

Aplicativo para detección y conteo de peatones usando método HOG

Sleyder Arteaga

Ciencias de la computación y electrónica
Universidad Técnica Particular de Loja
Loja, Ecuador
sdarteaga@utpl.edu.ec

Vanessa Narváez Guarnizo

Ciencias de la computación y electrónica
Universidad Técnica Particular de Loja
Loja, Ecuador
vlnarvaez@utpl.edu.ec

Abstract—Desde la evolución de la tecnología se abierto un abanico de posibilidades en las cuales se puede investigar e innovar, el siguiente artículo presenta el desarrollo de un programa que detecte los peatones usando la librería OpenCV mediante el lenguaje de programación Python usando técnicas visión artificial. Primeramente se analizará los distintos métodos para la detección de peatones con el fin de seleccionar el método mas preciso que permita detectar y contar las personas, seguidamente se detalla todos los pasos para la construcción del programa que nos permita detectar y contar peatones logrando así nuestro objetivo y finalmente se hará una análisis de los beneficios o limitaciones del método empleado.

Keywords; *OpenCv; detectores; segmentación; descriptors, técnicas, métodos HOG.*

I. INTRODUCCIÓN

La visión artificial es una parte fundamental en la inteligencia artificial, la detección de humanos y objetos tiene como objetivo abarcar temas de reconocimiento, segmentación y localización. Existe un tema muy interesante dentro de la visión artificial como es el reconocimiento de peatones que permite reconocer las personas que transitan por un lugar específico a través de cámaras de video.

La detección de peatones toma en cuenta varios elementos como el fondo, distancia, enfoque e iluminación que deben ser detectadas en tiempo real y su coste computacional es alto. Para tener una detección exitosa, se debe tener claro dos partes importantes como son, la extracción de la parte donde se encuentre la persona y el algoritmo clasificador que permiten reconocer si es una persona, un animal o un objeto, se puede lograr aplicando etiquetas supervisadas que son entrenados con anticipación.

Finalmente, para la detección de peatones se usa dos características fundamentales que son movimiento y forma, el movimiento permite segmentar la imagen para obtener el cuerpo humano a detectar y la forma permite modelar estructuras frontales, laterales y traseras que permiten detectar al peatón. El siguiente artículo muestra algunos métodos existentes en OpenCV para la detección de peatones y sobre todo se pondrá énfasis en el método HOG que es el que se usa en el desarrollo de la aplicación., además se encontrará un

análisis de cuales son los beneficios y limitación descubiertas en el desarrollo del aplicativo.

II. TRABAJOS RELACIONADOS

La visión artificial viene buscando nuevos caminos, es por ellos que existe algunos trabajos que han ido aportando mejoras de soluciones al uso de detección tanto de humanos como de objetos. A continuación se menciona algunos trabajos que ya han sido hechos y que ayuden a conocer mejor acerca del tema de detección de peatones.

2.1 Desarrollo de un sistema de reconocimiento visual para sistemas Linux embebidos.

El trabajo [1] abarca las diferentes técnicas de detección usando metodologías de Kitchenham, los algoritmos Support Vector Machine (SVM), Adaboost e Histograms of Oriented Gradients (HOG) estos algoritmos permiten la detección de objetos, personas o carros. El autor menciona que usa tres fases importantes como planificación de búsqueda, conducción de búsqueda y representación de resultados, además utiliza metodologías de búsqueda, recopilación, clasificación y análisis de la información.

2.2 Rápida detección de peatones utilizando una cascada de características de covarianza potenciadas

Este artículo muestra el método Haar Cascade para la detección de peatones, donde el autor [2] muestra primeramente un amplio estudio experimental usando campos receptivos locales, histogramas de gradientes orientados y covarianza de la región. Usando el algoritmo Adaboost el mismo que construye una estructura clasificadora en cascada en la fase de detección, además el autor menciona que se a adoptado estrategias de múltiples capas con características heterogenias.

2.1 Un sistema de detección y seguimiento de peatones basado en el procesamiento de video Tecnología

El artículo menciona que existen varios métodos para el conteo y detección de peatones basado en sensores infrarrojo. El autor [4] muestra un experimento que a realizado mediante un programa de visión de computadora en un espacio que permita contar y detectar las personas en la espacios específicos de la universidad, la visión por computador tiene algunos métodos que permiten procesar imágenes a partir de una cámara de video permitiendo adquirir, analizar y entender imágenes.

III. MÉTODOS

En la detección de peatones existen algunos algoritmos que llevan procesos de mejor manera, con el fin de apoyar a la visión artificial y llevar un mejor control en la seguridad de los peatones, a continuación se presenta tres métodos de los mas usados y que se pueden catalogar exitosos por sus características.

3.1 Método 1: Descriptores basados en Histogramas de Gradientes Orientados (HOG)

Los descriptores HOG se basan en la orientación del gradiente en áreas locales de una imagen. La imagen se divide en pequeñas celdas cada una de las cuales acumula direcciones del histograma de gradiente u orientaciones de los bordes de los píxeles de las celdas, como se representa [5] en la Figura 1. Se recomienda para una mejor respuesta normalizar el contraste en unas zonas más grandes (denominadas bloques) y utilizar dicho resultado para normalizar las celdas del bloque. Estos bloques de descriptores normalizados son lo que los autores denominan descriptores HOG, se utilizan los descriptores HOG de la ventana de detección como entrada a un clasificador SVM.

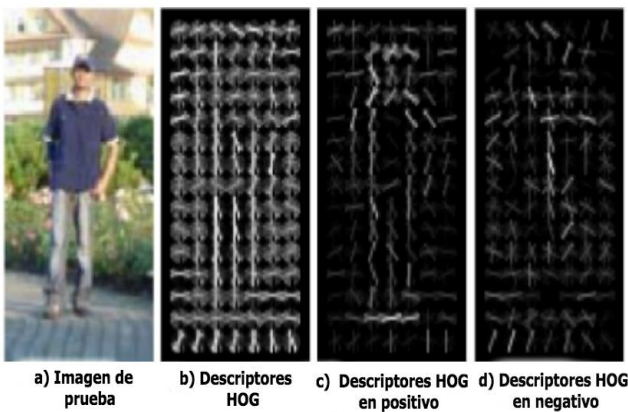


Figura 1. Ejemplo de descriptores HOG en una imagen

Los descriptores HOG nos proporcionan información tal como los cambios de intensidad debido a los contornos o bordes de una imagen. Al tener en cuenta la relación con sus zonas vecinas y colindantes, es posible reconocer cuándo existe una

frontera entre un objeto y otro. De esta manera, permite identificar objetos de siluetas más suaves o más pronunciadas. Es adecuado para la detección de personas, independientemente de su tamaño y sus colores, y fijándonos más en su relación con el entorno, distinguiendo los cambios más pronunciados. En la Figura 2 se puede visualizar el proceso de creación de los descriptores de HOG [5].

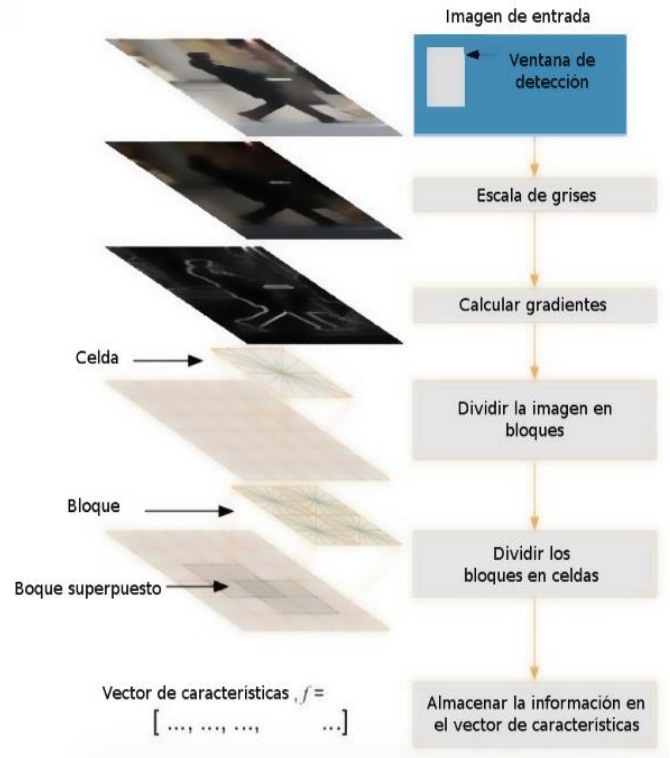


Figura 2. Proceso de descriptores de HOG

3.1.1 Descripción Matemática.

El histograma de una imagen divide el rango de valores posibles de los píxeles de la imagen en una serie de sub-rangos o clases del mismo o distinto tamaño entre ellos. Dado el rango de valores de píxel $[0, 255]$, se realiza una división en ocho clases igual tamaño: $[0, 32)$, $[32, 64)$, $[64, 96)$, ..., $[224, 255]$, y almacena en cada clase la frecuencia de píxeles con un valor comprendido entre ese sub-rango, es decir, el número de píxeles en la imagen cuyo valor está entre los valores de inicio y fin de cada sub-rango.

El Histograma de Gradientes Orientados de una imagen tiene como rango de valores posibles las distintas orientaciones que pueden tomar los gradientes de los píxeles, los distintos grados que pueden tomar sus ángulos de gradiente $[-90^\circ, 90^\circ]$, $[0^\circ, 180^\circ]$, $[0^\circ, 360^\circ]$, ... Este rango se divide en sub-clases del mismo tamaño o distintos para el rango $[0^\circ, 180^\circ]$, dividiendo en nueve sub-rangos: $[0^\circ, 20^\circ)$, $[20^\circ, 40^\circ)$, ..., $[160^\circ, 180^\circ]$, y se almacena en cada uno de ellas la suma de las magnitudes de gradiente de los

píxeles cuyo ángulo de gradiente se encuentra comprendido entre esos valores.

En [5], partiendo del concepto de HOG, se puede obtener más información de una imagen por medio de un descriptor de HOG (del inglés HOG Descriptor). En éste, la imagen se divide en un cierto número de sub-imágenes del mismo tamaño, denominadas celdas, y éstas se agrupan en bloques con un mismo número de celdas de ancho y alto todos ellos. Además, estos bloques se encuentran solapados de forma que el avance de bloques horizontalmente se realiza eliminando la columna de celdas de la izquierda y añadiendo la columna de la derecha y, verticalmente, eliminando la fila de celdas de arriba y añadiendo la fila de celdas de abajo.

De este modo, dada una imagen A de tamaño $W \times H$; un tamaño de celda $C_W \times C_H$ con $W \bmod C_W = 0$ y $H \bmod C_H = 0$; y un tamaño de bloque en celdas $B_W \times B_H$; el ancho y alto de la imagen en celdas, W_C y H_C , y el número de bloques distribuidos horizontalmente y verticalmente, N_{BW} y N_{BH} , se calculan aplicando las formulas propuestas en [5] de la siguiente manera:

$$W_C = \frac{W}{C_W}, \quad H_C = \frac{H}{C_H}$$

Donde:

W_C : Es el ancho de la imagen en celdas
 H_C : Es el alto de la imagen en celdas
 W, H : Son el tamaño de la imagen
 C_W, C_H : Son el tamaño de la celda de la imagen

$$N_{SW} = 1 + W_C - B_C, \quad N_{BH} = 1 + H_C - B_H$$

Donde:

N_{SW} : Es el número de solapamiento del tamaño de los bloques
 N_{BW} : Es el número de bloques verticales
 N_{BH} : Es el número de bloques horizontales

Y por tanto, el número total de celdas N_C y el número total de bloques N_B resultantes de la imagen A será igual a:

$$N_C = W_C * H_C \quad \square 3 \square$$

$$N_B = N_{BW} * N_{BH} \quad \square 4 \square$$

Y la distribución de celdas (C) y bloques (B) es la siguiente:

$$A = \begin{pmatrix} a_{00} & a_{10} & \dots & a_{(w-1)0} \\ a_{01} & a_{11} & \dots & a_{(w-1)1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{0(H-1)} & a_{1(H-1)} & \dots & a_{(w-1)(H-1)} \end{pmatrix} \quad \square 5 \square$$

$$C = \begin{pmatrix} C_{00} & C_{10} & \dots & C_{(w-1)0} \\ C_{01} & C_{11} & \dots & C_{(w-1)1} \\ \vdots & \vdots & \ddots & \vdots \\ C_{0(H_C-1)} & C_{1(H_C-1)} & \dots & C_{(W_C-1)(H_C-1)} \end{pmatrix} \quad \square 6 \square$$

$$B = \begin{pmatrix} b_{00} & b_{10} & \dots & b_{(N_{BW}-1)0} \\ b_{01} & b_{11} & \dots & b_{(N_{BW}-1)1} \\ \vdots & \vdots & \ddots & \vdots \\ b_{0(N_{BH}-1)} & b_{1(N_{BH}-1)} & \dots & b_{(N_{BW}-1)(N_{BH}-1)} \end{pmatrix} \quad \square 7 \square$$

Donde:

N_B : número de bloques.
 W_C y H_C : el ancho y alto de la imagen en celdas.
 A, B, C : Son las distribuciones en bloques de la imagen, H representa la altura y W el ancho de las celdas que conforman los bloques.

A partir de esta estructuración de la imagen, el descriptor de HOG calcula de forma independiente el HOG de cada celda y cada bloque, agrupando los descriptores HOGs de sus celdas correspondientes para la obtención de sus peatones. Entonces, el número de HOGs que contiene un descriptor será:

$$N_{HOG} = B_W * B_H * N_B \quad \square 8 \square$$

Y si se divide el HOG en n clases, dado que cada bloque contiene $B_W * B_H$ descriptores HOG, entonces el número total de valores N_V que se tomará de la imagen A será:

$$N_V = n * N_{HOG} \quad \square 9 \square$$

Para el modelo de detección, sobre la colección de imágenes se debe calcular el descriptor de HOG de cada imagen, etiquetando cada descriptor como positivo si es una imagen de persona “+1” o negativo si no lo es “-1”. El cálculo de los descriptores de HOG puede tener un coste en tiempo de computación bastante grande, puesto que requiere el cálculo de un HOG por cada celda. Para agilizar esto se puede usar una técnica denominada “Integral de HOG” que ayuda a mejorar el coste de computacional a cambio de aumentar el coste en memoria. Para calcular la Integral de HOG se necesitan primero n matrices auxiliares de tamaño igual a la imagen de entrada A de tamaño $W \times H$ y donde n es el número de clases en el HOG. Cada una de estas matrices estará asociada a una única clase, es decir, a un único rango de ángulos de gradiente, y donde cada uno de sus píxeles cumple:

$$M(b, i, j) \begin{cases} |G(i, j)|, & \theta(i, j) \in R(b) \\ 0, & \text{eoc} \end{cases} \quad \square 10 \square$$

Donde $b \in \{0, 1, \dots, n-1\}$ indica la clase $i \in \{0, 1, \dots, W-1\}$ indica la columna de la matriz, $j \in \{0, 1, \dots, H-1\}$ indica la fila de la matriz y $R(b)$ es el rango de valores de ángulo asociado a la clase b .

3.2 Método 2: Haar Cascade

En [6], Haar Cascade es un clasificador entrenado para la detección de personas, el algoritmo Haar Cascade es un clasificador propio de OpenCV, consta de ficheros con características y partes del cuerpo humano. Este clasificador esta formado por archivos xml integrados por patrones de las partes de los seres humanos con el fin de ser consumidos.

La idea principal es que la primera capa sea entrenada desde los pequeños pero eficientes clasificadores promovidos, los cuales pueden rápidamente rechazar a la mayoría de imágenes por no tener los rasgos positivos de interés. Después, con el resto de las imágenes que lograron pasar por estos primeros clasificadores se continua a las siguientes rondas de clasificación dentro de la cascada que son cada vez más exigentes hasta declarar que realmente existe en la imagen el objeto que se está buscando.

La cascada final es capaz entonces de rechazar a la mayoría de las imágenes/subregiones con pocas operaciones.

- Evaluar los rasgos de las regiones.
- Ejecutar los clasificadores débiles por cada rasgo.
- Convertir a los clasificadores débiles en uno más fuerte.

El proceso de cascada ha ordenado al final una estructura de árbol de descarte. Cada capa tiene la capacidad de descartar por completo a una región por no contener los rasgos de interés, logrando que sea más rápido el algoritmo en general. Como se representa en la Figura 3.

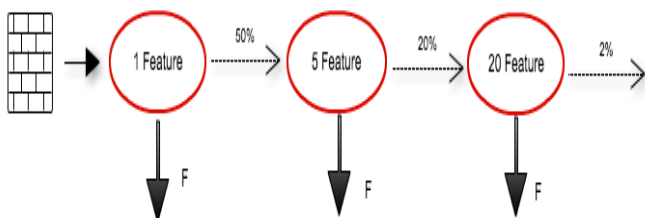


Figura 3. Clasificador, para lograr una mejor clasificación se realiza varias detecciones en cada paso va eliminando falsos positivos es por ello que podemos observar en la imagen se realiza tres veces el clasificar para obtener el 100% de la detección.

3.2.1 Algoritmo

El uso de Haar Cascade permite seguir una serie de pasos que nos ayudan a la utilización de los archivos xml propios del Haar Cascade. El autor [6] plantea los siguientes pasos para la obtención de peatones mediante la comparación de imágenes y los archivos xml, a continuación se puede observar el proceso y un diagrama propuesto en la Figura 4.

- Leer Imagen: Consiste en obtener la imagen mediante una cámara de video.
- Cargar Haar Cascade: Utilizar el clasificador Haar Cascade que se basa en los cambios de intensidad de la imagen para detectar un objeto.
- Comparar imagen con Haar Cascade: Compará los objetos encontrados en la imagen con los que se tiene en el clasificador.
- Detección de personas: Detectar la persona que coincide con las dimensiones descritas en el clasificador.
- Dibujar rectángulo: Graficar un rectángulo de la persona encontrada.
- Crear una nueva imagen: Se debe crear una copia de la imagen original para poder representar los objetos.

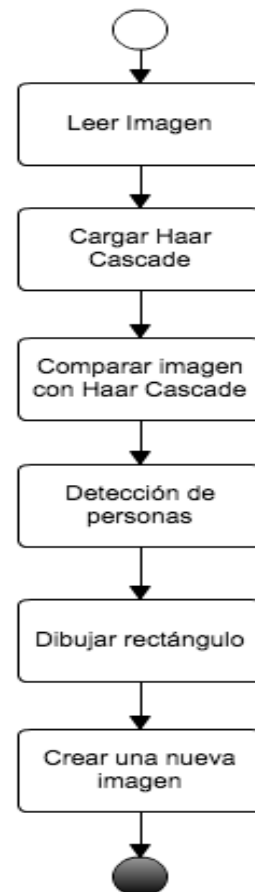


Figura 4. Proceso para usar Haar Cascade en OpenCV

3.2.2 Matriz de confusión

La matriz de confusión es de 2x2, evalúa dos instancias respecto a los resultados obtenidos, además estos resultados se convierten en instancias que deben ser clasificadas. Esta matriz trabaja con la clase persona y no persona, donde reconoce y clasifica los resultados, así lo plantea [7].

TABLA I
Matriz de confusión

Verdadero Positivo (VP)	Verdadero Negativo (VN)
# objetos detectados	# objetos detectados
Falso Positivo (FP)	Falso Negativo (FN)
# objetos detectados	# objetos detectados

- *Verdadero Positivo*: objeto detectado (p) y coincide con el objetivo (v).
- *Verdadero Negativo*: objeto no detectado (p) y no hay objetivo (v).
- *Falso Positivo*: objeto detectado (p) pero no es el objetivo (f).
- *Falso Negativo*: objeto no detectado (n) pero hay objetivo en la imagen (f).

3.2.3 Operaciones de medición de resultado

Las fórmulas para medir los resultados obtenidos en la matriz de confusión [7], permiten conocer que tan preciso puede ser el método Haar Cascada basándose en los resultados obtenidos en las pruebas. A continuación se presenta las fórmulas:

$$\text{precisión} = \frac{vp}{vp+fp} \quad \square\square1\square$$

$$\text{sensibilidad} = \frac{vp}{vp+fn} \quad \square\square12\square$$

3.3 Sistemas basados en detectores de rasgos PCA

Este tipo de algoritmos esta basado en la búsqueda de rasgos humanos, funciona seleccionando segmentos donde se identifique la presencia de rasgos humanos con el fin de extraerlos e identificarlos. Este tipo de algoritmos localiza en un segmento, donde se pueden añadir varianzas como la rotación o la iluminación.

Se recoge mas información mediante mapas de simetría de los bordes horizontales así como del numero de dichos bordes por columna. Los máximos en dicho mapa contribuyen a la identificación de las regiones de interés, que son encerradas para ser evaluadas, en la fase de reconocimiento posterior como lo muestra el autor [8]. La figura 5 presenta como se detecta los contornos tomando las simetrías verticales, en [9] lo detalla en la Figura 5.

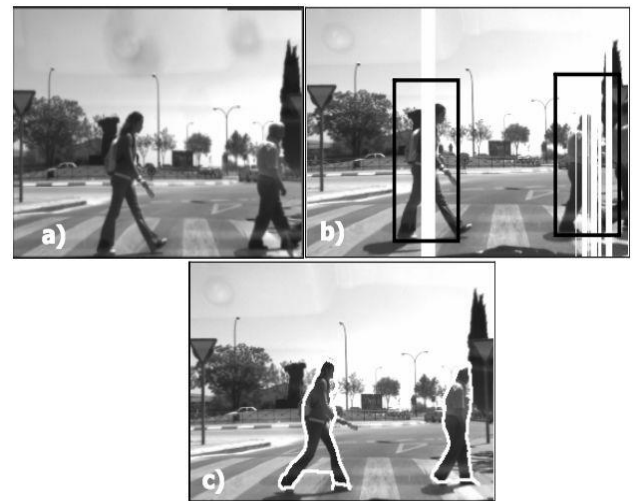


Figura 5: Extracción de bordes en el espectro visible: (a) Imagen original; (b) Simetrías verticales; (c) Detección de contorno deformable aparece en blanco.

3.3.1 Sistemas basados en descriptores

Un descriptor representa una región de la imagen, previamente extraída. Cada región seleccionada se caracteriza por un vector descriptor o vector de rasgos. Para ello, dichas regiones son transformadas al espacio de rasgos mas adecuado, donde la información se representa de una forma mas compacta y resulta más sencillo definir los correspondientes descriptores o vectores de rasgos.

3.3.2 Etapa de detección de objetos de interés

En [8] se ha desarrollado un sistema de detección de peatones mediante el uso de sensores infrarrojos que ayuda al funcionamiento integral de múltiples estrategias. Como podemos observar en la Figura 6.

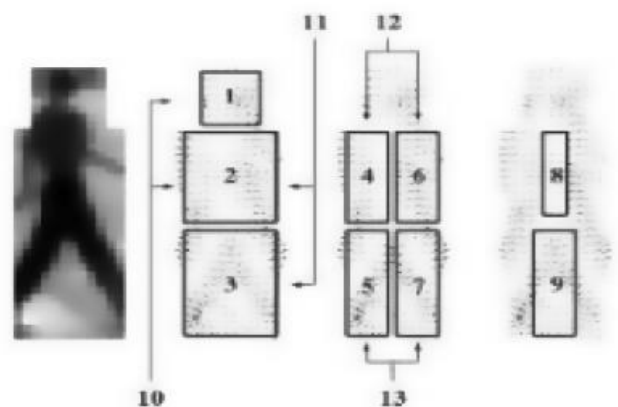


Figura 6: Se muestra la configuración de las nueve subregiones sobre la imagen del gradiente. La distribución de los vectores es evaluada en cada subregión. Además, se construyen cuatro pares de combinaciones (regiones 10-13).

3.3.3 Etapa de Reconocimiento de Personas

El reconocimiento de objetos es uno de los retos fundamentales de la visión por computador. Las recientes técnicas de visión y de aprendizaje de máquina (Machine Learning) han aportado avances significativos. Desde un punto de vista global, se pueden catalogar los sistemas de detección de objetos, así lo propone [8].

1. Sistemas basados en modelos, en los que se define un patrón para el objeto de interés y el sistema trata de hacer corresponder ese patrón con la imagen.
2. Métodos basados en invariantes de la imagen, que tratan de establecer la correspondencia a través de las relaciones existentes en patrones de la imagen, como los niveles de intensidad buscan solo los objetos de interés. También se denominan sistemas libres de modelos ("Model-free").
3. Sistemas de detección de objetos, se caracterizan por sus algoritmos de aprendizaje basados en ejemplos. Estos sistemas, aprenden los rasgos característicos de una clase a través de un conjunto de ejemplos etiquetados como positivos y negativos.

3.3.4 Patrones basados en el movimiento

Este esquema propuesto por [8], se basa en movimiento cíclico de las personas, pero presenta muchas limitaciones en la detección de peatones:

1. El reconocimiento se lleva a cabo en base a una secuencia de imágenes. En función del número de imágenes necesarios, la identificación del movimiento puede venir retardada en el tiempo, en el caso de necesitar varias imágenes, el tiempo de procesamiento se incrementa o la identificación puede ser inmediata.
2. No detecta a individuos que estén quietos o cuyos movimientos sean extra notables como correr, saltar, girar o vagar. Generalmente se limitan a gente caminando a velocidad constante.
3. Los peatones son objetos no-rígidos y cada parte del cuerpo tiene un movimiento diferente. Una posible alternativa para interpretar esos movimientos complicados, es representar conocimiento sobre la forma además del movimiento.

Podemos observarlos a continuación en la Figura 7 como lo presenta el autor [9].



Figura 7: Ejemplo del movimiento cíclico de un humano.

IV. APLICACIÓN

4.1 Diseño

Para la detección y conteo de peatones la aplicación está creada en OpenCV y el lenguaje de programación python. A continuación se detalla el proceso para la detección y conteo de peatones.

El algoritmo usado en el desarrollo de este proyecto es HOG (Histogramas de Gradientes Orientados). Cuando el método detecta los peatones se presentan un rectángulo de color verde que marca el número de personas encontradas en el espacio. La Figura 8 muestra la detección mediante el algoritmo HOG.



Figura 8. Detección de peatones mediante método HOG

La aplicación además permite contar las personas que transitan por un punto específico, como en este caso se cuenta el número de peatones que pasan por las líneas verticales. Por eso existe un punto central en el peatón detectado, cuando este punto se localiza en el centro de las líneas verticales el contar de personas aumenta. Así se puede controlar la magnitud de peatones que pasan por el lugar en específico donde se ubica la cámara. Como se puede observar en la figura 9.

La aplicación no reconoce cuando el peatón se dirige a la izquierda o derecha, si el peatón pasa una vez y retrocede o vuelve a pasar de lado contrario lo contará como dos peatones diferentes.

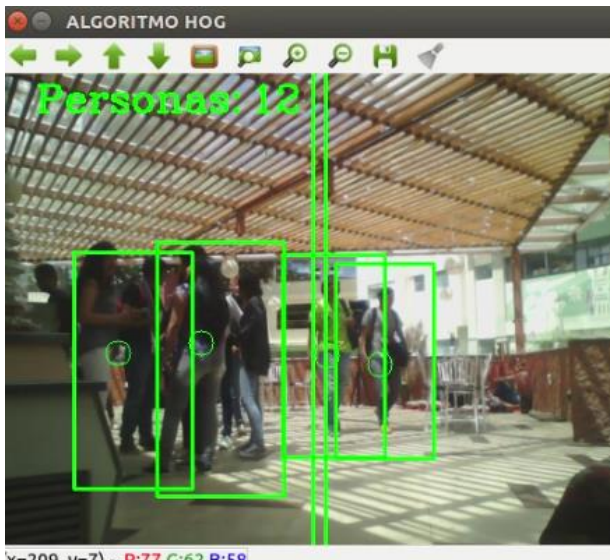


Figura 9. Conteo de peatones mediante el establecimiento de puntos centrales en el peatón y las líneas verticales

Para que la aplicación tenga éxito se debe controlar la distancia y la luz, para lograr una detección buena debe colocarse a mas o menos tres metros de distancia del lugar al que se desea controlar. La figura 10 es un ejemplo claro al detectar peatones según distancia, como podemos observar la persona esta muy cerca de la cámara es por ello que no logra detectarla.

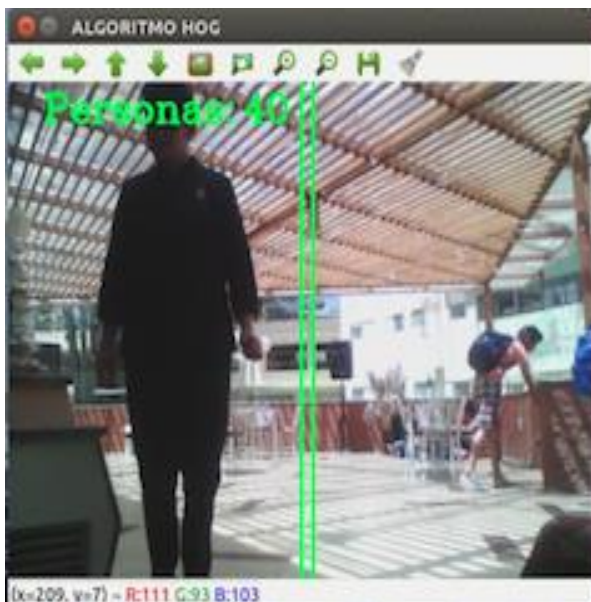


Figura 10. Detección de peatones a partir de tres metros.

Por otra parte, la luz debe ser natural y suave, este factor es uno de los mayores problemas al momento de detectar ya que cuando es muy intensa el método se concentra en ese punto tratando de identificar si es un peatón. La figura 11 muestra como el método HOG idéntica un espacio que esta muy iluminado.

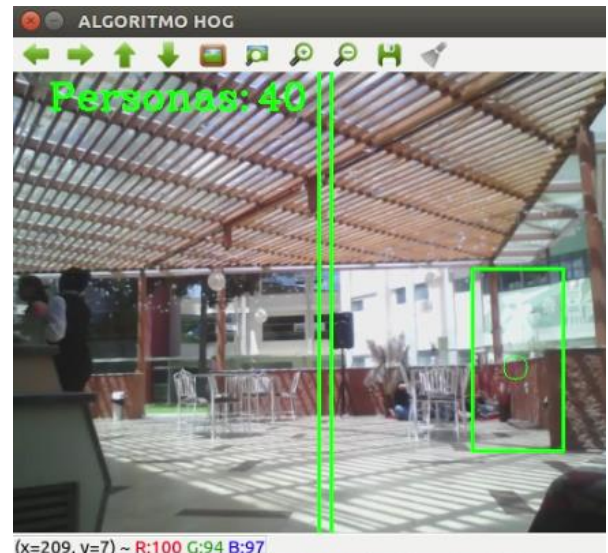


Figura 11. Detección de espacio de luz

4.2 Pruebas

Para analizar la precisión de la aplicación se ejecutaron algunas pruebas. A continuación se detalla el número de detecciones correctas e incorrectas. Las variables de verdadero positivo y falso positivo se las a tomado de la tabla de matriz de confusión propuesta por HaarCascade. Además se han añadido otras variables como distancia y luz.

Verdadero Positivo (VP): abarca a todos los peatones que fueron detectados sin ningún problema y son contados al pasar la línea central.

Falso positivo (FP): Son aquellos peatones que tuvieron problemas al ser detectar, estos problemas se pueden dar por distancia o por luz.

Distancia (D): peatones que no se detectaron porque se encontraban a menos de tres metros

Luz (L): detecciones no válidas ya que la aplicación confunde luz intensa con un peatón.

4.2.1. Pruebas unitarias

Para probar el funcionamiento de la aplicación de detección de peatones, se realizó pruebas individuales a dos personas en la que se pudo observar las fallas de distancia y de luz que son las debilidades del algoritmo HOG. A continuación se presenta los resultados de las pruebas realizadas.

En la tabla II y III se presentan los resultados de precisión en el reconocimiento de peatones, es decir, si lo reconoce o no lo hace, también especifica los factores que ocasionaron el problema.

TABLA II
Pruebas de Detección
Persona 1

Número Pruebas	Detección de peatón	Factores	
		Distancia a partir de 3m.	Luz
1	si	----	----
2	si	----	----
3	si	----	----
4	no	si	----
5	si	si	----
6	si	----	----
7	no	si	----
8	si	----	----
9	si	----	----
10	si	----	----
11	no	si	----
12	no	si	----
13	si	----	----
14	si	----	----
15	no	si	----

TABLA III
Pruebas de Detección
Persona 2

Número Pruebas	Detección de peatón	Factores	
		Distancia a partir de 3m.	Luz
1	no	si	----
2	si	----	----
3	no	si	----
4	si	----	----
5	si	----	----
6	si	----	----
7	no	si	----
8	si	----	----
9	si	----	----
10	no	si	----
11	no	si	----
12	si	----	----
13	si	----	----
14	si	----	----
15	no	si	----

Las tablas II Y III marcan un margen de error del 40% debido a problemas de distancia, estos dos factores son sensibles en el algoritmo de HOG.

4.2.2. Prueba final

Después de haber hecho pruebas unitarias y verificado que la aplicación funciona tomando en cuenta que consta con un margen de error del 40%, se puso en funcionamiento en un lugar abierto como se especificó en el apartado 4.1 de este documento. A continuación se detalla los hallazgos de la aplicación.

TABLA IV
Matriz de pruebas en un lugar abierto usando la matriz de confusión de Haar Cascade

Verdadero Positivo (VP)	40
Falso Positivo (FP)	5
• Distancia (D)	3
• Luz (L)	2

La tabla IV muestra la prueba realizada y se estima alrededor de 45 personas de las cuales el 5% de los peatones no han sido reconocidos con precisión, por problemas de distancia y luminosidad.

CONCLUSIONES

Después de haber probado y experimentado el método HOG en OpenCV y python. Se ha concluido que, este método permite la detección de peatones con un mejor desempeño. Se sugiere usar este algoritmo en lugares donde no haya demasiada luz artificial para tener una buena detección la cámara de video debe estar colocada a una distancia de aproximadamente tres metros.

La aplicación presenta un margen de error del 40% según las pruebas realizadas y demostradas en la Tabla IV, el método sigue los movimientos de las personas hasta obtener su silueta para detectar si se trata de un peatón, además el video no muestra retrasos al momento de la ejecución.

El conteo de personas es muy útil para tener conocimiento de la afluencia de personas que existen en el lugar, es por ellos que para lograr contar los peatones se deben encontrar los puntos del rectángulo y del centro de las líneas en el momento que el peatón cruce de lado a lado.

REFERENCES

- [1] S, E. L. A. S. L. S. Y., De, P. (2006). Escuela politécnica nacional. Implementación de algoritmos de detección de persona en imagen digital.
- [2] Paisitkriangkrai, S., Shen, C., & Zhang, J. (2008). Fast pedestrian detection using a cascade of boosted covariance features. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8), 1140–1151. <http://doi.org/10.1109/TCSVT.2008.928213>
- [3] Kalal, Z., Matas, J., & Mikolajczyk, K. (2008). Weighted Sampling for Large-Scale Boosting. *Methods*, 1(16), 42.1-42.10. <http://doi.org/10.5244/C.22.42>
- [4] Cortés, D. (2013). Implementación de un algoritmo para el reconocimiento y análisis de peatones utilizando visión por computador, (c).
- [5] J. Intxaurre Txarterina, *Detección de personas*, 1st ed. País Vasco: Unidad del País Vasco, 2013, pp. 31-34.
- [6] Barba, L. (2015). *Universidad Politécnica De Madrid*. Utilización de Métodos de Visión Artificial Para PC Como Apoyo En La Automoción, 52.
- [7] Miocénicas, S. (2006). Escuela politécnica nacional, 1–126. Estudio e implementación de algoritmos de detección de silueta de una persona en imagen digital proyecto
- [8] C. Gomez, "Detección de Peatones en el Espectro Visible e Infrarrojo para un Sistema Avanzado de Asistencia a la Conducción", I, UNIVERSIDAD CARLOS III DE MADRID, 2008.
- [9] Hilario, C., & Collado, J. M. (n.d.). BASADA EN MODELOS DE CONTORNOS ACTIVOS Y VISIÓN ESTÉREO.