REVISTA ESTUDIOS DE TRANSPORTE Vol. 24, N° 1, 1-16 (2023)

USO DE VISIÓN ARTIFICIAL PARA CONTAR PASAJEROS QUE SUBEN Y BAJAN DEL TRANSPORTE PÚBLICO

Sergio Velastin[1, 2], Rodrigo Fernández[3] *

[1] School of Electronic Engineering and Computer Science, Queen Mary University of London, UK
[2] Departamento de Informática, Universidad Carlos III de Madrid, España
[3] Facultad de Ingeniería y Ciencias Aplicadas, Universidad de los Andes, Chile

*Autor para correspondencia:
rfa@miuandes.cl

RESUMEN

La principal fuente de demoras en los sistemas de transporte público tiene lugar en sus estaciones. Por ejemplo, un tren de metro puede viajar a 80 km por hora entre estaciones, pero su velocidad comercial – velocidad media incluidas las detenciones en las estaciones – alcanza menos de la mitad de ese valor. Luego, el problema que deben resolver los operadores de transporte público es reducir las demoras en las estaciones para aumentar la velocidad comercial. Esta depende del número de pasajeros que sube y baja de cada vehículo. Tradicionalmente, estos datos se han recopilado mediante conteos a mano o con videos que luego se procesan visualmente. Este trabajo presenta una forma de extraer datos mediante visión artificial desde videos obtenidos en un laboratorio a escala real y, a partir de estos, observar el efecto en la demora en estaciones. Parte de los videos se procesaron manualmente para entrenar y evaluar algoritmos de seguimiento y detección usando aprendizaje profundo. En más de 300 secuencias de video los resultados del conteo de pasajeros alcanzaron una exactitud promedio del 92% con una desviación estándar menor al 0,12%. Estos resultados implican una subestimación menor al 7% en el cálculo de la demora en estaciones.

Palabras clave: transporte público, estaciones, demoras, conteo de pasajeros, visión artificial, aprendizaje profundo

ABSTRACT

The main source of delays in public transport systems occurs at their stations. For example, a subway train can travel at 80 km per hour between stations, but its commercial speed – average speed including stops at stations – reaches less than half that value. Then, the problem that public transport operators must solve is to reduce delays at stations to increase commercial speed. This depends on the number of passengers getting on and off each vehicle. Traditionally, this data has been collected through hand counts or videos that are then visually processed. This work presents a way to extract data using artificial vision from videos obtained at a full-scale laboratory and, from these, observe the effect on the delay in stations. Part of the videos were manually processed to train and test tracking and detection algorithms using deep learning. In 322 video sequences, passenger counts achieve a mean accuracy of 92% with less than 0.12% standard deviation These results imply an underestimation of less than 7% in the calculation of delays at stations.

Keywords *public transport, stations, delays, passenger counting, computer vision, deep learning*

1. INTRODUCCIÓN

La mayoría de los sistemas de transporte público del mundo han invertido en sistemas de circuito cerrado de televisión (CCTV) para mejorar la seguridad tanto de tránsito como personal. Lo mismos medios se pueden usar para realizar mediciones para mejorar la calidad del servicio, pero las grandes cantidades de datos de video pueden generar una sobrecarga humana y una efectividad reducida.

Los sistemas de CCTV se han diseñado en gran medida para la observación humana. En las últimas dos décadas ha habido mejoras significativas en los sensores de las cámaras en términos de calidad y resolución de imagen, y en los medios digitales para transmitir y almacenar datos de video. No obstante, con la posible excepción del reconocimiento automático de placas patentes (ANPR) para el control de velocidad (p. ej., véase Ziółkowski, 2018) y para analizar los movimientos de pasajeros y carga, como lo discutieron Hadavi et al. (2020), el análisis de los datos de video de los sistemas de CCTV en las redes de transporte público se ha basado en gran medida en el análisis manual, principalmente debido a los desafíos que plantea la localización de personas, sus movimientos y, en última instancia, su comportamiento en escenarios realistas. Más recientemente, sin embargo, ha habido importantes mejoras científicas y tecnológicas en la visión computacional, lideradas por sensores inteligentes que utilizan métodos de aprendizaje profundo y que pueden ofrecer un camino a seguir. Por lo tanto, este artículo explora si dichos métodos pueden lograr resultados razonables en el análisis de personas que suben y bajan de un vehículo de transporte público, lo que permitiría a los investigadores del transporte realizar una mayor cantidad de observaciones en menos tiempo.

En este trabajo se llevaron a cabo experimentos de laboratorio a escala real en el *Pedestrian Accessibility Movement Environment Laboratory* (PAMELA) de University College London, para probar diseños de andenes y vehículos de transporte público (ver Figura 1). El aspecto original de este trabajo es la experimentación de laboratorio a escala real como método de investigación que tiene como objetivo evaluar las respuestas de los pasajeros ante diferentes configuraciones que luego pueden implementarse en vehículos y estaciones para mejorar sus operaciones. Las variables experimentales se basan en características de los sistemas de transporte público de Santiago, el metro de Londres y la literatura en este campo.

Figura 1 Laboratorio PAMELA configurado como un bus Transantiago

Los experimentos involucraron a personas comunes y corrientes que subieron y bajaron de una maqueta a escala real de un vehículo de transporte público para obtener conteos de personas y los tiempos necesarios para subir y bajar. Se estudiaron cuatro variables, siguiendo el trabajo anterior de Fernández et al. (2010): el método de cobro de tarifas, la distancia vertical entre el andén y el vehículo (brecha vertical), el ancho de las puertas y la densidad de pasajeros dentro del vehículo.

La evaluación de la detección se lleva a cabo utilizando un conjunto estándar y conocido de métricas, como las propuestas por Yin et al. (2010) y el *VOC Challenge* (Everingham et al., 2010). Para determinar cuándo las detecciones son verdadero positivo (VP), falso positivo (FP) y falso negativo (FN), se utiliza el coeficiente de similitud de Jaccard, también conocido como Intersección sobre Unión (IoU) del 50%. Por lo tanto, se produce un VP cuando un objeto de detección (D) tiene un IoU > 0,5 con un objeto verdadero o *ground truth object* (GT), o "verdad básica"; se produce un FP cuando una detección no tiene un IoU > 0,5 con ningún objeto GT; y se produce un FN cuando un objeto GT no tiene un objeto de detección correspondiente. Luego:

$$IoU = \frac{D \cap GT}{D \cup GT} \quad : Jacard \tag{1}$$

$$P = \frac{VP}{VP + FP} \quad : Precision \tag{2}$$

$$R = \frac{VP}{VP + FN} \quad : Recall \tag{3}$$

$$F1 = \frac{2PR}{P + R} \quad : F - measure \tag{4}$$

$$A = \frac{VP}{VP + FP + FN} \quad : Accuracy \tag{5}$$

donde VP es el número total de verdaderos positivos, FP el número total de falsos positivos y FN el número total de falsos negativos.

La Tabla 1 explica las definiciones de las variables. Notar que los casos verdaderos negativos (VN) no se consideran en las métricas.

Tabla 1 Definiciones de variables para la detección y conteo de personas

Verdad básica / **Algoritmo**	Hay una persona	No hay una persona
Detecta una persona	VP	FP
No detecta una persona	FN	VN

La precisión media o *mean average precision* (mAP) es la media de P sobre el rango de R, y también es una métrica comúnmente utilizada para problemas de detección y clasificación. Todas las medidas definidas en las ecuaciones 1 a 5 están en el rango (0, 1).

Por otro lado, en el pasado se han propuesto gran cantidad de métricas de seguimiento o rastreo de objetos (personas en nuestro caso). De aquí nace el MOTC (*Multiple Object Tracking Challenge*, (Lealtaixé et al., 2015). Por simplicidad, en este trabajo se usan P, R, F1 y MOTA (*Multi Object Tracking Accuracy*) definido por Bernardin y Stiefelhagen (2008), que trata de combinar varios errores que surgen en el seguimiento y se define como:

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \quad (6)$$

donde: m_t, fp_t y mme_t son el número de errores, falsos positivos y discrepancias, dividido por el número total de observaciones de la verdad básica g_t para el tiempo t, respectivamente. Cuanto mayor sea la puntuación, mejores serán los resultados.

En lo que resta del texto se entenderá por "vehículo" a cualquier vehículo de transporte público mayor (bus, metro, tren, BRT – Bus Rapid Transit) y por "estación" a cualquier infraestructura destinada a la espera, subida y bajada de pasajeros (paradero de bus, estación de metro, tren o BRT).

Este artículo está organizado de la siguiente manera. El Capítulo 2 contiene la revisión del estado del arte de la visión computacional aplicada a la detección de personas. El Capítulo 3 describe el método de estudio utilizando videos obtenidos del estudio PAMELA-UANDES. Los resultados del conteo de personas se presentan en el Capítulo 4, además de su implicancia en la estimación de demoras del transporte público. Finalmente, el Capítulo 5 se destina a las conclusiones del trabajo.

2. REVISIÓN BIBLIOGRÁFICA

2.1. Detección de Personas

Los primeros trabajos sobre la detección de peatones en lugares públicos utilizando cámaras estáticas abordaron el problema de los posibles cambios en la iluminación a través de un proceso de estimación y eliminación del fondo por sustracción entre el fondo y la imagen. El enfoque de modelación del fondo por una mezcla de gaussianas, fue propuesto por Stauffer y Grimson (1999); luego fue mejorado por Zivkovic (2004), y ha sido ampliamente utilizado por muchos autores, por ejemplo, Hu et al. (2013). En su forma original, el método tiene problemas cuando los objetos son estáticos o semi-estáticos (ya que son "absorbidos" en el modelo de fondo y cuando se mueven, dejan atrás un "fantasma"). Para abordar este problema, Li et al. (2008) proponen modelar pequeños movimientos como personas que giran o mueven la cabeza. La probabilidad de ocurrencia de movimiento se predice a partir de los cambios de color entre dos fotogramas consecutivos, utilizando funciones MID (diferencia de imagen de mosaico).

Sin embargo, el uso de un modelo de fondo tiene problemas significativos bajo hacinamiento o desorden, ya que los modelos mixtos dependen de la hipótesis que los píxeles de fondo se observarán la mayor parte del tiempo; es decir, son aplicables solo en condiciones de poco tráfico. Un enfoque alternativo es inferir la presencia de personas a través del análisis explícito de formas. Los primeros trabajos sobre esto son los de Gavrila y Giebel (2001) y Mundel et al. (2008), pero el enfoque es difícil de aplicar en escenas desordenadas. La amplia variabilidad de las formas

humanas dificulta el modelado explícito; por esto, las investigaciones se han centrado en los métodos de aprendizaje automático o *machine learning* (ML). Tradicionalmente, esto ha requerido: (a) un conjunto de datos sobre el cual entrenar y probar un método ML dado; (b) una definición de una o más características extraídas de las imágenes; y (c) un clasificador de características para separar diferentes clases de objetos (por ejemplo, personas y no personas). Un trabajo seminal que aborda estos tres aspectos es el de Dalal et al. (2005), que creó el conjunto de datos de personas de INRIA con imágenes tomadas principalmente en entornos urbanos (614 personas para entrenamiento y 288 para pruebas). Este trabajo también propuso una característica llamada *Histogram of Oriented Gradients* (HOG), que extrae bordes y características de textura de las regiones de la imagen y demostró ser popular para detectar y reconocer objetos (Zhu et al., 2006; Déniz et al., 2011; Chen et al., 2016). Finalmente, las características obtenidas por el HOG se ingresan en un clasificador de Máquina Vectorial o *Support Vector Machine* (SVM), para separar a las personas de los objetos que no son personas. Para más detalles sobre este enfoque (extracción de características, entrenamiento, clasificación), el lector se puede remitir a la exhaustiva revisión de Benenson et al. (2014) sobre detección tradicional de peatones.

Los avances recientes en aprendizaje profundo o *deep learning* logran la detección/clasificación de objetos en una imagen a través de dos métodos principales, como lo discutieron Soviany e Ionescu (2018). En el primero, existe una etapa inicial con *Region Proposal Network* (RPN) para generar regiones de interés. En una segunda etapa, esas regiones se utilizan para la *bounding box regression* y clasificación de objetos, pero estos detectores aunque precisos son lentos. Una familia alternativa de detectores, los llamados *Single Shot Detectors* (SSD), abordan la detección de objetos como un problema de regresión, analizando la imagen de entrada para aprender las probabilidades de clase y las coordenadas del rectángulo delimitador *(bounding box)*. Estos modelos son mucho más rápidos, incluso en tiempo real, pero pueden tener problemas para detectar objetos pequeños o cuando hay objetos que aparecen demasiado cerca en la imagen.

Zhang et al. (2020) reportan un método basado en YOLO para detectar personas subiendo/bajando de un bus, en que se entrena un modelo con las partes superiores del cuerpo de los pasajeros. Al igual que en nuestro trabajo, debieron entrenar con datos específicos. Sin embargo, no brindan métricas de rendimiento y no está claro si su conjunto de datos es público, lo que impide su verificación; tampoco se intenta conteos de pasajeros. Guo et al. (2020) proponen un modelo llamado MetroNet para detectar personas dentro de un vagón de metro, centrándose en la velocidad de procesamiento con bajos recursos computacionales. Nuevamente, tuvieron que crear un conjunto de datos, llamado SY-Net, con 1.503 imágenes a fin de comparar MetroNet con Faster-RCNN (Ren et al., 2015), SSD (Lui et al., 2016) y YOLOv3 (Redmon y Farhadi, 2018). Para esto usan la tasa de fallas ($MR = 1 - R$) como principal métrica, reportando un mejor MR de alrededor del 46%. No está claro si su conjunto de datos es público. Hsu et al. (2020) informan sobre un sistema de conteo de flujo de pasajeros en buses utilizando un detector SSD y un rastreador de filtro de partículas para luego obtener conteos. Aunque los resultados son buenos (F1 de alrededor del 94% para detección de personas y casi 92% para el conteo), sus datos no están disponibles públicamente. Liu et al. (2019) reportan un método para medir flujos de pasajeros en el metro utilizando YOLOv3, obteniendo precisiones en el flujo del 95%. En su caso, la cámara se monta en el marco de la puerta mirando verticalmente hacia abajo, lo que facilita mucho la detección y el conteo, pero no hay indicios de que sus datos sean públicos.

Con el objeto de presentar una línea base para el nuevo conjunto de datos públicos presentado en nuestro trabajo, se han seleccionado tres detectores: Faster-RCNN, EspiNet y YOLO (*single shot*). Los detalles de cada uno se pueden encontrar en Velastin et al. (2020).

2.2. Seguimiento de Objetos

El seguimiento o rastreo de objetos en imágenes trata de estimar las trayectorias de los objetos en el plano de la imagen a medida que se mueven por una escena (ver, por ejemplo, Yilmaz et al., 2006; Ojha y Sakhare, 2015). Implica ubicar cada objetivo en imágenes de video sucesivas después de haber sido localizado por un detector. Este enfoque normalmente se denomina seguimiento por detección (*tracking by detection*). Puede implicar la predicción de la posición de cada objeto, emparejando objetos entre imágenes adyacentes, para obtener un historial o trayectoria para cada uno. Existen técnicas para extraer atributos dinámicos, incluidos los cambios de apariencia, la velocidad posicional, la dirección del movimiento, etc. La mayoría de los algoritmos de seguimiento siguen un principio simple: se asume que es probable que los objetos en dos imágenes adyacentes correspondan entre sí, si la medición de la distancia entre ellos es pequeña. Esa distancia puede ser la separación física, la apariencia, la velocidad, etc. La comparación entre los objetos rastreados y las nuevas detecciones, se conoce como asociación de datos; es decir, dado un conjunto conocido de objetos que se rastrean en el tiempo t y un nuevo conjunto de objetos detectado en el tiempo t + 1, cómo asociar los nuevos objetos a las trayectorias existentes y cómo identificar objetos nuevos y los que han abandonado la escena. Este proceso debe tener en cuenta tanto las detecciones faltantes como las múltiples para un objeto determinado.

En los rastreadores de puntos, se puede encontrar el rastreo basado en siluetas y el rastreo basado en contornos (Yilmaz et al., 2006). Mientras que los rastreadores de puntos exigen detección en cada imagen, el rastreo basado en contornos (o kernel) requiere que los objetos aparezcan por primera vez en la escena. Para el seguimiento de vehículos, las características se asocian combinando el seguimiento de puntos y la apariencia del objeto, como lo hicieron, por ejemplo, Hou et al. (2019) con D-SORT.

Si bien el seguimiento de un solo objeto ya es una tarea compleja, el seguimiento de múltiples objetos (*Multiple Object Tracking* - MOT) es aún más desafiante, ya que implica localizarlos y seguirlos durante la secuencia de video, teniendo en cuenta la oclusión, y la entrada y salida de objetos en la escena.

2.3. Desafíos para la Detección y Seguimiento de Objetos

El aprendizaje profundo ha demostrado éxito en la detección de objetos. Ha habido muchas propuestas y variantes de arquitectura diferentes en la literatura, como CenterNet (Duan et al., 2019), EfficientNet (Tan et al., 2019), RetinaNet (utilizado por Wang et al., 2019 para detectar barcos), Faster-RCNN, propuesto por Ren et al. (2015), SSD discutido por Soviany et al. (2018) en comparación con otros detectores, YOLO de Bochkovskiy et al. (2020), etc. Para detectar objetos, es popular probar y comparar tales arquitecturas utilizando el conjunto de datos COCO (*Common Objects in COntext*) presentado por Lin et al. (2014), que contiene 80 clases de objetos, incluidas personas. Por ejemplo, YOLOv4 entrenado previamente con el conjunto de datos COCO, obtuvo resultados en imágenes nunca vistas antes por el detector, como se muestra en la Figura 2.

Figura 2 Predicciones de YOLOv4 en imágenes de personas preentrenado con datos de COCO Las detecciones son mostradas en magenta (Fuente: http://pexels.com)

Muchos de los datos para la detección de personas, incluido COCO, contienen principalmente imágenes de personas capturadas en vistas semifrontales de pie o caminando, lo que limita su uso. Todavía se necesita investigación para resolver el problema general de adaptarse a diferentes puntos de vista en el mismo dominio y, más aun, en diferentes dominios. En términos más generales, estos métodos solo pueden capturar la apariencia de los objetos para una determinada vista de cámara, pero aún no pueden aprender las propiedades subyacentes de los objetos.

Sin embargo, al tratar de detectar personas con YOLOv4, preentrenado con los datos de COCO, y luego aplicándolo a los datos PAMELA-UANDES, se obtuvieron resultados deficientes, como se muestra en la Figura 3. En la fila superior izquierda, el modelo preentrenado con COCO no detecta a ninguna persona. En la fila inferior izquierda el modelo preentrenado con COCO genera incorrectamente dos cuadros etiquetados como "Oso de peluche". Las imágenes de la derecha se obtuvieron después de entrenar un modelo YOLOv3 con los datos PAMELA-UANDES. Aunque para el ojo humano las imágenes siguen siendo de personas, el cambio en el ángulo de visión tiene un efecto adverso, por lo que muchas personas no son detectadas. Esto ilustra que muchos métodos de aprendizaje profundo todavía dependen de la anotación manual de grandes cantidades de datos, incluso cuando detectan la misma clase de objetos, pero con vistas diferentes.

3. METODOLOGÍA

En el contexto del problema abordado, es razonable pensar que para medir el número de personas que suben y bajan de un metro y sus flujos, es necesario primero detectar a cada individuo con precisión en cada cuadro (o en intervalos de tiempo regulares) alrededor de las puertas y luego rastrearlos en el espacio-tiempo, de modo que cada individuo solo se cuente una vez. Esto tiene que hacerse incluso en condiciones de hacinamiento. Se puede suponer que el sensor de la cámara es estático, aunque el enfoque adoptado aquí también puede funcionar con cámaras en movimiento.

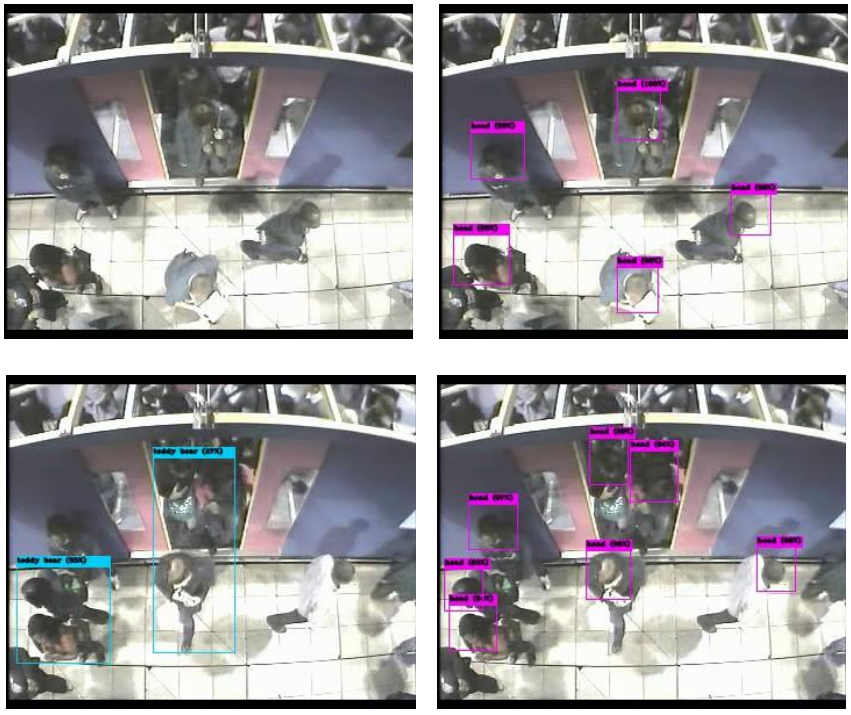

Figura 3 Predicciones de YOLO.
Arriba a la izquierda el ground truth. Abajo a la izquierda preentrenado con datos COCO.
A la derecha abajo y arriba, entrenado con datos PAMELA-UANDES

El conjunto de datos utilizado en este trabajo se obtuvo de un escenario simulado de un bus del sistema Transantiago, realizado en el laboratorio PAMELA-UCL (Fernández et al., 2010). Las imágenes se pueden encontrar en http://videodatasets.org/PAMELA-UANDES/ y son de uso libre para otros estudios. Los escenarios consideran dos anchos de puerta de los buses (800 y 1600 mm), tres alturas de escalón (0, 150 y 300 mm) y si al embarcar las personas utilizaron una tarjeta magnética sin contacto al subir al vehículo o con prepago antes de subir al vehículo, como las "zonas pagas" de Transantiago (paraderos cercados en los que se paga la tarifa al entrar al andén).

Dado el ángulo de la cámara que se tiene en la base PAMELA-UANDES, la hipótesis de trabajo es que, aún con una visión no cenital (Figura 3), la cabeza es suficiente para identificar a una persona y, una vez que esta se localiza, se le puede rastrear cuadro a cuadro para realizar conteos y flujos de pasajeros.

Parte del conjunto de imágenes de PAMELA-UANDES ha sido anotado manualmente, utilizando la herramienta ViPERtool (Doermann y Mihalcik, 2000), con el fin de probar los algoritmos de visión artificial. La anotación manual es una tarea importante, ya que se necesitaron alrededor de 2 meses-persona completos para anotar un subconjunto relativamente pequeño que constaba de 15 secuencias de video: 8 de personas que bajaban (denominadas videos "A") y 7 de personas que abordaban (denominadas videos "B"). Cada secuencia dura entre 1 y 2 min y tiene una resolución espacial de 352 × 288 a 25 imágenes por segundo.

A partir las imágenes PAMELA-UANDES, los videos denominados A_d800mm_R1.4.mpg y B_No_d800mm_R1.4.mpg se usan para entrenamiento, mientras que A_d800mm_R5.8.mpg y B_No_d800mm_R5.7.mpg se usan para pruebas. Los archivos *ground truth* están en formato ViPER y CSV, y contienen para cada pasajero un identificador único, el cuadro delimitador alrededor de sus cabezas y los números de cuadro en que aparece.

El conjunto de datos de PAMELA-UANDES contiene un total de 14.834 imágenes de entrenamiento y 13.237 imágenes de prueba. Para ambos, los casos de embarque y desembarque son más o menos similares en número. Como se ve a la izquierda de la Figura 4, la anotación original consta de las coordenadas de la imagen de una elipse definida por un rectángulo envolvente (tl_x, tl_y, w, h), donde tl_x, tl_y son las coordenadas x e y de la parte superior izquierda, y w, h son el ancho y la altura. Siguiendo la literatura, por ejemplo, Wolf et al. (2006) y Dalal et al. (2005), aquí también se plantea la hipótesis que expandir el cuadro delimitador para incluir parte del contexto de fondo puede ayudar a la detección. El nuevo cuadro delimitador se calcula como:

$$(c_x, c_y) = \left(tl_x + \frac{w}{2}, tl_y + \frac{h}{2}\right)$$

$$(w_e, h_e) = \big(w(1+f), h(1+f)\big), \tag{6}$$

$$gt_e = \left(tl_x - \frac{c_x}{2}\right), tl_v - \frac{c_y}{2}\ , w_e, h_e$$

donde (c_x, c_y) es el centroide del objeto (sin cambios), f es el factor de expansión (en el rango 0, 1), (w_e, h_e) el nuevo ancho y alto (expandidos), y gt_e el límite del objeto expandido. El efecto se ilustra en la derecha de la Figura 4.

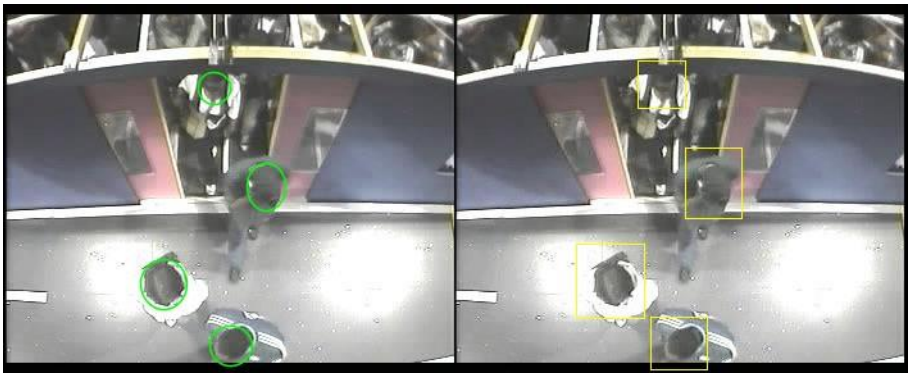

Figura 4 Modificación del *ground truth*.
Anotación original a la izquierda. Cuadros ampliados a la derecha.

Los modelos se entrenaron desde cero, sin utilizar ningún modelo previamente entrenado, para una comparación más objetiva. Por la misma razón, los anclajes de YOLO no se optimizaron ni se utilizó el aumento de datos. EspiNetV2 tardó 12 h en entrenarse (Windows), Faster R-CNN 18 h (Windows) y YOLOv3 14 h (Ubuntu). Se utiliza una GPU Nvidia Titan XP para entrenamiento y pruebas. La evaluación del rendimiento de detección se lleva a cabo utilizando las métricas definidas por las ecuaciones 1 a 5 mostradas en el Capítulo 2.

Las imágenes de PAMELA-UANDES utilizadas en este trabajo consistieron en secuencias de videos de subidas y bajadas independientes en las que se midió sólo el tiempo de servicio de pasajeros (*TSP*); es decir, el tiempo en que un vehículo de transporte público permanece completamente detenido para transferir sus pasajeros. El TSP se calculó como si las subidas y bajadas fuesen secuenciales; esto es el caso del metro, paraderos donde se paga al entrar al andén (zonas paga) y estaciones de BRT:

$$TSP = \beta_0 + \bar{\beta}_s P_s + \bar{\beta}_b P_b \tag{7}$$

donde *TSP* está en [s], β_0 es un tiempo muerto asociado a la apertura y cierre de puertas del vehículo [s]; $\bar{\beta}_s$ y $\bar{\beta}_b$ son los tiempos promedio que le toman a cada pasajero subir y bajar, respectivamente [s/pax]; P_s y P_b son, respectivamente, el número de pasajeros que sube y baja detectados por el método de visión artificial. Si a *TSP* se agrega el tiempo de frenado y aceleración en la estación, se obtendrá la demora de cada vehículo, es decir:

$$d = t_{af} + TSP \tag{8}$$

donde *d* es la demora que sufre un vehículo por detención en una estación [s] y t_{af} el tiempo de frenado y aceleración en la estación [s].

Para el cálculo del tiempo de aceleración y frenado se utilizó la siguiente ecuación:

$$t_{af} = \frac{V_r}{2}\left(\frac{1}{a} + \frac{1}{f}\right) \tag{9}$$

donde V_r es la velocidad de recorrido del vehículo al entrar y salir de la estación [m/s], y *a* y *f* son las tasas de aceleración y frenado, respectivamente [m/s^2]. Tanto *a* como *f* suelen asumirse iguales a 1,0 m/s^2, para comodidad de los pasajeros (TRB, 2003).

4. RESULTADOS

4.1. Resultados de Detección y Seguimiento de Pasajeros

Los resultados de detección y seguimiento de personas se pueden consultar en Velastin et al. (2020). No se entrará en mayores detalles, ya que el objetivo de este artículo es mostrar la eficiencia y resultados del conteo de pasajeros. A modo de ejemplo, la Figuras 5 y 6 muestran casos de detección y seguimiento. En la Figura 6 los números y colores corresponden a un identificador único de cada persona y las estelas muestran sus trayectorias.

4.2. Resultados del Conteo de Pasajeros

Como se indicó anteriormente, uno de los principales datos para estimar las demoras en estaciones es contar la cantidad de personas que suben y bajan del vehículo de transporte público. Una vez que las personas son detectadas y luego rastreadas, el conteo es relativamente más simple.

Figura 5 Ejemplo de detecciones
Caso simple a la izquierda. Caso complejo a la derecha.

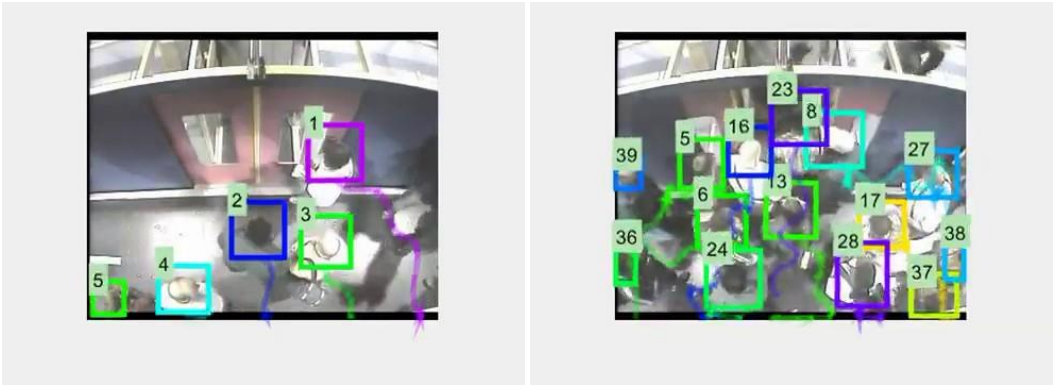

Figura 6 Ejemplo de seguimiento mostrando las trayectorias de las personas
Caso simple a la izquierda. Caso complejo a la derecha.

En este caso, se consideró una línea imaginaria x-y como se muestra en la Figura 7. Para cada persona se calcula su posición promedio $(\bar{x}_t\ ,\ \bar{y}_t)$ para todas sus posiciones hasta el instante *t*, comenzando desde su primera aparición. Por ejemplo, para una persona que baja, este promedio va a tener valores decrecientes de $\bar{y}_t$ a medida que *t* aumenta. Cuando ese promedio pasa de un lado al otro de la línea, la persona activa un conteo: hacia abajo (bajada) si se originó por encima de la línea y hacia arriba (subida) en caso contrario.

Figura 7 Línea imaginaria para contar pasajeros.
(hacia arriba: subida; hacia abajo: bajada)

La evaluación de conteos se realizó con 145 secuencias de bajada (la mayoría con 50 personas) y 177 secuencias de subida (la mayoría con 27-29 personas), en diferentes condiciones de aglomeración y factores físicos, como el ancho de puerta. Los resultados se resumen en la Tabla 2.

Tabla 2 Detección promedio de las 322 secuencias de video y su desviación estándar (%)

Proceso	***Precision***	***Recall***	***F1***	***Accuracy***
Bajada	95,49 ± 0,13	95,04 ± 0,12	95,10 ± 0,12	92,01 ± 0,12
Subida	93,59 ± 0,05	98,82 ± 0,02	96,01 ± 0,03	92,47 ± 0,05

En la Tabla 2 *Precision* indica cuán cerca están dos mediciones independientes entre sí y *Accuracy* es la cercanía del valor medido con la "realidad observada" o *ground truth*. Para evaluar el método se usó esta última métrica.

4.3. Efecto en la Estimación de la Demora en Estaciones

Con lo establecido en la sección anterior, se puede hacer una simulación de cuánto implicaría la exactitud en la predicción de las demoras en estaciones, utilizando valores estándar de los parámetros de las ecuaciones (7) y (8). En este ejercicio se usaron los parámetros de la Tabla 3 de TRB (2003) y Tirachini (2013).

Tabla 3 Parámetros usados para la predicción de la demora

Parámetro	**Valor**
Tiempo muerto (β_0)	3,0 s
Tiempo promedio de subida (β_s)	3,5 s/pax
Tiempo promedio de bajada (β_b)	2,1 s/pax
Tasas de aceleración (a) y frenado (f)	1,0 m/s^2
Velocidad de recorrido (V_r)	10,0 m/s

Se efectuaron 100 simulaciones en una planilla de cálculo para combinaciones aleatorias entre 1 y 20 pasajeros subiendo/bajando, así como valores aleatorios en el rango de la exactitud de la visión computacional; por ejemplo, entre 91,89% y 92,13% para las bajadas. La Tabla 4 muestra los resultados de las 10 primeras simulaciones. Para el resto, los resultados son prácticamente idénticos. Se observa un error promedio de 6% en la demora estimada usando valores recabados por la visión computacional con respecto a la realidad y, en el peor de los casos, el error no llegó al 7%. Estos valores se consideran apropiados para todo efecto práctico. Sin embargo, como la diferencia es sistemática, se podrían corregir en este mismo porcentaje para obtener la demora real.

5. CONCLUSIONES

Desde el punto de vista de la detección, seguimiento y conteo de objetos, se pone a disposición pública un conjunto de datos de video con 348 secuencias capturadas por una cámara CCTV estándar que muestra a personas subiendo y bajando de un modelo de tamaño real de un vehículo de transporte público (http://videodatasets.org/PAMELA-UANDES/). Un subconjunto de estos videos se procesó visualmente para ubicar y rastrear la cabeza de cada persona y permitir el entrenamiento y la prueba de algoritmos de detección. Con esto se espera que otras investigaciones puedan replicar y mejorar estos resultados.

Tabla 4 Comparación entre el *TSP* y demora (*d*) para valores aleatorios de pasajeros

Realidad observada				**Estimación con VC**				**Diferencia**
Suben	**Bajan**	***TSP***	***d***	**Suben**	**Bajan**	***TSP'***	***d'***	**c/r**
[pax]	**[pax]**	**[s]**	**[s]**	**[pax]**	**[pax]**	**[s]**	**[s]**	**realidad**
16	20	75	85	14,8	18,4	69,4	79,4	**6,62%**
11	19	63	73	10,2	17,5	58,4	68,4	6,36%
5	13	39	49	4,6	11,9	36,1	46,1	5,83%
7	7	31	41	6,5	6,4	28,8	38,8	5,26%
9	11	43	53	8,3	10,1	39,9	49,9	5,83%
16	15	65	75	14,8	13,8	60,2	70,2	6,38%
18	12	63	73	16,6	11,0	58,3	68,3	6,38%
10	11	45	55	9,3	10,1	41,8	51,8	5,87%
20	2	47	57	18,5	1,8	43,7	53,7	5,86%
18	3	45	55	16,6	2,8	41,8	51,8	5,80%
							Promedio	**6,02%**

Del análisis de conteos de personas para 322 secuencias de video, se obtuvo una exactitud superior al 92%. Este resultado no había sido encontrado en la literatura y fue lo que motivó este trabajo.

Además, mediante una simulación en una planilla de cálculo con parámetros estándar de operación del transporte público, se determinó que esta exactitud llevaba a una subestimación sistemática menor al 7% en la demora de un vehículo en una estación. Con este resultado, ya se puede utilizar la visión artificial en la operación del transporte público, sabiendo la magnitud del error que se comete. Hasta donde sabemos, este hallazgo tampoco había sido reportado en la literatura.

Los sistemas de transporte público cerrados, como metro, tren y BRT, poseen cámaras de vigilancia en sus estaciones. Hasta ahora, estos sistemas de CCTV se han utilizado para observar situaciones relacionadas con la seguridad, como un objeto en el andén, una persona sospechosa o un potencial suicidio (Velastin et al., 2006). Sin embargo, como se muestra en este artículo, las mismas cámaras se podrían utilizar para realizar mediciones que ayuden a mejorar la operación en las estaciones. Esta es la contribución que pretende este trabajo.

REFERENCIAS

Benenson, R., Omran, M., Hosang, J. y Schiele, B. (2014) Ten years of pedestrian detection, what have we learned? *Proceedings of the European Conference on Computer Vision*. 6–12 septiembre 2014. Zurich, Suiza.

Bernardin, K. y Stiefelhagen, R. (2008) Evaluating multiple object tracking performance: the CLEAR MOT metrics. *EURASIP Journal on Image and Video Processing* **2008**, 1-10.

Bochkovskiy, A., Wang, C.Y. y Liao, H.Y.M. (2004) YOLOv4: optimal speed and accuracy of object detection. a*rXiv:2004.10934*.

Chen, Z., Ellis, T. y Velastin, S.A. (2016) Vision-based traffic surveys in urban environments. *Journal of Electronic Imaging* **25**, 051206.

Dalal, N. y Triggs, B. (2005) Histograms of oriented gradients for human detection. *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'05). 20–25 junio 2005. San Diego, CA., EE.UU.

Déniz, O., Bueno, G., Salido, J. y De la Torre, F. (2011) Face recognition using histograms of oriented gradients. *Pattern Recognition Letters* **32**, 1598-1603.

Doermann, D. y Mihalik, D. (2000) Tools and techniques for video performance evaluation. *Proceedings 15th International Conference on Pattern Recognition*, ICPR-2000. 3–7 septiembre, 2000. Barcelona, España

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q. y Tian, Q. (2019) Centernet: keypoint triplets for object detection. *Proceedings of the IEEE International Conference on Computer Vision*. 27–28 octubre 2019. Seúl, Corea del Sur.

Everingham, M., Van Gool, L., Williams, C.K., Winn, J. y Zisserman, A. (2010) The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision* **88**, 303-338.

Fernández, R., Zegers, P., Weber, G. y Tyler, N. (2010) Influence of platform height, door width, and fare collection on bus dwell time: laboratory evidence for Santiago de Chile. *Transportation Research Record* **2143**, 59–66.

Gavrila, D. y Giebel, J. (2001) Virtual sample generation for template-based shape matching. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR 2001. 18–14 diciembre 2001. Kauai, EE.UU.

Guo, Q., Liu, Q., Wang, W., Zhang, Y. y Kang, Q. (2020) A fast occluded passenger detector based on MetroNet and Tiny MetroNet. *Information Sciences* **534**, 16–26.

Hadavi, S., Rai, H.B., Verlinde, S., Huang, H., Macharis, C. y Guns, T. (2020) Analysing passenger and freight vehicle movements from automatic-number plate recognition camera data. *European Transport Research Review* **12**, 1–17.

Hou, X., Wang, Y. y Chau, L.P. (2019) Vehicle tracking using deep sort with low confidence track filtering. *Proceedings of the 16th IEEE International Conference on Advanced Video and Signal Based Surveillance* (AVSS). 18–21 septiembre 2019. Taipéi, Taiwán.

Hsu, Y.W., Wang, T.Y. y Perng, J.W. (2020) Passenger flow counting in buses based on deep learning using surveillance video. *Optik* **202**, 163675.

Hu, X., Zheng, H., Wang, W. y Li, X. (2015) A novel approach for crowd video monitoring of subway platforms. *Optik* **124**, 5301–5306.

Lealtaixé, L., Milan, A., Reid, I., Roth, S. y Schindler, K. (2015) MOT challenge 2015: towards a benchmark for multi-target tracking. *arXiv:1504.01942.*

Li, M., Zhang, Z., Huang, K. y Tan, T. (2008) Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. *Proceedings of the 19th International Conference on Pattern Recognition*. 8–11 diciembre 2008. Florida, EE.UU.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. y Zitnick, C.L. (2014) Microsoft COCO: common objects in context. *Proceedings of the European Conference on Computer Vision*. 6–12 septiembre 2014. Zurich, Suiza.

Liu, W., Du, X., Geng, Q., Li, J., Li, H. y Liu, L. (2019) Metro passenger flow statistics based on YOLOv3. *Proceedings of the IOP Conference Series: Materials Science and Engineering*. Vol. 688. IOP Publishing, Bristol, UK.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C. (2016). SSD: Single Shot MultiBox Detector. En *Lecture Notes in Computer Science*, pp. 21–37. arXiv:1512.02325

Milan, A. y Ristani, E. (2020) MOT Challenge Development Kit. https://motchallenge.net/devkit.

Munder, S., Schnorr, C. y Gavrila, D.M. (2008) Pedestrian detection and tracking using a mixture of view-based shape–texture models. *IEEE Transactions on Intelligent Transportation Systems* **9**, 333-343.

Ojha, S. y Sakhare, S. (2015) Image processing techniques for object tracking in video surveillance – a survey. *Proceedings of the 2015 International Conference on Pervasive Computing* (ICPC). 8–10 enero 2015, Pune, India.

Redmon, J. and Farhadi, A. (2018) Yolov3: An incremental improvement. *arXiv:1804.02767v1*.

Ren, S., He, K., Girshick, R. y Sun, J. (2015) Faster R-CNN: towards real-time object detection with region proposal networks. *Proceedings of the Advances in Neural Information Processing Systems 28* (NIPS 2015). 7–12 diciembre 2015. Montreal, Canadá.

Soviany, P. y Ionescu, R.T. (2018) Optimizing the trade-off between single-stage and two-stage object detectors using image difficulty prediction. *arXiv:1803.08707.*

Stauffer, C. y Grimson, W.E.L. (1999) Adaptive background mixture models for real-time tracking. *Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 23–25 junio 1999. Colorado, EE.UU.

Tan, M. y Le, Q.V. (2019) EfficientNet: rethinking model scaling for convolutional neural networks. *arXiv 2019.*

Tirachini, A. (2013) Bus dwell time: the effect of different fare collection systems, bus floor level and age of passengers. *Transportmetrica A: Transport Science* **9**, 28-49.

TRB (2003) *Transit Capacity and Quality of Service Manual*, Transit Cooperative Research Program (TCRP) Report 100, 2nd Edition. Transportation Research Board, Washington, DC.

Velastin, S.A., Fernández, R., Espinosa, J.E., y Bay, A. (2020) Detecting, tracking and counting people getting on/off a metropolitan train using a standard video camera. *Sensors* **20**, 6251.

Velastin, S.A., Boghossian, B.A. y Vicencio-Silva, M.A. (2006) A motion-based image processing system for detecting potentially dangerous situations in underground railway stations. *Transportation Research Part C: Emerging Technologies* **14**, 96-113.

Wang, Y., Wang, C., Zhang, H., Dong, Y. y Wei, S. (2019) Automatic ship detection based on RetinaNet using multi-resolution Gaofen-3 imagery. *Remote Sensing* **11**, 531.

Wolf, L. y Bileschi, S. (2006) A critical view of context. *International Journal of Computer Vision* **69**, 251–261.

Yilmaz, A., Javed, O. y Shah, M. (2006) Object tracking: a survey. *ACM Computing Surveys* **38**, 13-es.

Yin, F., Makris, D., Velastin, S.A. y Orwell, J. (2010) Quantitative evaluation of different aspects of motion trackers under various challenges. *British Machine Vision Association* **5**, 1–11.

Zhang, S., Wu, Y., Men, C. y Li, X. (2020) Tiny YOLO optimization-oriented bus passenger object detection. *Chinese Journal of Electronics* **29**, 132–138.

Zhu, Q., Yeh, M.C., Cheng, K.T. y Avidan, S. (2006) Fast human detection using a cascade of histograms of oriented gradients. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'06). 17–22 junio 2006. Nueva York, EE.UU.

Ziółkowski, R. (2018) Speed management efficacy on national roads–early experiences of sectional speed system functioning in Podlaskie voivodship. *Transport Problems* **13**, 5–12.

Zivkovic, Z. (2004) Improved adaptive Gaussian mixture model for background subtraction. *Proceedings of the 17th International Conference on Pattern Recognition (*ICPR). 26–26 agosto 2004. Cambridge, Reino Unido.