

# Detección de caras utilizando el descriptor HOG multiescala

Rubén Darío Bohórquez Cortázar  
Universidad de los Andes  
Bogota, Colombia

rd.bohorquez10@uniandes.edu.co

Javier Felipe Cifuentes  
Universidad de los Andes  
Bogota, Colombia

jf.cifuentes10@uniandes.edu.co

## Abstract

*En este laboratorio se exploró la estrategia para la detección de objetos HOG multiescala y su aplicación como aproximación para solucionar el problema de detección de caras en un subconjunto de la base de datos wider face. Para realizar esto, se utilizó como base la biblioteca de funciones VLFeat, desarrollada por Andrea Vedaldi y Brian Fulkerson. Partiendo de esto, se utilizó como metodología de evaluación el cálculo del valor  $F$  máximo en una curva de precisión y cobertura. Se obtuvo un valor de  $F$  máximo de  $1,13E-05$  para nivel fácil,  $6,09E-06$  para nivel medio y  $2,54E-06$  para nivel difícil. Partiendo de estos resultados, se pudo concluir que el modelo del descriptor HOG multiescala estudiado no es el óptimo debido a que el buen funcionamiento de este descriptor esta estrechamente ligado con un entrenamiento fuerte del clasificador; no obstante, como en este estudio se utilizaron pocas imágenes para entrenamiento, el clasificador implementado no fue el mejor, lo que explica la gran cantidad de falsos negativos y falsos positivos obtenidos.*

## 1. Introducción

La detección de elementos u objetos dentro de las imágenes es una de las áreas de la visión por computador más estudiadas en la actualidad debido a la gran diversidad de aplicaciones que tienen los algoritmos de detección en múltiples áreas de conocimiento tanto en el ámbito académico como en la industria. Entre las distintas aplicaciones se pueden resaltar, la detección de peatones en la industria automotriz, la detección de tumores en imágenes de resonancia magnética, detección de armas de fuego en los equipajes, detección de rostros en la industria del desarrollo de cámaras y dispositivos móviles, entre otros [2][3]. De forma general, la detección de objetos se basa en el reconocimiento de un objeto específico o de cierto grupo de objetos dentro de una imagen, exponiendo no solo la presencia del objeto sino la ubicación del mismo dentro de la imagen [2]. Debido a la importancia de los problemas de

detección en el ámbito de la visión, se han desarrollado múltiples algoritmos para la solución de estos problemas. Entre estos algoritmos se encuentra la estrategia basada en extracción de características HOG (Histograma de gradientes orientados) y sus múltiples variaciones tales como el descriptor HOG multiescala, HOG utilizando espacio selectivo, HOG basado en la selección de bloques, entre otros [3]. Debido a la importancia de la estrategia HOG en la historia del estudio de los problemas de detección en el ámbito de la visión por computador, se establece como el objetivo principal de este artículo el estudiar con detalle la estrategia HOG multiescala para el desarrollo de un modelo adecuado para la detección de caras en imágenes de distintas categorías. Para esto, se va a hacer uso de la biblioteca de funciones VLFeat, desarrollada por Andrea Vedaldi y Brian Fulkerson.

## 2. Materiales y métodos

### 2.1. Wider Face

Wider face es una base de datos desarrollada para el estudio del problema de detección de caras. Cuenta con 32,203 imágenes de distintos escenarios y 393,703 etiquetas de caras presentes en esas imágenes citewf. Esta base de datos es especialmente retardadora debido a que las caras en las distintas imágenes presentan un alto grado de variabilidad en la escala, la postura y la oclusión. Wider face está organizada en 61 clases de escenarios, entre los que se encuentran traffic, funeral, ceremony y car Racing citewf. En la figura 1 se puede observar un ejemplo de las imágenes que contiene esta base de datos y ejemplos de las distintas etiquetas de caras [1].

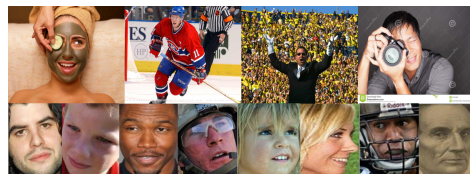


Figure 1. Imágenes base de datos Wider Face

## 2.2. HOG multiscala

La estrategia HOG, histogramas de gradientes orientados, fue descrita inicialmente por Dalal y Triggs en el contexto de la detección de peatones en imágenes [4] en vídeo [5]. Ahora bien, a partir de esta aproximación inicial, se han considerado distintas extensiones a escala múltiple del descriptor HOG. Por ejemplo, Ning He y su grupo de investigación estudiaron la detección de seres humanos y demostraron que, al combinar descriptores HOG en múltiples escalas en una única codificación, el rendimiento del algoritmo para este contexto, mejoraba considerablemente [6]. De igual forma, Felzenszwalb y su grupo de investigación, utilizaron pirámides HOG, donde los descriptores HOG se combinan en dos escalas, mostrando también un aumento en el desempeño de este algoritmo para la detección de peatones [2]. Como estos, existen varios ejemplos que demuestran de forma contundente que las extensiones del algoritmo HOG presentan un mejor desempeño en comparación con la estrategia HOG de una sola escala. Ahora bien, el algoritmo HOG multiscala parte del proceso clásico para el cálculo de HOG, en el cual se realizan tres etapas fundamentales: el cálculo de los gradientes de la imagen, la construcción de los histogramas ponderados por cada celda, y la construcción de histogramas normalizados por bloques, estas se pueden observar con más detalle en la figura 2. No obstante, para adicionar el factor multiscala, se realiza este mismo proceso, pero realizando la convolución con el descriptor HOG para distintos niveles de la imagen, es decir, realizando una pirámide de la imagen y determinando la respuesta del descriptor para cada nivel [3].

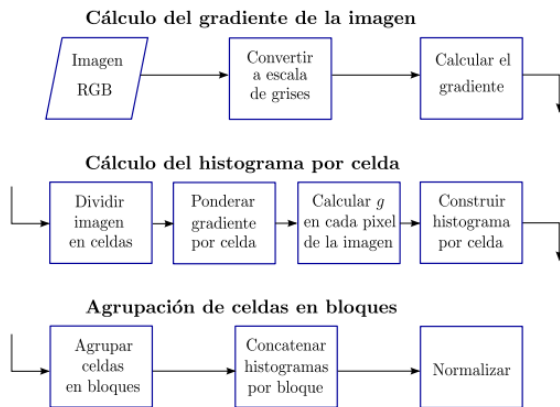


Figure 2. Etapas de la extracción de HOG [3]

### 2.2.1 Parámetros

El método de HOG multiscalar, tiene varios hiperparámetros, generalmente definidos según el problema que se esté enfrentando. Dentro de estos parámetros que se encuen-

tran, el tamaño de la celda, el cual determina la cantidad de pixeles los cuales compondrán cada una de las celdas, es decir, si el tamaño de celda es de 8, la imagen será dividida en celdas de 8x8. Adicionalmente se encuentra la variable de escalas. Adicionalmente, se pueden resaltar parámetros asociados a la formación de bloques, tales como el tamaño del bloque que está asociado al número de celdas contenidas en un bloque, el desplazamiento atinente al número de celdas superpuestas por bloque y la norma, que expone el esquema de normalización en el bloque. Finalmente, se pueden identificar parámetros asociados al histograma, entre estos se resalta el número de bins que para el caso específico del HOG multiscala estudiado, corresponde a 31 // Dentro de estos parámetros existen algunos no asociados directamente a HOG. Están relacionados con la forma en que varían las escalas, así como el entrenamiento y la detección. Dentro del escalamiento, el hiperparámetro más importante es la escala. Esta determina la medida en la cual la imagen será modificada, para cambiar su tamaño, es decir, modificara la escala resultante de la imagen, esto con el propósito de poder utilizar HOG a diferentes escalas y así determinar detecciones a diferentes tamaños de objeto. Dentro de la sección de detección, existen varios parámetros. Por ejemplo, parámetros propios del método de convolución. También existen parámetros de umbral, los cuales determinan en qué momento se considera una detección como verdadera, todo dependiendo del umbral que se tome como parámetro. Aquellos que no cumplan con el umbral, serán eliminados. Esto se conoce como supresión de no máximos. Finalmente, ahí hiperparámetros asociados al entrenamiento, por ejemplo, todas aquellas, variables asociadas al entrenamiento del SVM, como, por ejemplo, los positivos, los negativos, C, épsilon, etc. bins [7].

## 3. Resultados

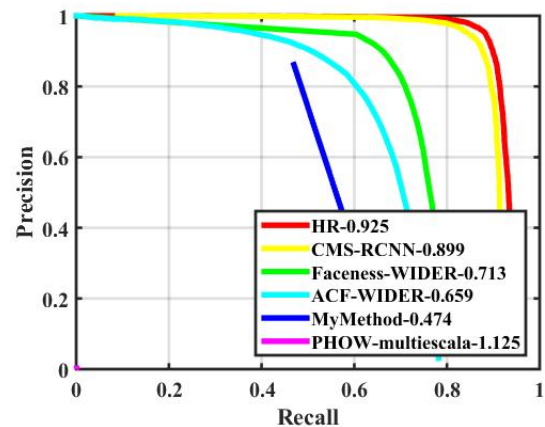


Figure 3. Curva precisión y cobertura: Nivel Fácil

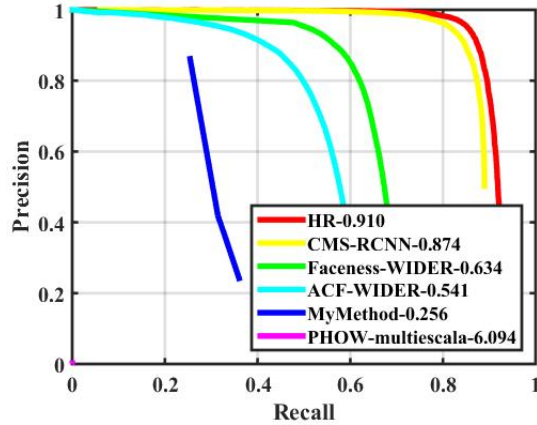


Figure 4. Curva precisión y cobertura: Nivel Medio

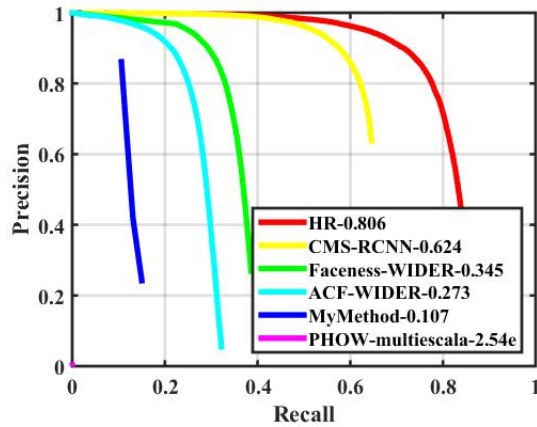


Figure 5. Curva precisión y cobertura: Nivel Difícil

Curva precisión y cobertura	Valor
Nivel Fácil	1,13E-05
Nivel Medio	6,09E-06
Nivel Difícil	2,54E-06

Table 1. Valores curvas precisión y cobertura para cada nivel

## 4. Análisis de resultados

Como se puede ver en los resultados, todos los valores resultantes son bastante bajos, implicando que los resultados no dieron valores significativamente superiores a 0. Se atribuyen estos resultados al hecho de que, el modelo no fue entrenado de la mejor manera posible. Ya que se utilizó un subconjunto pequeño de imágenes de training, así como imágenes de negativos, el proceso de entrenamiento de support vector machine, no fue optimo, generan un hiperplano que no separa o discrimina de manera eficiente entre las imágenes. Adicionalmente, dado el bajo número de iteraciones de hard negative mining, los negativos usados para entrenar el modelo, no son lo suficientemente parecidos a

los positivos, para que el modelo entrenado distinga eficazmente entre lo que es una cara y lo que no lo es. Finalmente, los hiperparámetros de HOG multiescala, deben ser escogidos cuidadosamente en la fase de experimentación, ya que este modelo depende en gran medida de estos. Se requiere de una experimentación detallada en el efecto de cada uno de estos parámetros en el resultado.

### 4.1. Patrones

#### 4.1.1 Falsos positivos:

La mayoría de los falsos positivos presentan forma circular, es decir, el modelo asocia la forma circular al rostro, por lo cual, objetos esféricos y circulares fueron detectados como rostros. Esto se puede evidenciar en la figura 6. Esto es un patrón que se repite a lo largo de la base de datos. Adicionalmente se pudo observar que muchos falsos positivos estaban asociados al calzado, como se muestra en la figura 7.

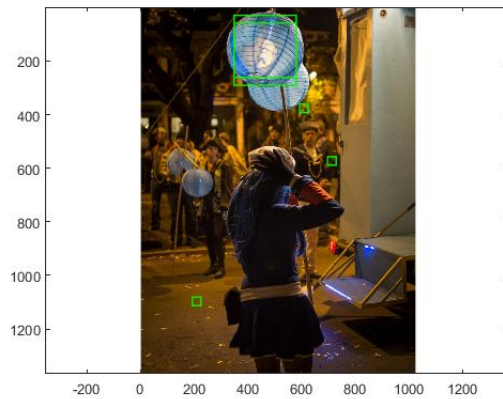


Figure 6. Falso positivo por parecido con formas circulares

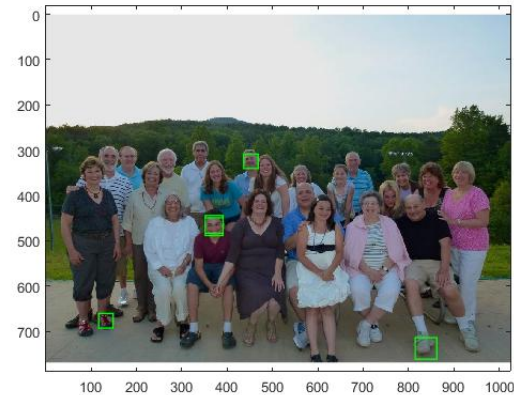


Figure 7. Falso positivo por calzado

#### 4.1.2 Falsos negativos

Se noto que en las imágenes con rostros demasiado grandes, el detector, reconocia partes circulares individuales, pero no el rostro completo, como se puede evidenciar en la figura 8. Se asocia esto a las escalas utilizadas. Adicionalmente, otro patrón que se identifico son las caras de perfil como se muestra en la figura 9, esto se asocia al entrenamiento realizado.

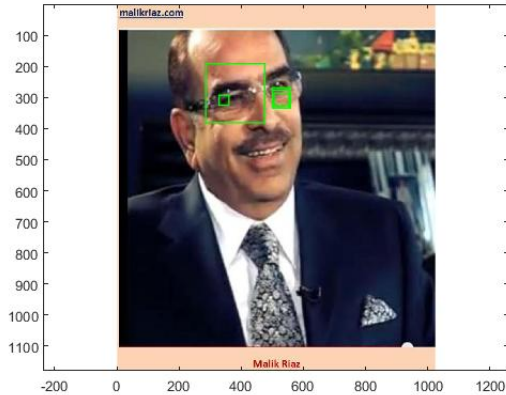


Figure 8. Falso negativo por rostro demasiado grande

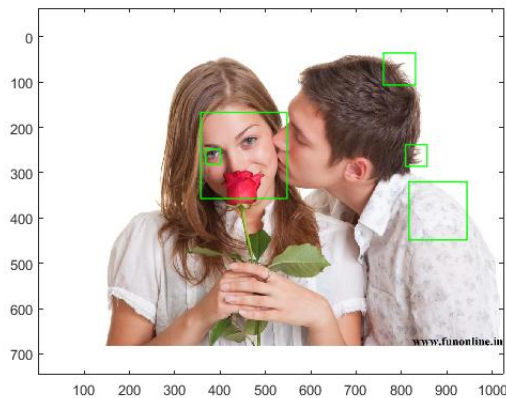


Figure 9. Falso negativo por rostro de perfil

## 5. Discusión

Para la evaluación de un problema general de detección se realiza una curva de precisión y cobertura. Las curvas de precisión y cobertura exponen la relación entre los parámetros precisión y cobertura para la detección de los distintos objetos en el total de las imágenes. La precisión y la cobertura se definen como:  $P = \frac{TP}{TP+FP}$   $R = \frac{TP}{TP+FN}$  TP: verdaderos positivos, FP: falsos positivos, FN: falsos negativos, P=precisión, R=Recall.

La curva indica que tan buenos son los resultados de detección obtenidos para el algoritmo de detección implementado, en comparación con las anotaciones de detección utilizadas. Para evaluar los distintos métodos, se debe calcular el área bajo la curva o reportar la F medida máxima. La F medida máxima se define como:  $FMM = \frac{2PR}{P+R}$  Los resultados ideales en una curva de precisión y cobertura, es la obtención de un valor de F medida máxima igual a 1. El cual indicaría una precisión del cien por ciento y una cobertura del 100 por ciento, es decir, esto implicaría que el algoritmo logró detectar en su totalidad los elementos de interés, sin generar falsos positivos ni falsos negativos.

## 6. Conclusiones

- Se puede afirmar que el modelo del descriptor HOG multiescala estudiado no es el óptimo debido a que el buen funcionamiento de este descriptor esta estrechamente ligado con un entrenamiento fuerte del clasificador, no obstante, como en este estudio se utilizaron pocas imágenes para entrenamiento, el clasificador implementado no fue el mejor, lo que explica la gran cantidad de falsos negativos y falsos positivos obtenidos.
- El descriptor HOG multiescala es una variante del algoritmo de detección clásico HOG, el cual permite realizar un análisis multinivel de las imágenes facilitando el desarrollo y entrenamiento del clasificador. Ahora bien, es imperativo realizar una correcta selección de los hiperparámetros para asegurar el buen desempeño del algoritmo.
- Se debe resaltar que como era de esperar, en las imágenes que presentaban mayores problemas asociados a la oclusión, la escala y la postura no se logró realizar una correcta detección. Esto está relacionado también con la baja cantidad de imágenes utilizadas en el entrenamiento.
- Finalmente, se debe resaltar que después de realizar una investigación detallada del algoritmo de violaciones y comparar su desempeño con el algoritmo HOG multiescala se puede afirmar que el primero es más eficiente, más rápido y arroja mejores resultado que el segundo. Exponiéndose como una herramienta más útil para la solución del problema de detección de caras.

## References

- [1] WIDER FACE: A Face Detection Benchmark. (n.d.). Retrieved April 27, 2017, from <http://mmlab.ie.cuhk.edu.hk/projects/WIDERFace/>
- [2] Newell, A. J., Griffin, L. D. (2011, September). Multi-scale histogram of oriented gradient descriptors for robust character recognition. In Document Analysis and



Recognition (ICDAR), 2011 International Conference on (pp. 1085-1089). IEEE.

- [3] Osorio Arroyave, E. (2015). Detección automatizada de objetos en secuencias de video utilizando histogramas de gradientes orientados (Bachelor's thesis, Pereira: Universidad Tecnológica de Pereira).
- [4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, volume 1, pages 886–893, 2005.
- [5] . Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In Ales Leonardis, Horst Bischof, and Axel Pinz, editors, ECCV 2006, volume 3952 of Lecture Notes in Computer Science, pages 428–441. Springer Berlin / Heidelberg, 2006.
- [6] Ning He, Jiaheng Cao, and Lin Song. Scale space histogram of oriented gradients for human detection. In Proceedings of the 2008 International Symposium on Information Science and Engineering - Volume 02, pages 167–170, Washington, DC, USA, 2008. IEEE Computer Society.
- [7] Suard, F., Rakotomamonjy, A., Bensrhair, A., Broggi, A. (2006, June). Pedestrian detection using infrared images and histograms of oriented gradients. In Intelligent Vehicles Symposium, 2006 IEEE (pp. 206-212). IEEE
- [8] Viola, P., Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, pp. I-I). IEEE.
- [9] Viola, P., Jones, M. J. (2004). Robust real-time face detection. International journal of computer vision, 57(2), 137-154.
- [10] Jones, M., Viola, P. (2003). Fast multi-view face detection. Mitsubishi Electric Research Lab TR-20003-96, 3, 14.

## 7. Anexos

### 7.1. Algoritmo de viola-jones

Aunque no se logró comprobar cuantitativamente el resultado de aplicar el algoritmo de viola-jones en la base de datos estudiada debido a una restricción en la descarga del paquete visión en Matlab, se puede afirmar, basado en una investigación detallada del mismo, que este algoritmo es más rápido, eficiente y arroja mejores resultados

que el algoritmo de HOG multiescala. Esto se debe principalmente a que el algoritmo de viola-jones que se estudió, utilizaba un clasificador que había sido entrenado con miles de imágenes, lo que está relacionado directamente con el desarrollo de un clasificador más preciso lo cual disminuye considerablemente la aparición de falsos negativos y falsos positivos en los resultados [8][9][10].

### 7.2. Detectando a Waldo



Figure 10. Imagen-Waldo

El código implementado para hallar a Waldo en la base de datos utiliza la última fecha de modificación de cada imagen. Para este fin, se utilizó la función `dir` de Matlab, la cual genera la cual aporta información acerca de cada imagen, como la última fecha de modificación, el nombre, el número de bits. Se generó un cell array con todas las fechas, y otro con todos los nombres de imágenes. Posteriormente se generó otro array, organizando el array de fechas en orden descendente. El ultimo componente de este, corresponderá a la última imagen modificada, es decir, la imagen de Waldo. Luego se busca, este Sting de fechas, en el cell array original y se obtiene el índice que cumple esta condición. Finalmente se busca el nombre de imagen en este índice y se muestra en pantalla. // Se muestra el código a continuación

```
d = dir('TestImages');
d = d(3:end);
l = length(d);
nombres = {};
datenum = {};
for i = 1:l
    d2 = dir(fullfile('TestImages',d(i).name));
    dat = {d2.date};
    nom = fullfile('TestImages',d(i).name,{d2.name});
    nombres = cat(1,nombres,nom);
    datenum = cat(1,datenum,dat);
end
dates = sort(datenum);
fechaWaldo = dates{end};
```

```
f = find(strcmp(datenum, fechaWaldo));  
imagenWaldo = imread(nombres{f});  
imshow(imagenWaldo)
```