

## Signal processing for people detection in crowded areas

Procesamiento de señales para detección de personas en aglomeraciones

Carlos Vicente Niño-Rondón<sup>1\*</sup>, Sergio Alexander Castro-Casadiago<sup>2</sup>

<sup>1\*</sup> Ingeniero Electrónico, carlosvicentemr@ufps.edu.co, <https://orcid.org/0000-0002-3781-4564>, Universidad Francisco de Paula Santander, Cúcuta, Colombia.

<sup>2</sup> MSc en Ingeniería Electrónica, sergio.castroc@ufps.edu.co, <https://orcid.org/0000-0003-0962-9916>, Universidad Francisco de Paula Santander, Cúcuta, Colombia.

**Cómo citar:** C. V. Niño-Rondón y S. A. Castro-Casadiago, "Procesamiento de señales para detección de personas en aglomeraciones", *Respuestas*, vol. 26, n.º 3, p. 33-43, sep. 2021. <https://doi.org/10.22463/0122820X.4806>

Received on March 22, 2021 - Approved on August 17, 2021

### ABSTRACT

#### Keywords:

People detection,  
Background subtraction,  
Raspberry Pi, Open-  
Source

This paper addresses the development and implementation of a signal processing system for detecting and counting people in open spaces, using background subtraction techniques with a Raspberry Pi 3B+ embedded board and the Python programming language. The methodology employed includes image conversion to grayscale, background segmentation using the Background Subtractor MOG2 algorithm, Gaussian smoothing filtering, and adaptive thresholding with the Otsu method, along with morphological techniques to enhance detection quality and contour detection for identifying objects. In the image capture phase, factors such as height, tilt angle, and environmental luminosity are considered to ensure the quality of the collected data. Grayscale conversion assigns values between 0 and 255 to pixels, and background subtraction uses Gaussian distributions to differentiate between moving objects and the background. Gaussian smoothing filters are applied to reduce noise, while Otsu's thresholding adapts the threshold to the specific characteristics of each image. Finally, morphological operations refine segmentation, and the simple approximation method is used for contour detection. The system was evaluated with videos captured from four buildings at the Universidad Francisco de Paula Santander and two public areas in Cúcuta, showing detection rates between 87.14% and 93.33% at the university and between 88.89% and 90.51% in public areas.

### RESUMEN

#### Palabras clave:

Detección de personas,  
sustracción de fondo,  
Raspberry Pi, Código  
abierto

En este documento se aborda el desarrollo y la implementación de un sistema de procesamiento de señales para la detección y conteo de personas en espacios abiertos, utilizando la técnica de sustracción de fondo con una placa embebida Raspberry Pi 3B+ y el lenguaje de programación Python. La metodología empleada incluye la conversión de imágenes a escala de grises, segmentación de fondo mediante el algoritmo Background Subtractor MOG2, filtrado mediante suavizado gaussiano, y umbralización adaptativa con el método de Otsu, además de la aplicación de técnicas morfológicas para mejorar la calidad de las detecciones y la búsqueda de contornos para identificar objetos. En la fase de captura de imágenes, se considera la altura, el ángulo de inclinación y la luminosidad del entorno para garantizar la calidad de los datos recolectados. La conversión a escala de grises asigna valores entre 0 y 255 a los píxeles, y la sustracción de fondo se realiza usando distribuciones gaussianas para diferenciar entre objetos en movimiento y el fondo. El filtro de suavizado gaussiano se aplica para reducir el ruido, mientras que la umbralización de Otsu permite adaptar el umbral a las características específicas de cada imagen. Finalmente, se utilizan operaciones morfológicas para refinar la segmentación y el método de aproximación simple para la detección de contornos. El sistema fue evaluado con videos capturados en cuatro edificios de la Universidad Francisco de Paula Santander y en dos áreas públicas de Cúcuta, mostrando tasas de detección entre 87.14% y 93.33% en la universidad, y entre 88.89% y 90.51% en las zonas públicas.

\*Corresponding author.

E-mail Address: carlosvicentemr@ufps.edu.co (Carlos Vicente Niño-Rondón)

Peer review is the responsibility of the Universidad Francisco de Paula Santander.  
This is an article under the license CC BY-NC 4.0



## Introducción

Las técnicas para la detección de personas tienen como objetivo identificar y extraer objetos presentes en una escena de video para su uso en una aplicación específica [1], [2]. Entre las metodologías más utilizadas se encuentran la sustracción de fondo[3], el histograma de gradientes orientados [4], los detectores en cascada y el flujo óptico, entre otros [5].

La aglomeración de personas en ambientes urbanos se refiere al agrupamiento masivo en áreas públicas, a menudo debido al crecimiento demográfico y urbanístico. San José de Cúcuta, una ciudad colombiana situada en la frontera, ha experimentado históricamente efectos negativos causados por el rápido aumento de la población y la falta de medidas para mitigar este problema [6].

La población de San José de Cúcuta y su área metropolitana alcanzó las 944,938 personas en 2019, y se espera que para mediados de 2020 supere el millón de habitantes, lo que representa un crecimiento cercano al 9%, convirtiéndola en la ciudad de mayor crecimiento poblacional en el país, superando a Cartagena y Barranquilla, que presentan un crecimiento de 5.7% y 5.6%, respectivamente [7].

El rápido aumento poblacional en la capital de Norte de Santander se debe a la falta de medidas gubernamentales para controlar la dinámica poblacional y al ingreso masivo de inmigrantes atraídos por la proximidad de Cúcuta a sus países de origen. Esto ha afectado negativamente aspectos fundamentales como la economía, la sociedad, la salud, la educación y la seguridad, ya que la ciudad no está preparada para satisfacer las necesidades de sus habitantes. Como resultado, Cúcuta se encuentra entre las ciudades colombianas con los índices más altos de desempleo, acceso limitado al sistema de salud e inseguridad.

En años recientes, se han desarrollado técnicas para la detección y monitoreo de personas, como clasificadores en cascada para detección de rostros, detectores de ojos y caras, reconocimientos faciales basados en histogramas de gradientes y métodos de reconocimiento de acciones humanas[8], [9], [10] [11], [12], [13]. La efectividad de estas técnicas depende de la ubicación de las personas con respecto al dispositivo de captura de video. Aunque logran una precisión superior al 90%, su aplicación se limita a situaciones en las que las personas están frontalmente frente a la cámara [14], [15], ya que los clasificadores en cascada se entrenan con características físicas específicas de las personas y adaptan casos de estudio a los modelos preentrenados [12], [16], [17]. Además, los métodos basados en histogramas de gradientes orientados requieren más recursos computacionales, como mayor memoria y capacidad de procesamiento de la unidad central de procesos [18].

El desarrollo de un sistema embebido para detectar en tiempo real la presencia de personas en ambientes controlados [19] resalta la utilización de un descriptor y un clasificador como componentes esenciales del detector de personas en tiempo real. El descriptor identifica las características de los objetos deseados, mientras que el clasificador se entrena utilizando conjuntos de imágenes de personas obtenidas de bases de datos como INRIA y PASCAL. El proyecto subraya la importancia de las librerías de OpenCV tanto para crear el vector de imágenes de personas como para entrenar el clasificador. Para desarrollar la aplicación, se emplearon herramientas como Raspberry Pi 3 y Python, junto con la librería de código abierto OpenCV, lo que facilitó el funcionamiento del sistema. Además, se enfatiza la importancia de ubicar el dispositivo de captura de video en lugares que cubran la mayor cantidad de espacio posible y donde las técnicas no se vean afectadas por variaciones en la iluminación.

Asimismo, el diseño e implementación de un sistema de visión artificial para determinar condiciones de fatiga en una persona mediante el índice Perclos, utilizando OpenCV y una tarjeta Raspberry Pi 3 [20] propone el uso de técnicas de reconocimiento facial como una alternativa a los métodos tradicionales de detección de personas. El proyecto detalla cómo implementar el FaceRecognizer disponible en las últimas versiones de OpenCV. El hardware diseñado incluye un sistema de iluminación basado en LEDs, compuesto por una cámara infrarroja y un sistema de notificación mediante una señal sonora. El código implementado utiliza el brillo de la imagen de entrada como referencia inicial para decidir si se debe iniciar con iluminación infrarroja o normal para una detección facial eficiente. Además, el estudio aborda consideraciones para la instalación del sistema, como la inclinación máxima de la cámara de video y la alimentación de energía del módulo.

De igual forma, "Sistema de conteo automático de flujo de personas por medio de visión artificial" [21] se implementó en una placa Raspberry Pi utilizando el lenguaje de programación Python y el paquete de OpenCV. El sistema emplea técnicas como el Histograma de Gradientes Orientados con la Máquina de Vectores de Soporte y la sustracción de fondo con extracción de blobs. Para la implementación de estos algoritmos, se realizó una caracterización del ambiente controlado, considerando factores como la iluminación, la distancia mínima para detección y el desplazamiento del peatón según la dirección.

El Desarrollo de un sistema de visión artificial para la detección de aglomeración de personas en un semáforo [22] está implementado en Matlab utilizando toolbox disponibles para visión por computadora. El sistema emplea el algoritmo de Viola-Jones en el detector de objetos en cascada para detectar caras (FrontalFaceCART) y la parte superior del cuerpo (Upperbody). Además, utiliza el objeto detector de personas (peopleDetector) basado en histogramas de gradientes orientados (HOG) y Máquinas de Vectores de Soporte (SVM) entrenadas. El proyecto integra los tres algoritmos mencionados (FrontalFaceCART, Upperbody y peopleDetector) para mejorar la precisión de las detecciones y reducir el error. También se compara el rendimiento y la eficiencia de estas técnicas con la técnica de sustracción de fondo utilizada en la propuesta actual, y la de histogramas de gradientes orientados.

En el presente artículo se detalla un sistema de procesamiento de señales para la detección y conteo de personas en espacios abiertos mediante la técnica de sustracción de fondo. El prototipo fue implementado en Python mediante una placa embebida Raspberry Pi 3B+, analizando capturas de video tomadas desde 4 edificios de la Universidad Francisco de Paula Santander, así como en 2 zonas de espacio público de la ciudad de Cúcuta.

## **Materiales y Métodos**

La metodología para el sistema de procesamiento de señales se basa en la implementación de procesos de visión por computadora, que se detallan a continuación:

## Captura de Imágenes

Las imágenes se capturan usando un dispositivo de captura de video (cámara) ubicado en un entorno controlado o en espacios públicos. Se deben considerar factores como la altura, el ángulo de inclinación y el nivel de luminosidad del entorno para optimizar la calidad de las imágenes.

## Conversión de la Imagen a Escala de Grises

Usando una matriz bidimensional, se recopila información sobre la intensidad de la luz en la imagen. Al convertir la imagen a escala de grises, se asignan valores entre 0 y 255 a cada píxel. El estándar de televisión en color NTSC define cómo convertir los colores rojo, verde y azul a escala de grises, como se indica en la ecuación 1, Donde, R hace referencia al color rojo, G refiere al color verde y B, corresponde al color azul.

$$\text{Imagen de salida en gris} = (R*0.39+(G*0.59)+(B*0.11)) \quad (1)$$

## Segmentación del fondo por Modelado de Mezclas Gaussianas GMM

Se utiliza el algoritmo Background Subtractor MOG2, que modela los valores de los píxeles y sus vecinos como una distribución Gaussiano. Este enfoque permite identificar qué píxeles corresponden a objetos en movimiento y cuáles forman parte del fondo de la imagen. Primero, se analiza cómo cambian los píxeles durante el intervalo de tiempo considerado, teniendo en cuenta el comportamiento previo de los píxeles en la imagen. La ecuación 2 muestra esta relación, donde I representa la secuencia de imágenes en procesamiento. Asimismo, en la ecuación 3 se muestra la probabilidad de conocer el valor del píxel de la imagen procesada; donde  $W_{i,t}$ , representa la estimación de la aproximación Gaussiana en el instante de tiempo analizado,  $\mu_{i,t}$ , refiere a la aproximación gaussiana en función del tiempo y donde K y  $\Sigma_{i,t}$ ; corresponden al número de distribuciones gaussianas y la matriz de covarianza respectivamente. Si al realizarse el barrido por las K distribuciones, no se obtiene coincidencia con el valor del píxel actual, la distribución se reemplaza por el valor actual del píxel, con peso de priorización mínimo y varianza alta. En la ecuación 4, se muestra la forma en cómo se ajustan los pesos de prioridad, donde  $\alpha$  es la razón de aprendizaje, y  $M_{k,t}$  funciona a modo binario, donde toma el valor de 1 si hay coincidencia, y 0 en casos contrarios.

$$\{X1, Xt\} = \{I(Xo, Yo, s); 1 < i < t\} \quad (2)$$

$$p(x_t) = \sum_{i=1}^K W_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (3)$$

$$W_{k,t} = (1 - \alpha)W_{k,t-1} + \alpha(M_{k,t}) \quad (4)$$

## Filtro de Suavizado Gaussiano

En la ecuación 5, se muestra la forma en cómo se obtienen las distribuciones con desplazamientos a lo largo del eje x.

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \quad (5)$$

Dicha distribución debe ser truncada, ya que se extiende con tendencia al infinito hacia el eje negativo del plano, por lo que dicho valor para truncamiento debe estar comprendido en la relación dada por:  $X \in [-3\sigma, 3\sigma]$ .

### *Filtro por operaciones morfológicas*

Está basado en operaciones de conjuntos. Los filtros comúnmente utilizados corresponden a los procesos de dilatación, erosión, apertura y cierre [23]. En las ecuaciones 6, 7, 8 y 9 respectivamente, se muestran las expresiones matemáticas de cada una de las operaciones mencionadas.

$$A \oplus B = \bigcup_{b \in B} A_b \quad (6)$$

$$A \ominus B = \bigcap_{b \in B} A_{-b} \quad (7)$$

$$A \circ B = (A \ominus B) \oplus B \quad (8)$$

$$A \bullet B = (A \oplus B) \ominus B \quad (9)$$

### *Búsqueda de contornos*

Se requiere de métodos de aproximación de contornos, en el que se denotan los límites de determinada forma. Los métodos de aproximación comúnmente utilizados corresponden al aleatorio y al simple, donde en imágenes simples como formas geométricas (rectángulos y cuadrados), se obtienen 734 contornos y 4 contornos respectivamente. Es por esto, que se utiliza el método de aproximación simple, ya que ahorra espacio de memoria respecto al método aleatorio.

## **Resultados y discusión**

En la sección superior de la *figura 1*, se muestran tomas de video capturados al interior de la Universidad Francisco de Paula Santander y en zonas céntricas de la ciudad de Cúcuta. Por otra parte, en la sección inferior se muestra la conversión de dichas imágenes a escala de grises.



**Figura 1.** Conversión de la imagen de entrada a escala de grises

Asimismo, en la *figura 2* se muestra la segmentación de las imágenes previamente convertidas a escala de grises.

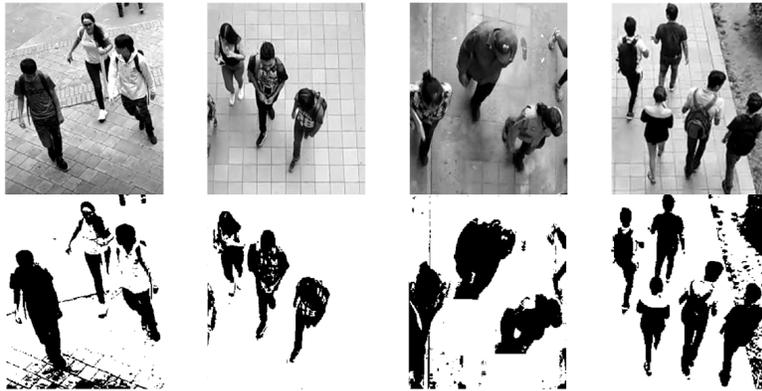


Figura 2. Segmentación de las imágenes en proceso.

En la *figura 3*, se observa la imagen previa para filtrado; así mismo, se observan filtrados con  $5 \times 5$  y  $11 \times 11$  de tamaño de núcleo para la matriz de Gauss respectivamente, mientras que la imagen de la sección derecha, presenta una matriz de  $21 \times 21$  siendo la utilizada debido a su mejoría en el suavizado de imagen.



Figura 3. Suavizado por filtrado gaussiano

Se utilizó el método adaptativo de Otsu, con umbral de previamente establecido en caso de que no se realice la adaptación autónoma. En la *figura 4*, se observa la imagen con el valor de umbral establecido y el histograma de la imagen respectivamente. Así mismo, en la penúltima y última sección se muestra la imagen con el umbral automático y el histograma obtenido para dicha imagen.

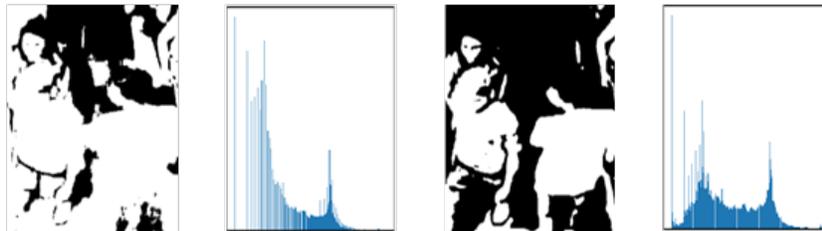


Figura 4. Umbralización convencional vs Umbralización adaptativa

Se implementó el método de aproximación de contornos simple. En la figura 5 se observa la imagen original, así como la imagen procesada para la posterior distinción de contornos. Así mismo, se muestra umbralización inversa en las detecciones sin método de aproximación de contornos y con el método de aproximación simple respectivamente, obteniendo detecciones con menor ruido en esta última metodología.



Figura 5. Distinción de contornos en las detecciones con umbralización inversa

En la figura 6 se muestra la representación del procesamiento con sus etapas inmersas para el conteo de las personas.

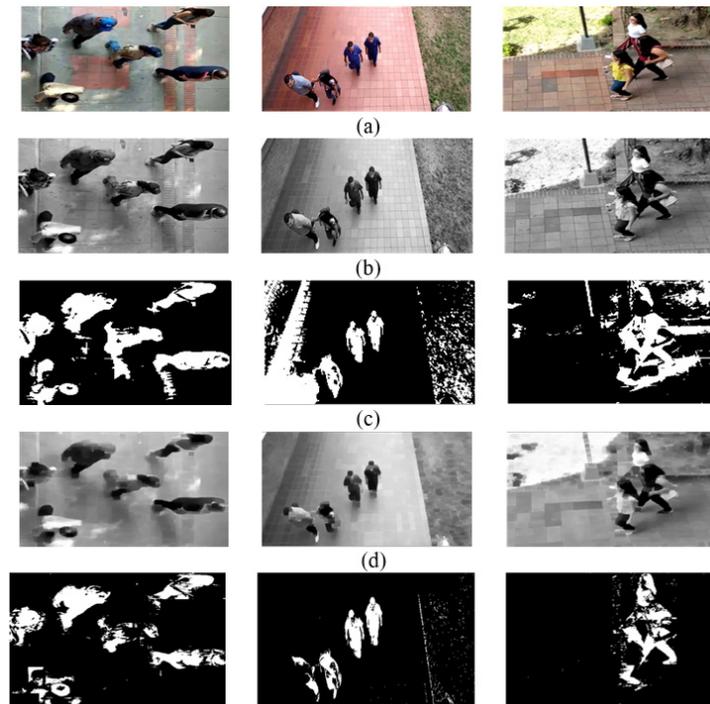


Figura 6. Procesamiento para el conteo de personas

Asimismo, en la figura 7 se muestra el desempeño de la Unidad Central de Procesos (CPU) al procesar los videos grabados en la Universidad Francisco de Paula Santander. Los resultados indican que el uso de la CPU para el video del Edificio Fundadores varía entre el 38 % y el 43 %. Para el video del Edificio Semipesados, el uso de la CPU está en el rango de 39 % a 44 %. El procesamiento del video desde el Edificio

Aula Sur Bloque A requiere entre el 39 % y el 43 % de la capacidad de la CPU, mientras que, para el video del Edificio Aula Sur Bloque F, el porcentaje de uso fluctúa entre el 35 % y el 44 %.

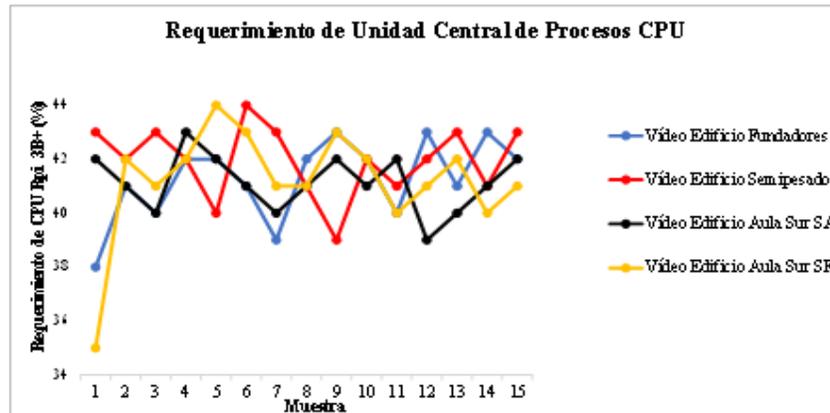


Figura 7. Requerimientos de CPU en ambiente controlado.

De igual forma, se muestra el requerimiento de CPU para el procesamiento de los videos tomados en las zonas de espacio público de la ciudad de Cúcuta. Para el video tomado desde la ClI 8<sup>a</sup> con Av. 6<sup>a</sup> y 7<sup>a</sup>, el requerimiento fluctúa entre 40 % y 45 %. Así mismo, para el video tomado desde la Av. 6<sup>a</sup> con ClI 6<sup>a</sup> y 7<sup>a</sup>, dicho valor se encuentra en el rango de 37 % y 45 %.

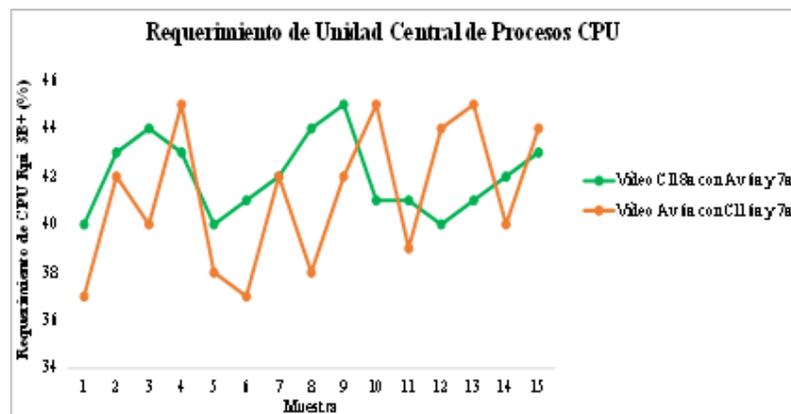


Figura 8. Requerimientos de CPU en zonas de espacio público.

## Conclusiones

El uso de la Raspberry Pi 3B+ y el lenguaje Python reduce significativamente los costos de implementación en comparación con herramientas de hardware y software más costosas y con licencias caras. La técnica de sustracción de fondo para detección de movimiento resulta más efectiva en identificar una amplia gama de objetos en comparación con técnicas como histogramas de gradientes orientados y detectores en cascada, que pueden pasar por alto ciertos objetos debido a su ubicación respecto al dispositivo de captura. El modelado de mezclas gaussianas y el umbral adaptativo de Otsu optimizan el rendimiento del sistema, logrando tasas

de detección entre 87.14% y 93.33% en videos tomados en la Universidad Francisco de Paula Santander, y entre 88.89% y 90.51% en zonas públicas de Cúcuta, superando las tasas de proyectos similares en ciudades como Londres y Roma.

Durante la pandemia de COVID-19, los videos capturados en espacios públicos evidencian el incumplimiento de las medidas de confinamiento impuestas por el gobierno. El procesamiento de estos videos con la Raspberry Pi 3B+ utiliza entre 35% y 45% de la capacidad de la CPU y tiene un tiempo de respuesta promedio de 0.43 segundos, lo cual es adecuado considerando que el dispositivo opera a 1.4 GHz con 1 GB de RAM.

## Referencias

- [1] M. Leo, G. Medioni, M. Trivedi, T. Kanade, and G. M. Farinella, “Computer vision for assistive technologies,” *Computer Vision and Image Understanding*, vol. 154, pp. 1–15, 2017, doi: 10.1016/j.cviu.2016.09.001.
- [2] X. Li and Y. Shi, “Computer vision imaging based on artificial intelligence,” in *Proceedings - 2018 International Conference on Virtual Reality and Intelligent Systems, ICVRIS 2018*, IEEE, 2018, pp. 22–25. doi: 10.1109/ICVRIS.2018.00014.
- [3] N. S. Sakpal and M. Sabnis, “Adaptive Background Subtraction in Images,” in *2018 International Conference On Advances in Communication and Computing Technology, ICACCT 2018*, IEEE, 2018, pp. 439–444. doi: 10.1109/ICACCT.2018.8529323.
- [4] S. A. Korkmaz, A. Akcicek, H. Binol, and M. F. Korkmaz, “Recognition of the stomach cancer images with probabilistic HOG feature vector histograms by using HOG features,” in *SISY 2017 - IEEE 15th International Symposium on Intelligent Systems and Informatics, Proceedings*, 2017, pp. 339–342. doi: 10.1109/SISY.2017.8080578.
- [5] J. Hernández Santiago, J. S. Ruiz Castilla, C. H. Moreno Montiel, and B. Hernández Santiago, “Segmentación de placas vehiculares usando Haar-AdaBoost y Clustering,” *Research in Computing Science*, vol. 147, no. 5, pp. 269–279, 2018, doi: 10.13053/rcs-147-5-20.
- [6] A. J. A. Barreto and C. P. A. Barreto, “Some commercial customs in the footwear industry of the municipality of San José de Cúcuta (Colombia),” *Revista de Derecho Privado*, no. 36, pp. 319–331, 2019, doi: 10.18601/01234366.n36.12.
- [7] Diario La Opinión, “Cifras del censo no les cuadran a alcaldes del área metropolitana de Cúcuta,” *Actualidad*. Accessed: Jun. 29, 2020. [Online]. Available: <https://www.laopinion.com.co/economia/cifras-del-censo-no-les-cuadran-alcaldes-del-area-metropolitana-de-cucuta-182061#OP>
- [8] V. Fremont, M. T. Bui, D. Boukerroui, and P. Letort, “Vision-based people detection system for heavy machine applications,” *Sensors (Switzerland)*, vol. 16, no. 1, pp. 1–30, 2016, doi: 10.3390/s16010128.

- [9] E. N. Kajabad and S. V. Ivanov, "People Detection and Finding Attractive Areas by the use of Movement Detection Analysis and Deep Learning Approach," *Procedia Comput Sci*, vol. 156, pp. 327–337, 2019, doi: 10.1016/j.procs.2019.08.209.
- [10] A. Gavriilidis, J. Velten, S. Tilgner, and A. Kummert, "Machine learning for people detection in guidance functionality of enabling health applications by means of cascaded SVM classifiers," *J Franklin Inst*, vol. 355, no. 4, pp. 2009–2021, 2018, doi: 10.1016/j.jfranklin.2017.10.008.
- [11] C. H. Setjo, B. Achmad, and Faridah, "Thermal image human detection using Haar-cascade classifier," in *Proceedings - 2017 7th International Annual Engineering Seminar*, InAES 2017, 2017, doi: 10.1109/INAES.2017.8068554.
- [12] H. A. M. Alhamzawi, "Faces and eyes Detection in Digital Images Using Cascade Classifiers," *Computer Engineering and Applications Journal*, vol. 7, no. 1, pp. 57–66, 2018, doi: 10.18495/comengapp.v7i1.222.
- [13] H. A. M. Alhamzawi, "Faces and eyes Detection in Digital Images Using Cascade Classifiers," *Computer Engineering and Applications Journal*, vol. 7, no. 1, pp. 57–66, 2018, doi: 10.18495/comengapp.v7i1.222.
- [14] M. S. Fuentes, N. A. L. Zelaya, and J. L. O. Avila, "Coffee Fruit Recognition Using Artificial Vision and neural NETWORKS," in *2020 5th International Conference on Control and Robotics Engineering*, ICCRE 2020, 2020, pp. 224–228. doi: 10.1109/ICCRE49379.2020.9096441.
- [15] P. Constante, O. Chang, E. Pruna, and I. Escobar, "Artificial Vision Techniques for Strawberry 's Industrial Classification," *Ieee Latin America Transactions*, vol. 14, no. 6, pp. 2576–2581, 2016, doi: 10.1158/1078-0432.CCR-12-2618.
- [16] C. H. Setjo, B. Achmad, and Faridah, "Thermal image human detection using Haar-cascade classifier," *Proceedings - 2017 7th International Annual Engineering Seminar*, InAES 2017, no. January 2020, 2017, doi: 10.1109/INAES.2017.8068554.
- [17] R. A. Maulana Budiman, B. Achmad, Faridah, A. Arif, Nopriadi, and L. Zharif, "Localization of white blood cell images using Haar Cascade classifiers," in *Proceedings of 2016 1st International Conference on Biomedical Engineering: Empowering Biomedical Technology for Better Future*, IBIOMED 2016, 2017. doi: 10.1109/IBIOMED.2016.7869822.
- [18] C. Li, Z. Qi, N. Jia, and J. Wu, "Human face detection algorithm via Haar cascade classifier combined with three additional classifiers," in *ICEMI 2017 - Proceedings of IEEE 13th International Conference on Electronic Measurement and Instruments*, Institute of Electrical and Electronics Engineers Inc., Jul. 2017, pp. 483–487. doi: 10.1109/ICEMI.2017.8265863.
- [19] J. J. Soto Espinoza, "Desarrollo de un sistema embebido para detectar en tiempo real la presencia de personas en ambientes controlados," Universidad de las Fuerzas Armadas, 2018.

- [20] D. M. López Mena, “Diseño E Implementación De Un Sistema De Visión Artificial Para Determinar Condiciones De Fatiga En Una Persona Mediante El Índice Perclos, Utilizando Open Cv Y Una Tarjeta Raspberry Pi 3,” Escuela Politécnica Nacional, 2017. [Online]. Available: <https://bibdigital.epn.edu.ec/bitstream/15000/17531/1/CD-8036.pdf>
- [21] A. M. García Piragua and W. A. Moreno Gómez, “Sistema de conteo automático de flujo de personas por medio de visión artificial,” Universidad Distrital Francisco José de Caldas, 2016. doi: 10.1017/CBO9781107415324.004.
- [22] M. G. Jiménez Ochoa, “Desarrollo de un sistema de visión artificial para la detección de aglomeración de personas en un semáforo,” Universidad Nacional de Loja, 2015. [Online]. Available: [http://dspace.unl.edu.ec/jspui/bitstream/123456789/11225/1/Jiménez Ochoa%2C Magaly Gabriela.pdf](http://dspace.unl.edu.ec/jspui/bitstream/123456789/11225/1/Jiménez%20Magaly%20Gabriela.pdf)
- [23] 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 Geoinf*, vol. 5, no. 6, 2016, doi: 10.3390/ijgi5060093.