

Comparación multiplataforma de técnicas basadas en visión artificial para detección de personas en espacios abiertos

A multiplatform comparison of techniques based on artificial vision for detection of people in open spaces

Carlos Vicente Niño Rondón



Sergio Alexander Castro Casadiego



Byron Medina Delgado



Dinael Guevara Ibarra



Jhon Jairo Ramírez Mateus



Karla Cecilia Puerto López



Universidad Francisco de Paula Santander, Colombia

OPEN ACCESS

Recibido: 12/08/2020

Aceptado: 11/11/2020

Publicado: 01/04/2021

Correspondencia de autores:
carlosvicentnr@ufps.edu.co



Copyright 2020
by Investigación e
Innovación en Ingenierías

Resumen

Objetivo: Realizar una comparación multiplataforma entre los algoritmos de las técnicas de sustracción de fondo y de detectores en cascada por medio de una computadora personal y una placa Raspberry Pi 3B+ con Windows 10 y Debian GNU/LINUX respectivamente, en lenguaje de programación Python 3.7. **Metodología:** Se proponen tres etapas correspondientes a las mejoras en la imagen de video, la implementación de las técnicas de detección de personas y la evaluación del rendimiento de los algoritmos de dichas técnicas respecto al tiempo de respuesta, espacio de memoria requerido y aciertos en las detecciones. **Resultados:** La técnica de sustracción de fondo presenta una exactitud de 89.7% mientras que dicho valor para la técnica de detectores en cascada corresponde al 93.65%. Así mismo, la técnica de sustracción de fondo presenta mejor comportamiento respecto al tiempo de respuesta obteniendo 0.5934 segundos para Windows y 2.6338 segundos para Linux. **Conclusiones:** Tanto la técnica de sustracción de fondo como la técnica de detectores en cascada responden 5 veces más rápido en la computadora personal que en la placa Raspberry Pi 3B+, por su parte, el espacio de memoria requerido por la técnica de sustracción de fondo es 26.28% y 55% superior al espacio requerido por la técnica de detectores en cascada en el computador personal y en la placa Raspberry Pi 3B+ respectivamente.

Palabras clave: Comparación multiplataforma, Python, sustracción de fondo, detectores en cascada, rendimiento

Abstract

Objective: This article presents a cross-platform comparison between Python 3.7 background subtraction and cascade object detection algorithms using a Windows 10 personal computer and Debian GNU/LINUX running on a Raspberry Pi 3B+ board. **Methodology:** This study was divided in the following three stages: video image enhancements, implementation of the people detection techniques, and assessment of detection algorithms based on response times, memory space requirements, and successful detection rates. **Results:** The background subtraction technique has an accuracy of 89.7%, while this value for the cascade detector technique corresponds to 93.65%. Likewise, the background subtraction technique presents better performance regarding response time, obtaining 0.5934 seconds for Windows and 2.6338 seconds for Linux. **Conclusions:** Both the background subtraction and the cascade object detection technique responded 5 times faster on the personal computer than on the Raspberry Pi 3B+ board, whereas the memory space required by the background subtraction technique is 26.28% and 55% higher than the space required by the cascade object detection technique for the personal computer and the Raspberry Pi 3B+ board, respectively.

Keywords: Multiplatform comparisons, Python, background subtraction, cascade detection, performance.

Introducción

La visión artificial corresponde al conjunto de técnicas por las cuales se obtiene información del mundo real por medio de imágenes y videos para el análisis y seguimiento de la información obtenida. Se presenta como una alternativa a las funciones sensoriales y mentales humanas en donde las necesidades más relevantes corresponden a los procesos de localización, detección y rastreo de objetos, estimación de características faciales y reconocimiento de actividades humanas [1]. La detección de personas tiene aplicaciones en procesos de video vigilancia, rastreo visual y conteo automático de personas y si bien ha sido una temática abarcada desde las últimas décadas del siglo XX, actualmente se presentan retos en procesos en los que la imagen es variante y susceptible a cambios en la iluminación [2].

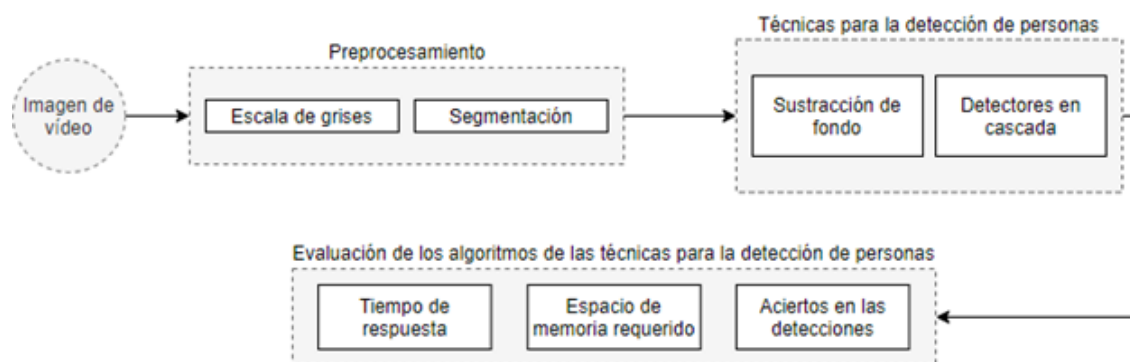
La técnica de sustracción de fondo toma relevancia puesto que en las aplicaciones de detección de personas se requiere la separación del objeto del fondo de la imagen para posteriormente realizar procesos más complejos como rastreo y conteo [3]; la técnica depende de las etapas de inicialización del fondo, detección en el cuadro principal y sostenimiento del fondo, donde se analizan cada uno de los objetos de la imagen principal y se determina si corresponden al fondo o al cuadro de la imagen [4]. Por otra parte, la técnica de detectores en cascada permite mediante filtrado progresivo asignar un conjunto de características dado a una clase con la que se encuentra una mayor similitud. Los clasificadores en cascada inicialmente se enfocaban a las partes inferior y superior del cuerpo humano, pero han sido mejorados y en la actualidad permiten distinguir características faciales y realizar el seguimiento a rostros [5].

En este artículo se muestra la implementación en dos plataformas (GNU/Linux por medio de una placa Raspberry Pi 3B+ y Windows 10 a través de un computador personal), de las técnicas para la detección de personas basado en sustracción de fondo así como detectores en cascada entrenados con dataset obtenido de videos de prueba disponibles en internet, y se muestra un comparativo de eficiencia de los mismos respecto al tiempo de respuesta de los algoritmos, requerimientos en la unidad central de procesos y detecciones erróneas (falsos positivos y falsos negativos).

Metodología

La metodología propuesta en el presente artículo se basa en tres etapas: mejoras a la imagen de video mediante el preprocesamiento de la imagen de video, aplicación de las técnicas para detección de personas y la evaluación del rendimiento de dichas técnicas [6]. La metodología se expone en la Figura 1 y posteriormente se describen cada una de las fases que intervienen en el proceso.

Figura 1. Metodología para la implementación y evaluación de las técnicas de detección de personas



Fuente: Elaboración propia

Preprocesamiento

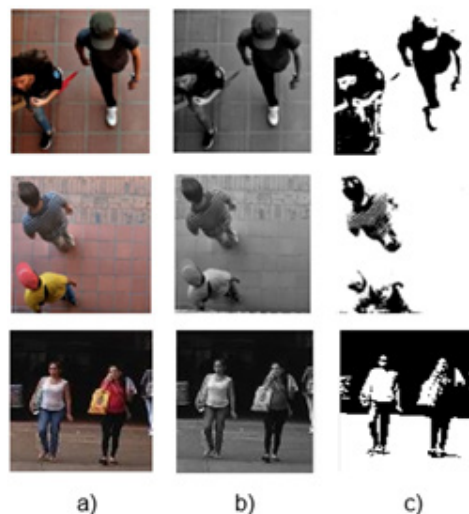
Las técnicas de preprocesamiento se aplican a imágenes de video capturadas para facilitar la sustracción de información, aun cuando la imagen de video presenta complejidad en el fondo y estructuras robustas [7]. A continuación, se detallan los procesos de binarización en escala de grises y segmentación, fundamentales en la etapa de preprocesamiento de imagen de video.

Binarización en escala de grises: Busca regiones dentro de las imágenes que funcionen como patrón de comportamiento en la secuencia de píxeles. Los píxeles toman valores comprendidos entre 0 y 255 [8]. La imagen resultante se logra haciendo un mezclado de los canales rojo, verde y azul con porcentajes de 30%, 59% y 11% respectivamente, como se describe en la ecuación (1).

$$Y = R * 0.3 + G * 0.59 + B * 0.11 \quad (1)$$

Segmentación: Divide en zonas la imagen dependiendo de la luminosidad presente en la misma. El éxito del proceso es dependiente de la calidad de imagen y se requiere que los objetos que intervengan en la imagen correspondan a formas cerradas.

Figura 2. Preprocesamiento. (a) Imagen original (b) Imagen binarizada c) Imagen normalizada.



Fuente: Elaboración propia.

En la Figura 2a se observan las imágenes originales analizadas, en la Figura 2b se muestran dichas imágenes sometidas al proceso de binarización en escala de grises y en la Figura 2c se aprecia la última etapa de la metodología de preprocesamiento propuesta correspondiente al proceso de segmentación. Una vez las imágenes de video se someten a este proceso, las técnicas de detección de personas tomarán características de las imágenes que permiten optimizar los resultados en dichas detecciones.

Técnicas para la detección de personas

En esta fase se detallan los fundamentos técnicos requeridas para la implementación de las técnicas de sustracción de fondo y detectores en cascada, una vez culminada la etapa de preprocesamiento de la imagen de video.

Sustracción de fondo: Realiza el seguimiento a los objetos en movimiento para extraerlos del fondo de la imagen, como se muestra en la ecuación (2).

$$\text{cuadro sustraído} = \text{cuadro actual} - \text{fondo de la imagen} \quad (2)$$

Se plantea el uso de filtros morfológicos acompañado de un proceso de umbralización de modo que los contornos detectados presenten la menor cantidad de detecciones erróneas y permitan agilizar el proceso.

a) Filtros morfológicos: Basados en operaciones no lineales de teoría de conjuntos que se relacionen con las características de forma de la imagen. En la Tabla 1 se presentan las operaciones morfológicas aplicadas.

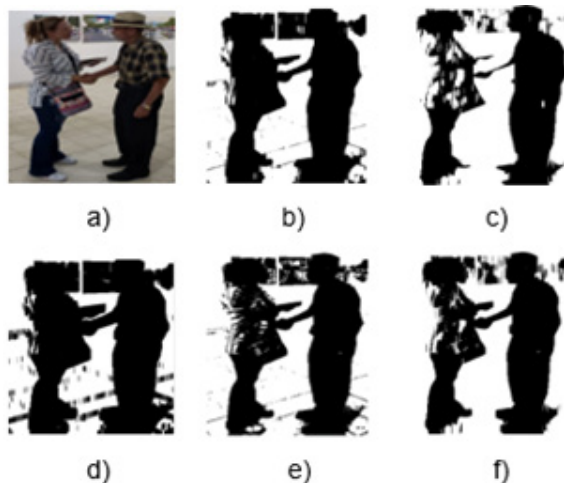
Tabla 1. Fundamento matemático de las operaciones morfológicas.

Operación	Ecuación
Dilatación	$A \oplus B = \bigcup_{b \in B} A_b$
Erosión	$A \ominus B = \bigcap_{b \in B} A_{-b}$
Apertura	$A \bullet B = (A \oplus B) \ominus B$
Cierre	$A \circ B = (A \ominus B) \oplus B$

Fuente: Elaboración propia

Las operaciones de dilatación y erosión son fundamentales para todo proceso de sustracción de fondo, no obstante, en dichos procesos las características mejoradas en la primera etapa se superponen en la segunda [9]. Es allí donde intervienen los filtrados de apertura y cierre, que mediante la combinación de operaciones de dilatación y erosión solucionan este problema y mejoran la calidad de la imagen[10].

Figura 3. Operaciones morfológicas. (a) Imagen original (b) imagen normalizada (c) imagen dilatada (d) imagen erosionada (e) imagen con filtro de apertura (f) imagen con filtro de cierre



Fuente: Elaboración propia.

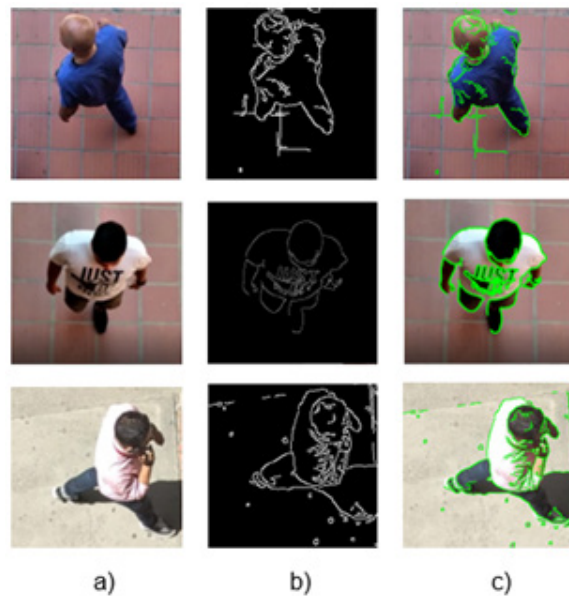
En la Figura 3a se muestra la imagen de prueba tomada al interior de la Universidad Francisco de Paula Santander, la Figura 3b corresponde a la imagen normalizada, requerida para la aplicación de filtros morfológicos, las Figuras 3c y 3d corresponden a las imágenes dilatadas y erosionadas respectivamente, y las Figuras 3e y 3f corresponden a la imagen con los filtros de apertura y cierre, donde se mejoran las características que los procesos de dilatación y erosión superponen.

b) Umbralización: Los cuadros se comparan con un valor de umbral establecido de modo que se rechacen los píxeles estáticos y sólo continúen en el procesamiento los objetos que se encuentran en el primer plano [11]; en la ecuación (3) se muestra la forma en cómo se realiza esta comparación.

$$fondo\ sustraido = \begin{cases} 1 & \text{si cuadro sustraido} > \text{umbral} \\ 0 & \text{si cuadro sustraido} < \text{umbral} \end{cases} \quad (3)$$

c) Distinción de contornos: Se relacionan dos etapas, la búsqueda y el dibujo de los contornos. Para la búsqueda de contornos se aplican los métodos de recuperación de contornos así como los métodos de aproximación [12]; en la presente propuesta se hace uso del método de aproximación simple que permite eliminar puntos redundantes disminuyendo así la capacidad de memoria a ocupar.

Figura 4. Distinción de contornos. (a) Imagen original (b) imagen con los contornos distinguidos (c) visualización de contornos

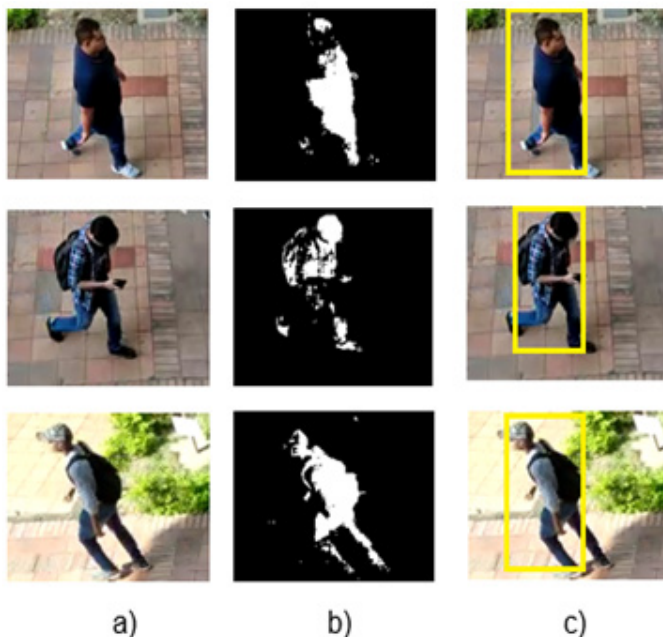


Fuente: Elaboración propia

En la Figura 4a se observan las imágenes originales a procesar, en la Figura 4b se muestran las imágenes con los bordes y contornos detectados, mientras que en la Figura 4c se visualizan dichos contornos aplicados a la imagen en color.

Una vez aplicados los procesos de filtrado morfológico, umbralización y distinción de contornos se puede visualizar el resultado de aplicación de la técnica de sustracción de fondo. En la Figura 5a se muestran las imágenes de prueba tomadas desde el edificio de Aula Sur de la Universidad Francisco de Paula Santander, en la Figura 4b se muestran a las personas sustraídas del fondo de la imagen y en la Figura 4c se denota la visualización de la persona detectada en la ventana principal de la plataforma.

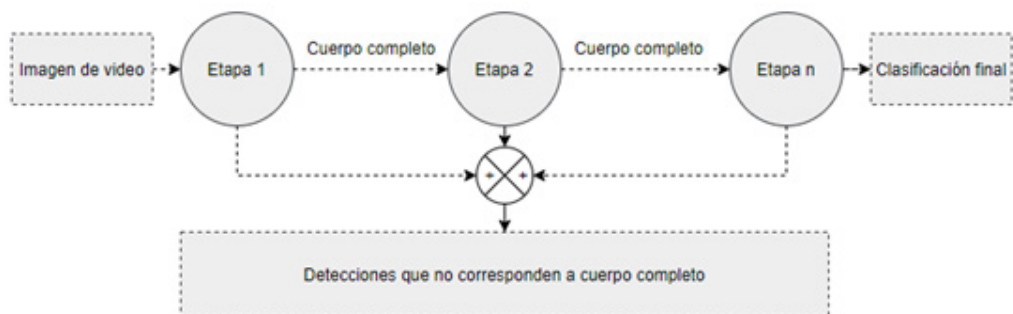
Figura 5. Sustracción de fondo. (a) Imagen original (b) persona sustraída del fondo de la imagen (c) visualización de la detección



Fuente: Elaboración propia

Detectores en cascada: Se evalúa por etapas de modo que progresivamente se obtenga nueva información sobre los datos de entrada y se pueda clasificar la información [13]. En la Figura 6 se observa el funcionamiento de un sistema de detector de cuerpo completo en cascada, en donde la imagen de video inicial se somete a una cantidad n de etapas de detección de modo que se eliminen redundancias y superposiciones en la imagen etapa tras etapa y solo las detecciones correspondientes al cuerpo completo de un humano se muestran en la visualización final. El rendimiento del detector es dependiente de los cambios en la iluminación del fondo de la imagen y de la calidad de las imágenes que conforman el dataset para entrenamiento.

Figura 6. Diagrama de funcionamiento de un sistema de detector de cuerpo completo en cascada.



Fuente: Elaboración propia

Un dataset es un conjunto de datos que proporciona imágenes que pueden usarse para entrenar algoritmos de aprendizaje automático [14]. El dataset utilizado en la presente investigación se obtiene de videos capturados previamente y contiene 400 imágenes de personas de diferente textura y color en diferentes posiciones sometidos a cambios en la luminosidad. En la Figura 7 se observan las primeras 105 de las imágenes utilizadas para el entrenamiento del clasificador.

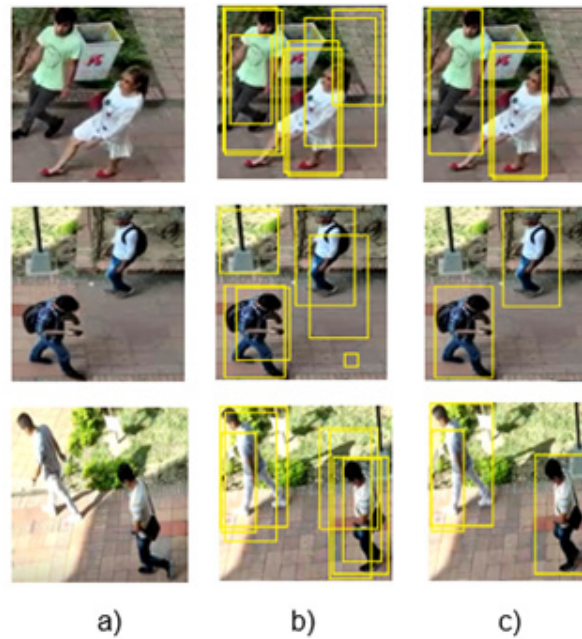
Figura 7. Dataset para el entrenamiento del clasificador



Fuente: Elaboración propia

Una vez obtenido el dataset se entrena el clasificador de modo que al someter el video de entrada al algoritmo de detector en cascada se mejore el rendimiento de la técnica [15]. En la Figura 8a se observan las imágenes de prueba tomadas desde el edificio de Aula Sur de la Universidad Francisco de Paula Santander, en la Figura 8b se observa el sondeo inicial realizado por el clasificador donde se observan todos los candidatos a ser seleccionados como personas en el cuadro principal, y en la Figura 8c se muestra la detección final que entrega el clasificador una vez la imagen de entrada se filtra progresivamente en cada una de las etapas.

Figura 8. Detectores en cascada. (a) Imagen original (b) Visualización de las detecciones



Fuente: Elaboración propia

Evaluación de los algoritmos de las técnicas de detección de personas

El método propuesto para la evaluación de las técnicas de detección se basa en el análisis del rendimiento de cada una de las técnicas de detección de personas [16]. La primera plataforma en la que se realiza dicho análisis corresponde a Debian GNU/LINUX con distribución Raspbian por medio de una placa Raspberry Pi 3B+ con procesador Broadcom BCM2837B0, frecuencia de reloj de 1.4GHz y memoria de 1GB. La segunda plataforma corresponde a Windows a través de una computadora personal con procesador Intel Core i7, frecuencia de reloj de 2.4GHz y memoria de 4GB. Los algoritmos de las técnicas de detección se codifican en el lenguaje de programación Python en su versión 3.7 mediante los entornos de desarrollo Thonny Python para Raspberry Pi 3B+ y Spyder para la computadora personal.

Se plantea determinar el rendimiento de los algoritmos de técnicas de sustracción de fondo y detectores en cascada mediante un método clásico cuantificando el tiempo de respuesta en ejecución [17], y el espacio de memoria requerido para 15 pruebas aleatorias. Adicionalmente se propone comparar el rendimiento de las técnicas mediante el número de detecciones acertadas, por lo que se requiere cotejar la cantidad de falsos positivos y falsos negativos que presenta cada técnica [18], y cuantificar su error respecto a la cantidad real de personas que aparecen en la imagen de video para determinar el porcentaje de acierto de cada una de las técnicas de detección.

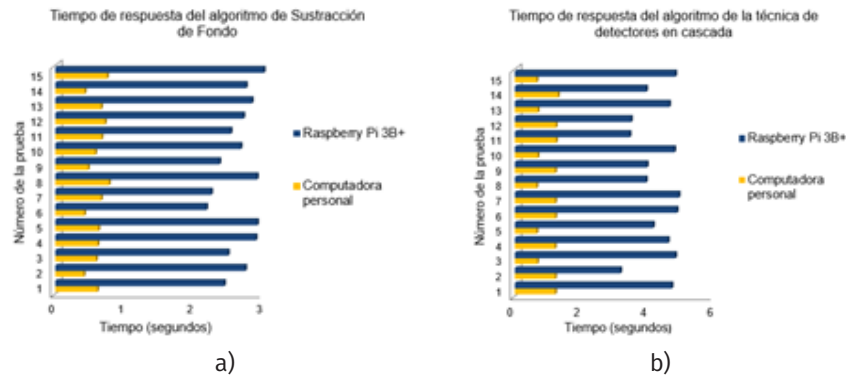
Resultados

A continuación se muestran los resultados obtenidos al realizar la comparativa de eficiencia multiplataforma de los algoritmos de las técnicas de sustracción de fondo y de detectores en cascada implementados en el lenguaje de programación Python.

Tiempo de respuesta

En la Figura 9 se observan las gráficas correspondientes al tiempo de respuesta de los algoritmos de las técnicas de detección de personas implementadas en Windows y en Debian GNU/LINUX. En la Figura 9a se observa la gráfica de comportamiento del tiempo de respuesta de la técnica de sustracción de fondo que presenta un tiempo promedio de 0.5934 segundos en la computadora personal y 2.6338 segundos en la placa Raspberry Pi 3B+. De igual forma, en la Figura 9b se muestra el tiempo de respuesta de la técnica de detectores en cascada que en promedio corresponde a 0.9988 segundos en la computadora personal y 4.2736 segundos para la placa Raspberry Pi 3B+.

Fig. 9. Tiempo de respuesta. (a) Sustracción de fondo (b) Detectores en cascada

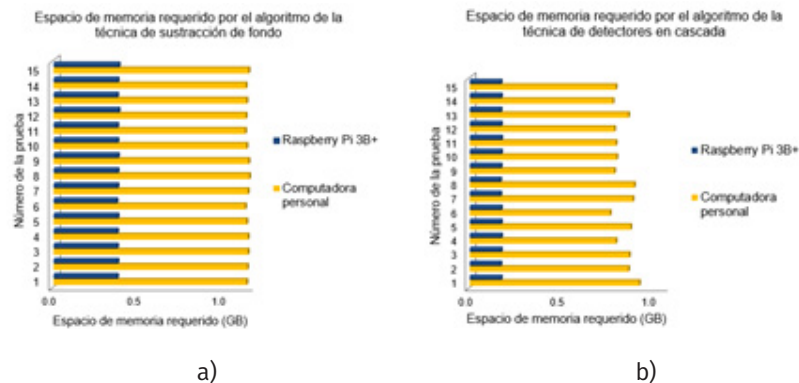


Fuente: Elaboración propia

Espacio de memoria requerido

En la Figura 10, se aprecian las gráficas que describen el espacio de memoria que requieren los algoritmos de las técnicas de detección de personas implementados a través de Python en Windows y en Debian GNU/LINUX. La Figura 10a muestra la gráfica correspondiente al espacio de memoria utilizado por la técnica de sustracción de fondo, que en promedio requiere de 1.1389 GB en la computadora personal y 0.3776 GB en la placa Raspberry Pi 3B+. Así mismo, la Figura 10b indica el espacio de memoria requerido para la técnica de detectores en cascada con requerimientos de memoria promedio de 0.8325 GB para la computadora personal y 0.1709 GB para la placa Raspberry Pi 3B+.

Fig. 10. Espacio de memoria requerido. (a) Sustracción de fondo (b) Detectores en cascada



Fuente: Elaboración propia

Aciertos en las detecciones

La Tabla 2, muestra la cantidad de falsos positivos y falsos negativos y la cuantificación del error respecto al número real de personas que intervienen en la imagen de video.

Tabla 2. Valores de detecciones acertadas y erróneas para las técnicas de detección de personas

	Falsos positivos	Falsos negativos	Total de detecciones	Número real de personas	Error por falsos positivos	Error por falsos negativos
Sustracción de fondo	3	2	36	35	8.57%	5.71%
Detectores en cascada	2	1	36	35	5.71%	2.78%

Fuente: Elaboración propia

Una vez consolidados los errores de falsos positivos y falsos negativos por independiente, se obtiene el valor del error medio para cada una de las técnicas de detección de personas, tal y como se muestra a continuación.

$$Error\ medio\ de\ sustracción\ de\ fondo = \sqrt{(0.0857)^2 + (0.0571)^2} = 10.30\% \quad (4)$$

$$Error\ medio\ de\ detectores\ en\ cascada = \sqrt{(0.0571)^2 + (0.0278)^2} = 6.35\% \quad (5)$$

En la Figura 1, se observan las gráficas correspondientes al rendimiento de las técnicas respecto al número de personas detectadas. Las técnicas de sustracción de fondo y detectores en cascada presentan una exactitud de 89.7% y 93.65% respectivamente.

Fig. 11. Rendimiento respecto al número de detecciones. a) Sustracción de fondo b) detectores en cascada



Fuente: Elaboración propia

Conclusiones

Las pruebas realizadas para las técnicas de sustracción de fondo y detectores en cascada en la plataforma de Windows mediante una computadora personal arrojan un tiempo de respuesta promedio aproximado 5 veces menor respecto al empleado en Linux por medio de la placa embebida Raspberry Pi 3B+.

La técnica de sustracción de fondo requiere 26.28% más de espacio disponible en el computador personal respecto a la técnica de detectores en cascada. En la placa Raspberry Pi 3B+, el espacio de memoria requerido por la técnica de sustracción de fondo supera en 55% al requerido por la técnica de detectores en cascada.

Los filtros morfológicos aplicados en la técnica de sustracción de fondo y el entrenamiento realizado por medio del dataset en la técnica de detectores en cascada permiten mejorar los procesos de visualización y conteo de personas.

Referencias Bibliográficas

1. M. Leo, G. Medioni, M. Trivedi, T. Kanade, and G. M. Farinella, "Computer vision for assistive technologies," *Comput. Vis. Image Underst.*, vol. 154, pp. 1–15, 2017. DOI: <https://doi.org/10.1016/j.cviu.2016.09.001>
2. D. L. Siqueira and A. Manso Correa MacHado, "People Detection and Tracking in Low Frame-rate Dynamic Scenes," *IEEE Lat. Am. Trans.*, vol. 14, no. 4, pp. 1966–1971, 2016. DOI: <https://doi.org/10.1109/TLA.2016.7483541>
3. B. García-García, T. Bouwmans, and A. J. Rosales Silva, "Background subtraction in real applications: Challenges, current models and future directions," *Comput. Sci. Rev.*, vol. 35, p. 100204, 2020. DOI: <https://doi.org/10.1016/j.cosrev.2019.100204>
4. A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Comput. Vis. Image Underst.*, vol. 122, pp. 4–21, 2014. DOI: <https://doi.org/10.1016/j.cviu.2013.12.005>
5. H. A. M. Alhamzawi, "Faces and eyes Detection in Digital Images Using Cascade Classifiers," *Comput. Eng. Appl. J.*, vol. 7, no. 1, pp. 57–66, 2018. DOI: <https://doi.org/10.18495/comengapp.v7i1.222>
6. G. Cavanzo, M. Pérez, and F. Villavisan, "Medición de eficiencia de algoritmos de visión artificial implementados en raspberry pi y ordenador personal mediante Python," *Ingenium*, vol. 18, no. 35, pp. 105–119, 2017. DOI: <https://doi.org/10.21500/01247492.3218>
7. Ş. Öztürk and B. Akdemir, "Effects of Histopathological Image Pre-processing on Convolutional Neural Networks," *Procedia Comput. Sci.*, vol. 132, no. Iccids, pp. 396–403, 2018. DOI: <https://doi.org/10.1016/j.procs.2018.05.166>
8. J. A. Taquíá-Gutiérrez, "El procesamiento de imágenes y su potencial aplicación en empresas con estrategia digital," *Interfases*, vol. 0, no. 010, p. 11, 2017. DOI: <https://doi.org/10.26439/interfases2017.n10.1767>
9. B. Rodríguez-Cuenca, S. García-Cortés, C. Ordóñez, and M. C. Alonso, "Morphological operations to extract urban curbs in 3d mls point clouds," *ISPRS Int. J. Geo-Information*, vol. 5, no. 6, 2016. DOI: <https://doi.org/10.3390/ijgi5060093>

10. B. Ding, G. Wen, C. Ma, and X. Yang, "Target recognition in synthetic aperture radar images using binary morphological operations," *J. Appl. Remote Sens.*, vol. 10, no. 4, p. 046006, 2016. DOI: <https://doi.org/10.1117/1.JRS.10.046006>
11. M. Elhoseny, "Multi-object Detection and Tracking (MODT) Machine Learning Model for Real-Time Video Surveillance Systems," *Circuits, Syst. Signal Process.*, 2019. DOI: <https://doi.org/10.1007/s00034-019-01234-7>
12. P. Suárez and M. Villavicencio, "Detección de Contornos utilizando el Algoritmo Canny en Imágenes Cross-Espectrales Fusionadas," *Enfoque UTE*, vol. 8, no. 1, p. 16, 2017. DOI: <https://doi.org/10.29019/enfoqueute.v8n1.127>
13. K. Mahkonen, T. Virtanen, and J. Kämäräinen, "Cascade of Boolean detector combinations," *Eurasip J. Image Video Process.*, vol. 2018, no. 1, 2018. DOI: <https://doi.org/10.1186/s13640-018-0303-9>
14. J. Parraga-Alava, K. Cusme, A. Loor, and E. Santander, "RoCoLe: A robusta coffee leaf images dataset for evaluation of machine learning based methods in plant diseases recognition," *Data Br.*, vol. 25, 2019. DOI: <https://doi.org/10.1016/j.dib.2019.104414>
15. R. A. Maulana Budiman, B. Achmad, Faridah, A. Arif, Nopriadi, and L. Zharif, "Localization of white blood cell images using Haar Cascade classifiers," *Proc. 2016 1st Int. Conf. Biomed. Eng. Empower. Biomed. Technol. Better Futur. IBIOMED 2016, 2017*. DOI: <https://doi.org/10.1109/IBIOMED.2016.7869822>
16. G. Bizai, A. Salvatelli, J. Calderón and B. Drozdowicz, "Implementación y evaluación de algoritmos para la visualización de imágenes de campos de luz", *Ventana Informatica*, no. 39, pp. 17-33, 2019. DOI: <https://doi.org/10.30554/ventanainform.39.3306.2018>
17. N. Maya, A. Nuñez Bedoya and H. Romo Romero, "Análisis de rendimiento de algoritmos de reconocimiento de placas de números de vehículos desarrollado mediante la transformación de ondas discretas y la correlación de imagen digital", *Investigación e innovación en ingenierías*, vol. 7, no. 1, 2019. DOI: <https://doi.org/10.17081/invinno.7.1.2990>
18. G. A. Duarte-Medrano, D. Herrera-Mora, and F. I. Téllez-Ávila, "Pruebas diagnósticas: aspectos básicos de lo que el endoscopista debe saber," *Endoscopia*, vol. 28, no. 3, pp. 128-132, 2016. DOI: <https://doi.org/10.1016/j.endomx.2016.09.001>