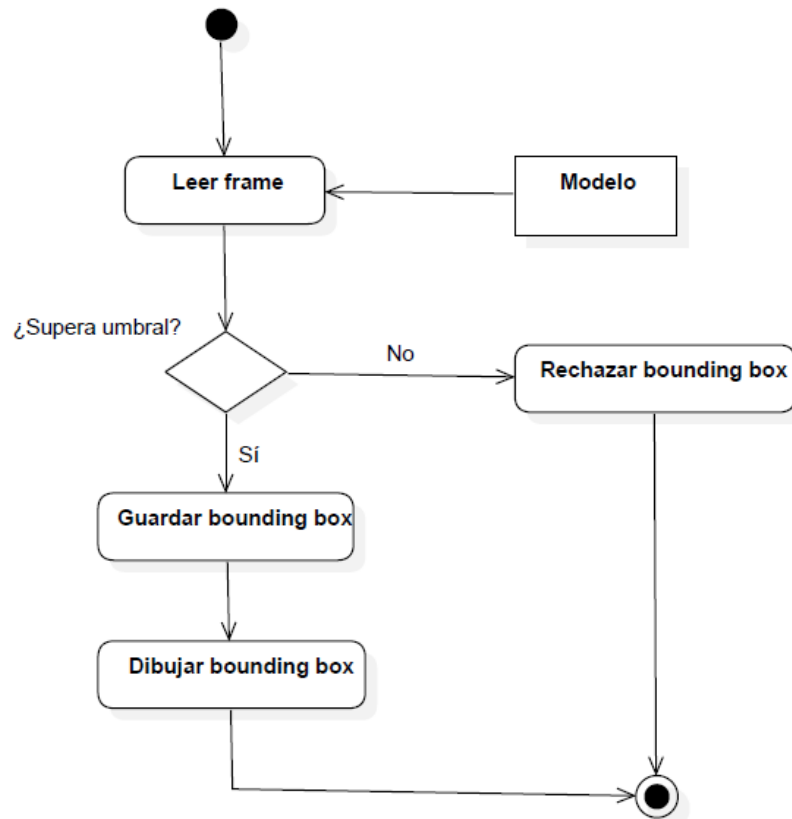


Figura 5: Diagrama de actividad para detección de personas

Para hacer la detección se ha utilizado el modelo de una red YOLOv5 preentrenada. Por tanto, una vez leído el *frame*, lo primero es pasarle el modelo para hacer la detección. Después, la detección obtenida se compara con un umbral. Si es menor se rechaza el *bounding box* y si es mayor se guarda y se dibuja en el *frame*.

Esta detección permite contar el número de personas que pertenecen a cada uno de los grupos comentados anteriormente.

2.4.2. Contar el número de personas que entran en la tienda

En la Figura 6 se puede observar el diagrama que explica esta tarea.

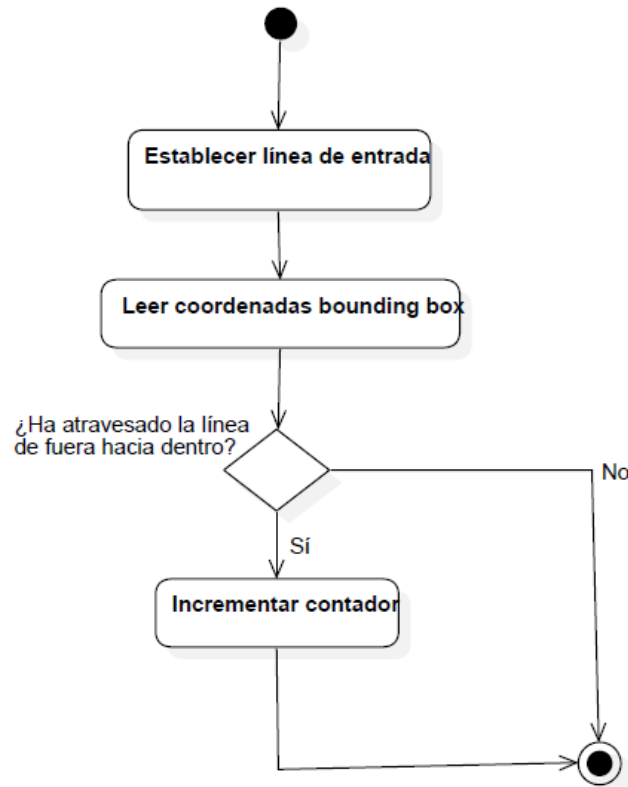


Figura 6: Diagrama de actividad para contar el número de personas que entran en la tienda

Para contabilizar el número de personas que entran en la tienda, se ha definido una línea en la puerta de la tienda, pero desplazada ligeramente hacia dentro (línea negra de la Figura 2). De esta manera, comparando el valor de la coordenada y de la esquina inferior derecha del *bounding box* con el umbral establecido por la línea se puede saber si la persona está sobre la línea o no.

Sin embargo, si solo se mira si la coordenada y es igual al umbral, se cuentan tanto las personas que entran como las que salen. Debido a esto, se mira si la persona ha llegado a la línea desde fuera o desde dentro. Para ello, se comprueba si la posición y en el *frame* anterior es mayor, lo que implica que la persona está entrando a la tienda.

Al final se obtiene el número de personas que entran en un *frame*, después hay que hacer la suma de todas las detecciones hechas para obtener el total.

2.4.3. Contar el número de personas que pasan de largo

El diagrama asociado a esta tarea se representa en la Figura 7.

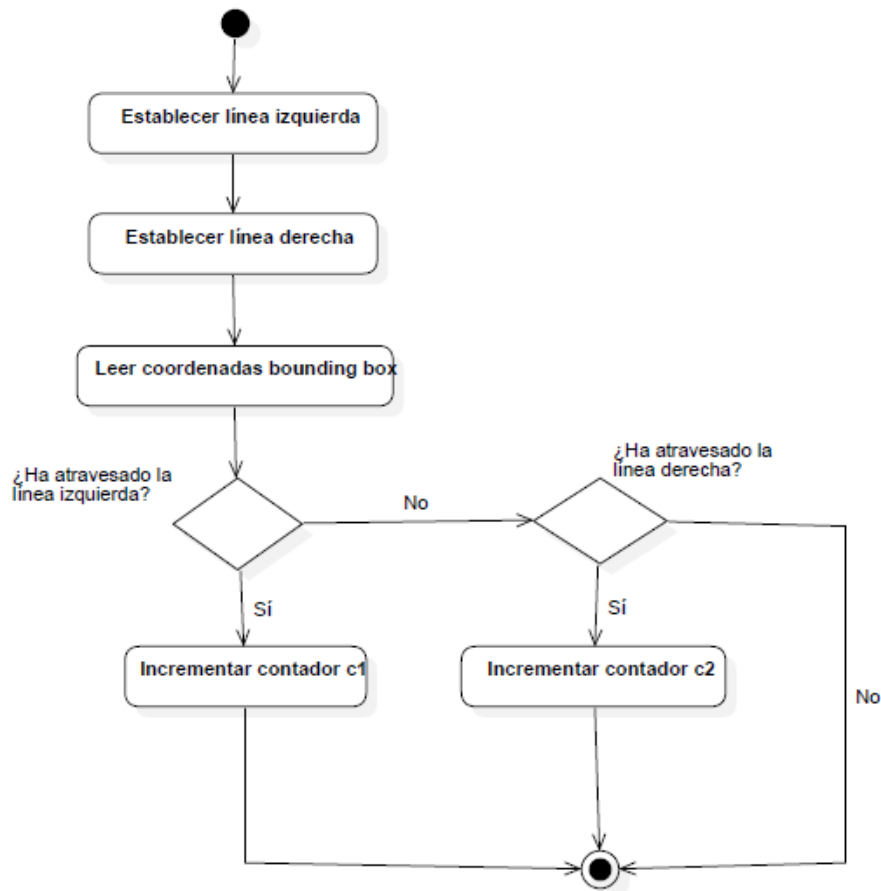


Figura 7: Diagrama de actividad para contar el número de personas que pasan de largo

En este caso se han definido dos líneas: una a la izquierda de la imagen y otra a la derecha (líneas rojas de la Figura 2). Al igual que en el caso anterior, se compara la coordenada del *bounding box* con el umbral definido por las líneas. Sin embargo, en este caso no se usa la coordenada y sino la *x*.

Con esto lo que se obtiene es el número de personas que están sobre ambas líneas en cada *frame*. Al igual que en el caso anterior, hay que sumar todas las detecciones para tener el resultado final. No obstante, hay que hacer un paso más y es coger el mínimo de ambos contadores para quedarse solo con el número de personas que ha pasado por encima de las dos líneas.

2.4.4. Contar el número de personas que se paran

El diagrama correspondiente a esta tarea se representa en la Figura 8.

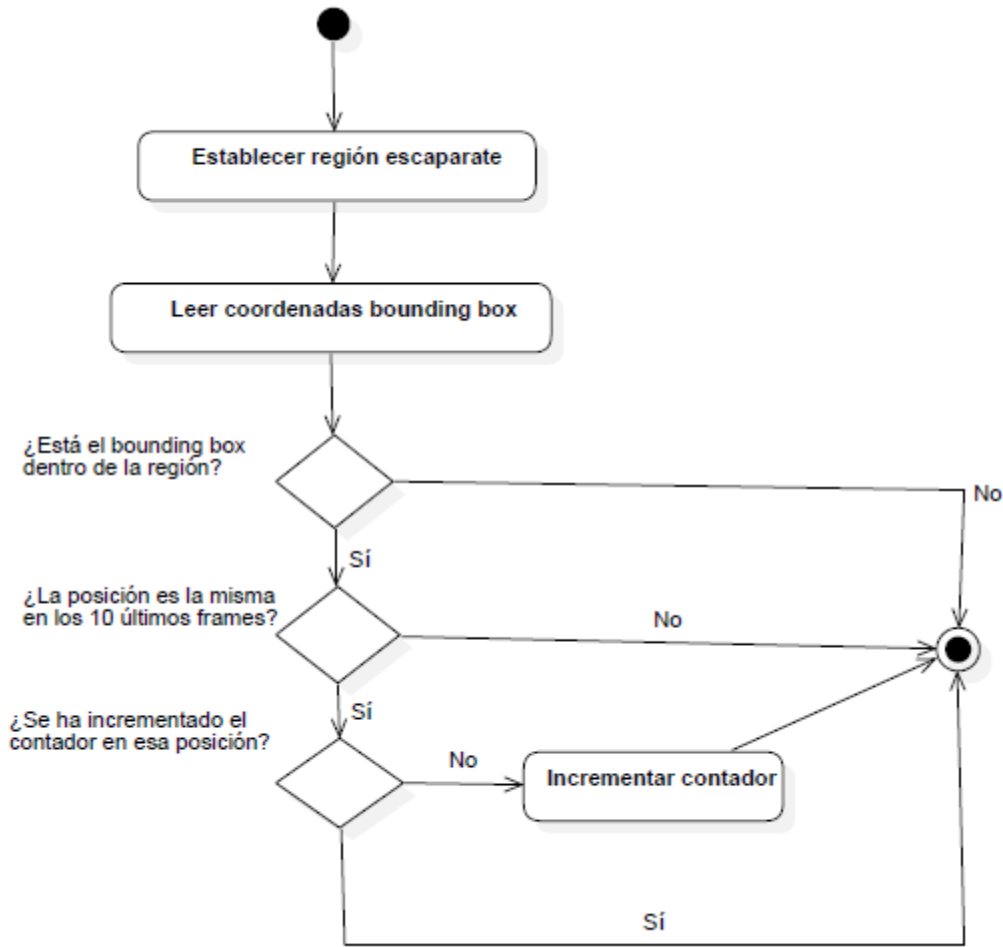


Figura 8: Diagrama de actividad para contar el número de personas que se paran

En este caso también se define una región de detección, pero menor que en el contador anterior, solo incluye la zona del escaparate (representada en la Figura 2 con las líneas azules). De la misma manera se compara la coordenada x con los umbrales de las líneas.

Para este contador, además, se comprueba si la posición es la misma durante los últimos 10 *frames*. Si se da esta condición es que la persona se ha parado. No obstante, hay que tener en cuenta si en esa posición ya se ha aumentado el contador anteriormente. De esta manera se evita contar varias veces a la misma persona si se para mucho tiempo.