

Identificación de horarios óptimos para almorzar en el comedor de la UPCH mediante el conteo de sillas ocupadas y desocupadas con YOLO

1. Integrantes del equipo

- Mauricio Camino Quintanilla
- Vanesa Morales Taipe
- Luis Arenas Torres
- Sophia Escalante Rodriguez

2. Justificación del problema

El comedor de la Universidad Peruana Cayetano Heredia (UPCH) es un espacio muy concurrido por estudiantes, profesores y personal administrativo. En horas pico, encontrar un asiento libre puede ser un desafío, generando pérdida de tiempo y frustración en los usuarios. Para optimizar el mejor momento para ir a almorzar, se propone implementar un modelo de YOLO para calcular la *occupancy* (proporción de sillas ocupadas) en tiempo real usando una cámara en el comedor universitario de la UPCH. Esto permitirá a los usuarios conocer la disponibilidad de asientos y planificar mejor su tiempo de almuerzo, mejorando así la experiencia general en el comedor.

3. Antecedentes

En el estudio "A Computer Vision-Based Proposal for Seat Occupancy Monitoring Applied to FEUP's Library" (Velo, 2021), se desarrolló un sistema de detección de *occupancy* de asientos en bibliotecas utilizando una cámara enfocada en el área donde se encuentran las sillas. Como la visión de la cámara es estática, se tomó una imagen estática del área, y luego se identificaron manualmente las áreas donde se encuentran las sillas, dibujando *bounding boxes*. Esto se hizo una sola vez al configurar el sistema y no es parte del proceso de detección automatizado continuo. Durante el monitoreo en tiempo real, el sistema emplea el modelo SSD300 acoplado con MobileNet para detectar personas en las áreas predefinidas con *bounding boxes*; se capturan fotogramas periódicos, se redimensionan a 300x300 píxeles, se normalizan mediante sustracción de la media y se procesan a través de la red neuronal. Si se detecta una persona, el estado del área se actualiza a "ocupado"; si no se detecta ninguna persona después de un límite de no detecciones, el estado se cambia a "vacío".

En el estudio "A serial dual-channel library occupancy detection system based on Faster RCNN" (Yang, Chang, Wang, Yang, & Chen, 2023), se desarrolló un sistema de detección de *occupancy* de asientos en bibliotecas. Primero se capturaron imágenes de la biblioteca y se segmentaron en áreas que corresponden a los asientos. Luego, utilizando una Red de Propuestas de Regiones (RPN) del algoritmo Faster R-CNN, se identifican y generan "cajas propuestas" alrededor de las áreas donde se encuentran las

sillas. Estas cajas propuestas se analizan mediante la red VGG19 preentrenada, para clasificar los contenidos y determinar si hay personas dentro de las “cajas propuestas”. Posteriormente, se utiliza la red AlexNet para clasificar los objetos dentro de las cajas propuestas. Si la VGG19 detecta la presencia de personas dentro de las cajas propuestas, se considera que el asiento está ocupado por una persona. Si la AlexNet detecta solo objetos como libros o mochilas y no personas, se concluye que el asiento está ocupado por pertenencias personales. El dataset utilizado para entrenar las redes incluye tanto imágenes reales capturadas en diversas bibliotecas como imágenes virtuales generadas con Unreal Engine 5.

En el proyecto “Empty Seat Detection in UTS Library with YOLOv5”, se desarrolló una aplicación de visión por computadora en tiempo real para detectar los asientos vacíos dentro de la biblioteca de la UTS. Se utilizó un dataset personalizado de 5200 imágenes capturadas desde cuatro ángulos diferentes y clasificadas en dos clases: “ocupado” y “vacío”. Este dataset se utilizó para entrenar el modelo YOLOv5 para poder realizar la tarea de detección de objetos de una sola etapa que puede realizar la localización de objetos y la clasificación de imágenes dentro de la misma red.

4. Planteamiento

Inicialmente, se utiliza el modelo YOLOv9, para identificar y localizar personas en cada fotograma del video. El modelo se carga al inicio del programa y se aplica a cada frame procesado. También, para evaluar la *occupancy* de asientos, se define un conjunto de bounding boxes estáticos que representan las ubicaciones de los asientos dentro del campo de visión de la cámara. Estos bounding boxes se especifican como coordenadas fijas en el código y se mantienen constantes durante todo el análisis.

Para asociar las personas detectadas con los asientos, se realizó lo siguiente:

- a. Cálculo de solapamientos y distancias:
 - i. Para cada persona detectada, se verifica el solapamiento con los bounding boxes de las sillas.
 - ii. Si existe solapamiento, se calcula la distancia euclidiana entre los puntos medios.
- b. Asignación de sillas con persona según distancia mínima:
 - i. Las sillas se ordenan según la distancia mínima a cualquier persona.
 - ii. Se asigna cada silla a la persona más cercana no asignada previamente.

Este procedimiento se realiza en cada frame. Por cada frame obtenemos una cantidad de sillas libres y con esa información podemos analizar la evolución de la *occupancy* a lo largo del tiempo, que será usado para identificar las horas óptimas para ir a almorzar basado en los asientos libres en ese momento dado.

5. Tecnologías implementadas

Se utilizó Roboflow para realizar los bounding boxes de las sillas y obtener más fácil las coordenadas de ellos. Para la detección de objetos se utilizó YOLOv9 y para el procesamiento de imágenes se utilizaron las librerías de OpenCV. Inicialmente se usaron los modelos yolov8s, yolov8m, yolov8x, yolov9m. Al usar YOLOv9 a diferencia de los demás, se redujo los boxes duplicados en una persona.

6. Bibliografía

- Veloso, J. M. S. (2021). A Computer Vision-Based Proposal for Seat Occupancy Monitoring Applied to FEUP's Library (Doctoral dissertation, Universidade do Porto (Portugal)).
<https://repositorio-aberto.up.pt/bitstream/10216/135353/2/486171.pdf>
- Yang, G., Chang, X., Wang, Z., Yang, M., & Chen, X. (2023). A serial dual-channel library occupancy detection system based on Faster RCNN. arXiv preprint arXiv:2306.16080.
<https://arxiv.org/pdf/2306.16080>
- Kneex. (2021). EmptySeatDetection .
<https://github.com/kneex/EmptySeatDetection>