Propuesta de Herramienta de Reconocimiento de Personas Mediante Técnicas de Inteligencia Artificial para Conteo de Abordaje al Transporte Público

Ing. Emmanuel de Jesús Santiago Saavedra[1] y Mtro. Fernando González Díaz[2]

***Resumen*—**Se propone el desarrollo e integración de un sistema de visión artificial, el cual incorpore un algoritmo de reconocimiento de personas capaz de detectar el número de pasajeros que abordan a los camiones urbanos del transporte público, es decir, que sea capaz de determinar el número de viajes efectivos, con lo anterior se pretende mejorar la precisión de los conteos de pasajeros que actualmente se contabilizan mediante barras con sensores infrarrojos. Se implementará un detector You Only Look Once (por sus siglas en inglés YOLO) + rastreador Deep Simple Online Real Time Tracker (por sus siglas en inglés DeepSORT).

***Palabras clave*—**Conteo de pasajeros, OpenCV, YOLO, DeepSORT.

Conteo de pasajeros

Existen diversas tecnologías exploradas para el conteo de pasajeros, entre ellas, se encuentran: barreras de detección de cruce mediante sensores de luz Infrarroja, torniquetes, cámaras de visión computacional. Actualmente en las unidades del transporte público del estado de Jalisco se opera el conteo de abordajes/descensos a las unidades del transporte público mediante la implementación de una tecnología que incorpora barreras de luz infrarroja a bordo de las unidades; los torniquetes por el espacio físico tan amplio que requieren, son una opción menos viable para instalar en autobuses; por último, las cámaras de visión computacional son, la herramienta objeto del presente trabajo de estudio y que se pretende explorar como solución para el conteo de pasajeros.

El estado del arte en visión artificial

Referente al tema de desarrollo de sistemas de visión artificial encontramos, hoy en día, diferentes librerías y toolkits, como Open CV, TensorFlow, MATLAB, Keras, CAFFE:

Open CV

Es una librería de visión computacional. Provee acceso a más de 2500 algoritmos. Se puede programar en C++, Python, Java, MATLAB, soporta los sistemas operativos: Windows, Android, Linux, Mac.

¿Cuál herramienta utilizar?

Referente al tema de detección de personas (peatones) encontramos: “Desde hace muchos años, la detección de peatones se resuelve casi exclusivamente mediante algoritmos de deep learning.” [3]. De la comparativa analizada por varios autores tenemos que mencionan una tasa de éxito mayor para CNN (Convolutional Neural Networks) en aplicaciones de detección de personas. Por esta razón usaremos un modelo basado en redes neuronales para la etapa de detección. En cuanto a algunos de los métodos de detección de objetos, basados en Deep learning, tenemos: Faster R-CNN (Region Based Convolutional Neural Network), YOLO (You Only Look Once) y SSD (Single Shot Detector).

El detector YOLO

YOLO (You Only Look Once) tiene varias ventajas sobre los sistemas basados en clasificadores, esto, debido a que YOLO observa la imagen completa en el momento del muestreo, por lo que sus predicciones se basan en el contexto global de la imagen. También hace predicciones con una sola evaluación a diferencia de sistemas como R-CNN (Region Based Convolutional Neural Network) que requieren miles de evaluaciones para una sola imagen. Lo anterior, hace que YOLO sea un detector más de 1000 veces más rápido que R-CNN y 100 veces más rápido que Fast R-CNN.

[1] El Ing. Emmanuel de Jesús Santiago Saavedra es egresado de la carrera de Ing. en electrónica por parte del Centro de Enseñanza Técnica Industrial (CETI), Guadalajara Jalisco, así como egresado de la Especialidad en sistemas embebidos por parte del Instituto Tecnológico y de Estudios Superiores de Occidente (ITESO), en Guadalajara, Jalisco, y estudiante de la Maestría en Sistemas Inteligentes Multimedia en el Centro de Tecnología Avanzada (Posgrado CIATEQ), sede Zapopan, Jalisco esantiagosaavedra@gmail.com

[2] El Mtro. Fernando González Díaz es líder de la especialidad de Telecom en el Centro de Tecnología Avanzada (CIATEQ), El Marqués, Qro. fernando.gonzalez@ciateq.mx

Detectores de una y de dos etapas

En el detector de una sola etapa, la detección sea aplica directamente al mapa de características (feature map), mientras que, en el de dos etapas, primero se aplica una red de propuesta de región (region proposal network) en los mapas de características. Luego, estas regiones pasan a la segunda etapa, que hace predicciones para cada región. Faster-RCNN y Mask-RCNN son algunos de los detectores de objetos de dos etapas más populares. Si bien los detectores de dos etapas se consideran más precisos que los detectores de objetos de una sola etapa, tienen una velocidad de inferencia más lenta, ya que involucra múltiples etapas. YOLO es un detector de una etapa.

En cuanto a la relación velocidad-precisión de detección podemos ver la Tabla 1, en donde se hace una comparativa de diferentes detectores de objetos.

Method	Backbone	Size	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv4: Optimal Speed and Accuracy of Object Detection									
YOLOv4	CSPDarknet-53	416	38 (M)	41.2%	62.8%	44.3%	20.4%	44.4%	56.0%
YOLOv4	CSPDarknet-53	512	31 (M)	**43.0%**	**64.9%**	**46.5%**	**24.3%**	**46.1%**	**55.2%**
YOLOv4	CSPDarknet-53	608	23 (M)	43.5%	65.7%	47.3%	26.7%	46.7%	53.3%
Learning Rich Features at High-Speed for Single-Shot Object Detection [84]									
LRF	VGG-16	300	76.9 (M)	32.0%	51.5%	33.8%	12.6%	34.9%	47.0%
LRF	ResNet-101	300	52.6 (M)	34.3%	54.1%	36.6%	13.2%	38.2%	50.7%
LRF	VGG-16	512	38.5 (M)	36.2%	56.6%	38.7%	19.0%	39.9%	48.8%
LRF	ResNet-101	512	31.3 (M)	37.3%	58.5%	39.7%	19.7%	42.8%	50.1%
Receptive Field Block Net for Accurate and Fast Object Detection [47]									
RFBNet	VGG-16	300	66.7 (M)	30.3%	49.3%	31.8%	11.8%	31.9%	45.9%
RFBNet	VGG-16	512	33.3 (M)	33.8%	54.2%	35.9%	16.2%	37.1%	47.4%
RFBNet-E	VGG-16	512	30.3 (M)	34.4%	55.7%	36.4%	17.6%	37.0%	47.6%
YOLOv3: An incremental improvement [63]									
YOLOv3	Darknet-53	320	45 (M)	28.2%	51.5%	29.7%	11.9%	30.6%	43.4%
YOLOv3	Darknet-53	416	35 (M)	31.0%	55.3%	32.3%	15.2%	33.2%	42.8%
YOLOv3	Darknet-53	608	20 (M)	33.0%	57.9%	34.4%	18.3%	35.4%	41.9%
YOLOv3-SPP	Darknet-53	608	20 (M)	36.2%	60.6%	38.2%	20.6%	37.4%	46.1%
SSD: Single shot multibox detector [50]									
SSD	VGG-16	300	43 (M)	25.1%	43.1%	25.8%	6.6%	25.9%	41.4%
SSD	VGG-16	512	22 (M)	28.8%	48.5%	30.3%	10.9%	31.8%	43.5%

Tabla 1. Comparación de la velocidad y precisión de diferentes detectores de objetos en el conjunto de datos MS COCO (MS COCO Dataset) (test-dev 2017). Los detectores en tiempo real con FPS (frames por segundo) 30 o superior se destacan aquí. YOLO alcanza un mAP (mean Average Precision) de entre 41.2% a 43% [4].

El rastreador DeepSORT (Deep Simple Online RealTime Tracker)

DeepSORT es un algoritmo de rastreo de visión computacional, para rastrear objetos, mientras asigna un identificador a cada objeto. DeepSORT es una extensión del algoritmo SORT (Simple Online Realtime Tracking). DeepSORT introduce el aprendizaje profundo (Deep Learning) en el algoritmo SORT al agregar un descriptor de apariencia para reducir las incidencias en los intercambios no deseados de ID's, lo que hace que el seguimiento sea más eficiente.. SORT se compone de 4 etapas fundamentales, que son:

Detección: En este paso, un detector de objetos detecta los objetos en la escena que se van a rastrear. Estas detecciones luego se pasan al siguiente paso. Detectores como F-RCNN y YOLO son los más utilizados.

Estimación: en este paso, se propagan las detecciones del cuadro actual al siguiente, el cual estima la posición del objetivo en el cuadro siguiente utilizando un modelo de velocidad constante. Cuando una detección está asociada con un objetivo (target), el cuadro delimitador detectado (bounding box) se usa para actualizar el estado del objetivo.

Asociación de datos: En este punto, se tiene el cuadro delimitador objetivo (target bounding box) y el cuadro delimitador detectado. Por lo tanto, se calcula una matriz de costos como distancia de intersección sobre unión (IOU), entre cada detección y todos los cuadros delimitadores (bounding boxes) previstos de los objetivos (targets) existentes. Esto se resuelve de manera óptima utilizando el algoritmo húngaro. Si el IOU de detección y objetivo es inferior a un cierto valor de umbral llamado IOUmin, entonces se rechaza esa asignación. Esta técnica soluciona el problema de oclusión y ayuda a mantener los ID's detectados.

Creación y eliminación de identidades rastreadas: este módulo es responsable de la creación y eliminación de ID's. Las identidades únicas se crean y destruyen de acuerdo con el IOUmin. Si la superposición de la detección y el objetivo (target) es inferior a IOUmin, significa que el objeto no se ha rastreado. El rastreo se termina si no se detecta durante cierto número de cuadros. Si un objeto reaparece, el seguimiento se reanudará implícitamente con una nueva identidad (ID). Los objetos se pueden rastrear con éxito utilizando algoritmos SORT que superan a muchos algoritmos de última generación. El detector nos da detecciones, los filtros de Kalman nos dan pistas y el algoritmo húngaro realiza la asociación de datos.

Efectividad de DeepSORT

Es posible notar una reducción en el número de cambios de ID. En comparación con SORT, los intercambios de ID se reducen de 1423 a 781. Esta es una disminución de aproximadamente 45%. También vemos un aumento significativo en el rastreo de objetos y una disminución en la pérdida del rastreo de estos. En general, debido a la integración de la información de apariencia, es posible mantener ID's durante oclusiones más largas [5].

Nuestra herramienta de conteo YOLO+DeepSORT

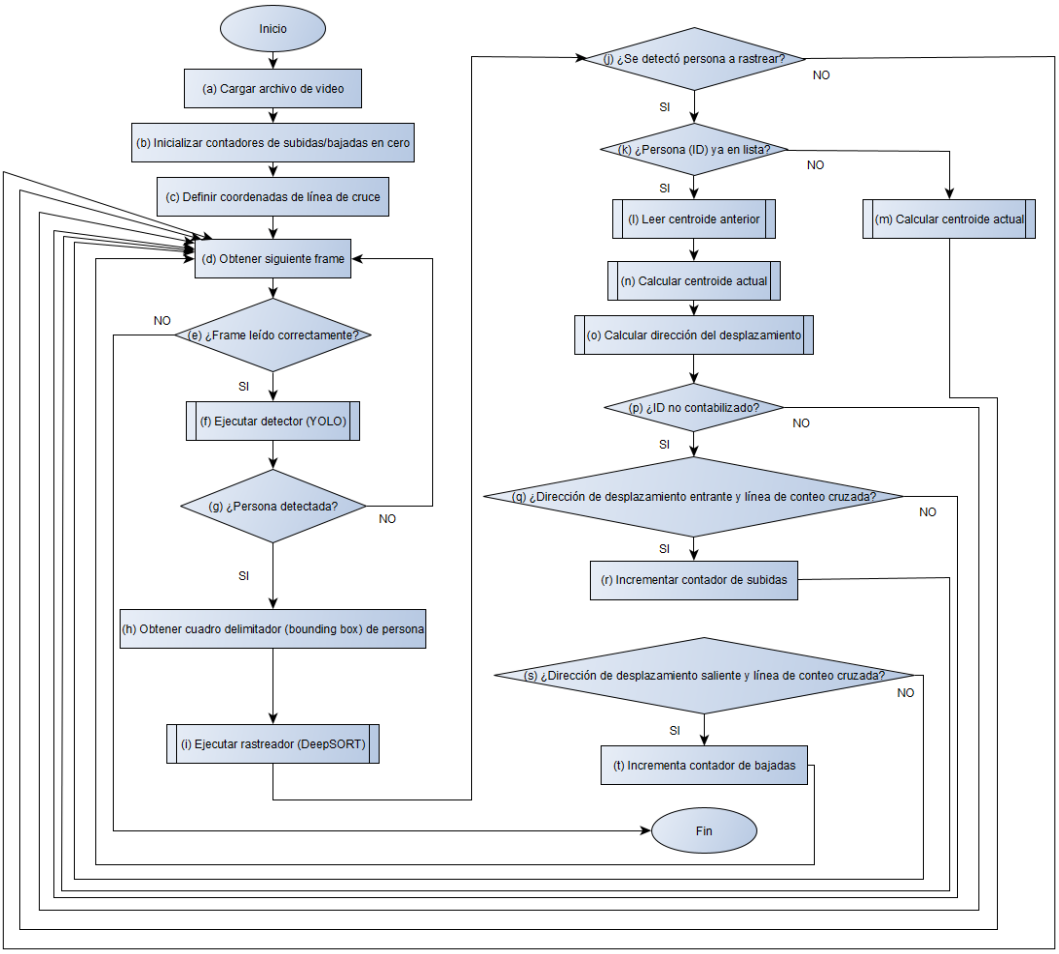

Figura 1. Algoritmo de conteo de abordajes/descensos de pasajeros, con nuestra herramienta de visión artificial.

De la Figura 1, vemos que en (f) se ejecuta el detector YOLO, al cual se le ha aplicado un filtro de clases para que únicamente detecte la categoría "persona". Cuando el detector encuentra una persona en escena, retorna información acerca de la detección, como las coordenadas del cuadro delimitador (bounding box) de la persona detectada. Si a su vez, el rastreador DeepSORT (el cual se ejecuta en (i)), identifica una persona a rastrear, entonces habremos de hacer el cálculo del centroide (conjunto de coordenadas x,y del punto central de la caja delimitadora identificada) de dicha detección, esto para determinar la dirección del movimiento de la persona detectada. El cálculo del centroide se calcula de la siguiente manera: Para la imagen que se ilustra en la Figura 2, tenemos la caja delimitadora (bounding box) que identifica a "person-8", esta tiene 4 vértices nombrados como v1,v2,v3,v4 (aparecen en color amarillo en la Figura 2). Para cada vertice vN tenemos un par de coordenadas xN,yN, si queremos encontrar las coordenadas xC,yC, del punto central C (aparece en color verde en la Figura 2), al que llamaremos centroide de la caja delimitadora, tenemos que

$$xC = (x1 + x2) / 2 = (x4 + x3) / 2$$
$$yC = (y1 + y2) / 2 = (y4 + y3) / 2$$

es decir, el centroide en "x" se calcula con la coordenada de inicio y de fin en "x", y el centroide en "y" se calcula con la coordenada de inicio y de fin en "y"

Cabe señalar que, dependiendo de la posición de la cámara, podemos ajustar este centroide para que en lugar de que sea calculado al centro de la caja delimitadora, se calcule al centro de una arista lateral de esta.

Para el cálculo de la dirección del desplazamiento de la persona, hacemos la resta de coordenadas del centroide de la detección anterior, menos el centroide de la detección actual,

$$\text{dirección} = xC \text{ anterior} - xC \text{ actual}$$

esto en el eje x, ya que nuestra línea de cruce (ver Figura 3a) será colocada de tal manera que las personas hagan el cruce paralelamente a esta dirección. Si el valor obtenido para la dirección es un valor positivo, entonces significa que la persona se desplaza en dirección de derecha a izquierda, es decir, de manera ascendente, por el contrario, si la dirección resulta en un valor negativo, la persona se desplaza en dirección de izquierda a derecha, es decir, de manera descendente.

Figura 2. Detección de persona con nuestra herramienta de visión artificial. En color amarillo se identifican los cuatro vértices de la caja delimitadora (v1,v2,v2,v4), en color verde se identifica el centroide (C), o centro de la caja delimitadora,

Resultados

Se somete a prueba nuestra herramienta de visión artificial (detector YOLO + rastreador DeepSORT), haciendo una comparativa de conteo de ascensos y descensos en escalera, comparando con la detección que arrojan las barras infrarrojas para el mismo evento. Las barras con sensores infrarrojos no son capaces de detectar conteos cuando un pasajero obstruye (permanece estático) alguna de las barras, mientras otro pasajero pasa junto a él. Ver Figura 3.

Secuencia de descenso (bajada)

Figura 3. Secuencia de descenso (bajada) de escaleras. (a) La línea de cruce (línea roja) se coloca a la par de las barras con sensores infrarrojos. En la parte inferior izquierda de la pantalla se muestran los conteos de subidas

(ascensos, cuando se cruzan las barras desde la escalera hacia el pasillo) y bajadas (descensos, cuando se cruzan las barras desde el pasillo hacia la escalera). Inicialmente el conteo comienza en cero, tanto para subidas como para bajadas. (b) Dos personas aparecen en escena realizando un descenso. En este caso la persona marcada con el ID "person-4", realiza intencionalmente una obstrucción de una de las barras, permaneciendo estática en esta posición durante un momento, con el fin de evitar que la persona marcada con el ID "person-5" sea contabilizada al cruzar las barras; esto para simular un caso real de obstrucción de barras a bordo de un autobús del transporte público. (c) Las dos personas en la escena finalizan el descenso (el ID "person-5" sale de cuadro). Note como los contadores de la parte inferior izquierda de la pantalla, indican los dos descensos realizados por "person-4" y "person-5".

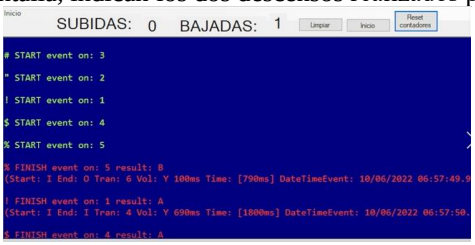

Figura 4. El software contabilizador de cruce de las barras infrarrojas detectó solamente una (siendo dos, el valor real) bajada (descenso) durante la prueba, esto debido a la obstrucción intencional de una de las barras.

Secuencia de ascenso (subida)

Figura 5. Secuencia de ascenso (subida) de escaleras. (a) La línea de cruce (línea roja) se coloca a la par de las barras con sensores infrarrojos. En la parte inferior izquierda de la pantalla se muestran los conteos de subidas (ascensos, cuando se cruzan las barras desde la escalera hacia el pasillo) y bajadas (descensos, cuando se cruzan las barras desde el pasillo hacia la escalera). Inicialmente el conteo comienza con dos bajadas debido a que esta prueba arranca al término de la anterior (prueba de descenso). Una persona aparece en escena ("person-6") en dirección ascendente; esta aún no es contabilizada en este punto, ya que no se ha realizado el cruce de la línea de conteo. (b) Una segunda persona ("person-8") aparece en escena. En este caso la persona marcada con el ID "person-6", realiza intencionalmente una obstrucción de una de las barras, permaneciendo estática en esta posición durante un momento, con el fin de evitar que la persona marcada con el ID "person-8" sea contabilizada al cruzar las barras; esto para simular un caso real de obstrucción de barras a bordo de un autobús del transporte público. (c) La persona "person-8" completa el ascenso. Note como el contador de subidas de la parte inferior izquierda, ha contabilizado la subida. La persona "person-6" continúa obstruyendo una de las barras sin haber cruzado la línea todavía. (d) La persona "person-6" cruza la línea de conteo y sale de escena. Note como los contadores de la parte inferior izquierda de la pantalla, indican los dos ascensos realizados por "person-6" y "person-8".

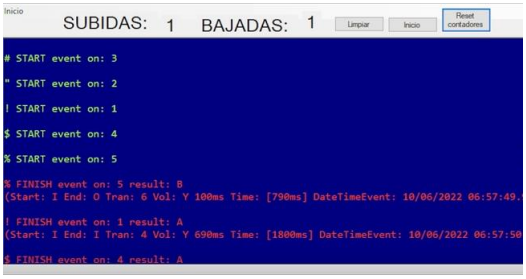

Figura 6. El software contabilizador de cruce de las barras infrarrojas detectó solamente una (siendo dos, el valor real) subida (ascenso) durante la prueba, esto debido a la obstrucción intencional de una de las barras.

Dirección	Total	Cámara	Barras
Subida	20	19	19
Bajada	20	19	19

Tabla 2. Conteo de subidas/bajadas sin obstrucción de ruta de abordaje. En condiciones ideales (sin obstrucción del camino para ascender/descender de la unidad del transporte), tanto nuestra herramienta de visión artificial, como las barras contadoras tuvieron la misma precisión en el conteo.

Dirección	Total	Cámara	Barras
Subida	20	19	17
Bajada	20	19	17

Tabla 3. Conteo de subidas/bajadas con obstrucción de ruta de abordaje. Mediante esta prueba se simula un escenario donde hay saturación de pasajeros en la unidad del transporte público. Al obstruir intencionalmente una de las barras contadoras, tal como en el ejercicio mostrado en la Figura 5b, el conteo con nuestra herramienta de visión artificial fue un 10% mejor que con el conteo mediante barras con sensores infrarrojos.

Comentarios finales

Aun cuando las barras con sensores infrarrojos presentan una confiabilidad del 96% en operación normal (sin obstrucciones debido a saturación de pasajeros), existen casos, que no son posibles de manejar, esto debido a que las barras no son capaces de identificar pasajeros, sino, únicamente, responder al estímulo que la presencia de estos (pasajeros) causa en los sensores infrarrojos colocados en ellas. Vemos que el conteo mediante cámara es especialmente útil en los casos de obstrucciones de barras, ya que, en horarios de gran afluencia de pasajeros en el transporte público, estos eventos suelen aumentar considerablemente, lo cual afecta la precisión de los conteos mediante barras con sensores. Las barras pueden actuar como un sistema auxiliar de conteo, pudiendo incluso (una vez completado con éxito un set de pruebas en unidades del transporte público), ser sustituidas por un sistema de conteo mediante técnicas de inteligencia (visión) artificial como el propuesto en este artículo.

Recomendaciones

En condiciones de poca iluminación el algoritmo presentó dificultades para la detección precisa de pasajeros. Al cambiar por una cámara con función de modo nocturno se obtiene mejor resultado. Para condiciones nocturnas habrá que considerar mejorar las condiciones de iluminación en los buses, o, utilizar una cámara con características adecuadas para los ambientes nocturnos.

Referencias

[1] Liu, X., Tu, P. H., Rittscher, J., Perera, A., & Krahnstoever, N. (2005). Detecting and counting people in surveillance applications. Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2005., 306–311. https://doi.org/10.1109/AVSS.2005.1577286

[2] Siebel, N. (2003). Design and Implementation of People Tracking Algorithms for Visual Surveillance Applications. http://www.cvg.reading.ac.uk/papers/advisor/Siebel-thesis-onesided.pdf

[3]Multiple Object Tracking in Realtime. (2020, October 27). OpenCV. https://opencv.org/multiple-object-tracking-in-realtime/

[4]Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection (arXiv:2004.10934). arXiv. http://arxiv.org/abs/2004.10934

[5] MOT Challenge—Data. (n.d.). Retrieved October 14, 2022, from https://motchallenge.net/data/MOT16/