

إعداد:

نينورتا زياد بركات 2316

جعفر غسان رَيَّا 2061

منار عز الدين ضاحي 2213

## كشف المشاة باستخدام HOGs في Python

### ملخص:

نتيجة للحوادث المرورية وزيادة عددها أصبح من الضروري وضع حد لهذه الامور وكان من أحد الحلول هي تصميم نظم برمجية مختصة باكتشاف المشاة واعطاء انذار للمركبة بوجود مشاة أمامها. يتم تصميم هذه النظم باستخدام لغات البرمجة وتقنيات معالجة الصورة. في هذا المشروع سنقوم بتصميم نظام لكشف المشاة باستخدام لغة البرمجة بايثون وسنستفيد من مكاتبها المختصة بمعالجة الصورة وسنعمد على خوارزمية HOG لاكتشاف المشاة بحيث نقوم بفحص صورة لمعرفة إن كان فيها كائنات هم عبارة عن بشر في شارع.

الكلمات المفتاحية: HOG, opencv

## **Pedestrian Detection using HOGs in Python**

**Abstract:** As a result of traffic accidents and the increase in their number, it became necessary to put an end to these matters, and one of the solutions was to design software systems specialized in detecting pedestrians and giving a warning to the vehicle of the presence of pedestrians in front of it. These systems are designed using programming languages and image processing techniques. In this project, we will design a pedestrian detection system using the Python programming language, and we will take advantage of its image processing offices, and we will rely on the HOG algorithm to detect pedestrians, so that we examine an image to see if there are objects in it that are humans on a street.

**Keywords:** HOG, open cv

## مقدمة:

نعلم أنه يتم تزويد السيارات الحديثة بالعديد من أجهزة الاستشعار والرادارات والكاميرات الصغيرة للحصول على صورة أفضل لمحيط السيارة أثناء ركنها أو السير بها على الطرقات، وذلك بهدف التحرك السريع عند إمكان حصول أي حادث. ولكن ما هو تحديدا نظام الكشف عن المشاة والدراجات ؟

التكنولوجيا الأكثر انتشارا اليوم والمرتبطة بهذه الأجهزة الاستشعارية تركز في أغلب الأحيان على مهمة واحدة تتمثل في مراقبة موقع وسرعة الأجسام الكبيرة الصلبة التي تظهر أمام السيارة، أي اقترابها من مركبات أخرى أو شاحنات أو كرسي متحرك أو عربة أطفال أو غيره. بعد تطبيق هذا النظام، تبين أنه يواجه مشكلة في رصد الأجسام الصغيرة أو المتحركة بسرعة.

وفي سبيل البحث عن حل جذري لتأمين سلامة السائق من جهة ومحيط السيارة من جهة أخرى، تم اختراع وتنفيذ نظام الكشف عن المشاة والدراجات المصمم خصيصا لتتبع الأجسام المتحركة الأصغر حجما والأكثر تعقيدا التي تظهر فجأة في مسار سير السيارة. وبالتالي يقوم النظام بتحذير السائق واستخدام الفرامل إذا لزم الأمر بهدف تجنب عملية الاصطدام.

يستخدم النظام وحدة رادار مدمجة في مصبغة السيارة وكاميرا رقمية مدمجة في مرآة الرؤية الخلفية الداخلية، لمراقبة مجال الرؤية أمام السيارة لأمتار عدة. يتم إرسال البيانات من كلي النظامين إلى وحدة التحكم المركزية المخولة تمييز وتصنيف الأجسام المتحركة، وذلك عبر استخدام حجمها وسرعتها بالنسبة للسيارة نفسها للتمييز بين الاثنين.

يمكن للرادار والكاميرا تتبع تحركات كل من المشاة وراكبي الدراجات باستخدام مسارات السفر المتوقعة، وإذا كان خطر الاصطدام وشيك، يقوم النظام بإصدار إنذار مسموع كخطوة أولى، وفي حين غياب ردة الفعل المناسبة من السائق، سيتخذ النظام بنفسه الإجراء المناسب من تخفيض سرعة أو توقف تام عبر تطبيق الفرامل تلقائيا.

بين الرادار والكاميرا أدوار مختلفة، ولكن كلاهما يحتاج إلى تأكيد الكائن لكي يتخذ النظام إجراء ما: يقوم الرادار باكتشاف الأشياء وتتبع مسافة السيارة إليهم، مع أخذ سرعة السيارة في الاعتبار، في حين أن الكاميرا عالية الدقة تحدد نوع الكائن استنادا إلى نمط الحركة، الطول والحجم.

## أهمية البحث وأهدافه:

يفقد أكثر من 270 000 من المشاة حياتهم على الطرق، في كل عام على مستوى العالم، حيث يغادر الكثيرون منازلهم، كما جرت العادة يومياً، متوجهين إلى المدارس، وأماكن العمل، والورش، وزيارة الأصدقاء ولا يعودون أبداً. ويمثل المشاة 22 % من إجمالي الوفيات التي تحدث على الطرق على الصعيد العالمي، وفي بعض البلدان ترتفع هذه النسبة لتصل إلى ثلثي إجمالي هذه الوفيات. وعلاوة على ذلك، هناك الملايين الذين يصابون في تصادمات مرورية أثناء سيرهم، حتى أن البعض منهم يصابون بإعاقة مستديمة جراء هذه التصادمات التي تتسبب في الكثير من المعاناة والحزن، ناهيك عن الصعوبات الاقتصادية التي تواجهها أسرهم ومحبوهم. وبالتالي يجب البحث عن حلول لهذه المشكلة ومن هنا تكمن أهمية البحث في تصميم جهاز لكشف المشاة وتطبيق هذا الجهاز في السيارات والمركبات بشكل عام من أجل تخفيف كمية الحوادث المرورية.

## أدوات وطرائق البحث:

- 1- لغة بايثون
- 2- مكتبة open cv
- 3- مكتبة numpy
- 4- مكتبة imutils

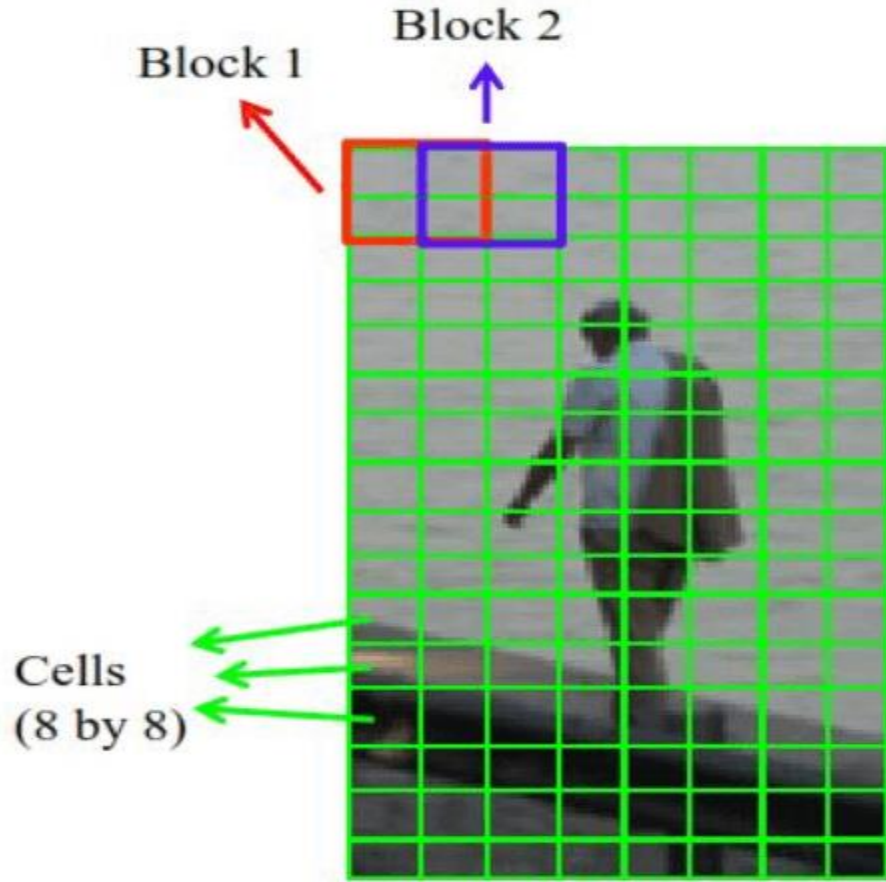
## المنهجيات العلمية:

### The Histogram of Oriented Gradients (HOG)

الرسم البياني للتدرجات الموجهة (HOG) هو واصف خاصة يستخدم في رؤية الكمبيوتر وتطبيقات معالجة الصور لغرض اكتشاف الكائن. إنها تقنية تحسب أحداث اتجاه التدرج في جزء معين من الصورة أو منطقة الاهتمام. في عام 2005 ، نشر دلال وتريغز ورقة بحثية بعنوان "مخططات بيانية للتدرجات الموجهة لاكتشاف الإنسان". بعد إصدار هذه الورقة ، يتم استخدام HOG في الكثير من تطبيقات الكشف عن الكائنات.

### فيما يلي أهم جوانب HOG:

- يركز HOG على هيكل الكائن. يستخرج معلومات حجم الحواف بالإضافة إلى اتجاه الحواف.
- يستخدم نافذة كشف تبلغ  $128 \times 64$  بكسل، لذلك يتم تحويل الصورة أولاً إلى شكل (64 ، 128).
- ثم يتم تقسيم الصورة إلى أجزاء صغيرة، ثم يتم حساب التدرج والاتجاه لكل جزء. وهي مقسمة إلى خلايا  $8 \times 8$  و16 وإلى كتل مع تداخل بنسبة 50% ، لذلك سيكون هناك  $15 \times 7 = 105$  كتلة في المجموع ، وتتكون كل كتلة من  $2 \times 2$  خلية مع  $8 \times 8$  بكسل.
- نأخذ 64 متجهاً متدرجاً لكل كتلة (خلية  $8 \times 8$  بكسل) ونضعها في مخطط بياني مكون من 9 صناديق.



فيما يلي الخطوات الأساسية التي نتخذها لاستخراج ميزة HOG:

- تغيير حجم الصورة

كما ذكرنا سابقًا ، إذا كانت لديك صورة عريضة ، فقم بقص الصورة إلى الجزء المحدد الذي تريد تطبيق استخراج ميزة HOG فيه، ثم قم بتغيير حجمها إلى الشكل المناسب.

- حساب التدرجات

الآن بعد تغيير الحجم ، علينا حساب الانحدار في اتجاهي  $x$  و  $y$ . التدرج اللوني هو ببساطة التغيرات الصغيرة في اتجاهي  $x$  و  $y$  ، نحتاج إلى ربط مرشحين بسيطين على الصورة.

مرشح حساب الانحدار في اتجاه  $x$  هو:

-1	0	1
----	---	---

ما يلي هو عندما نطبق هذا المرشح على صورة:



$f$

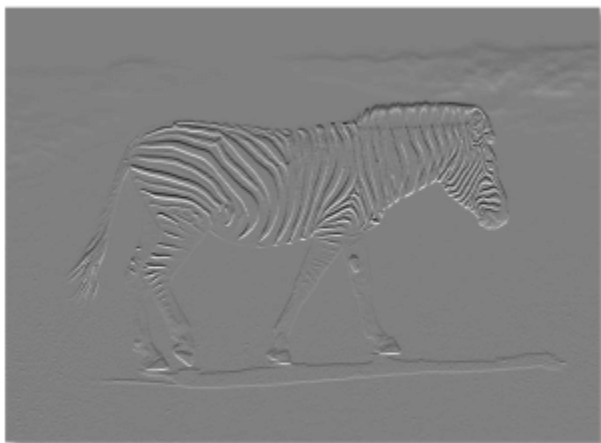


$\frac{\partial f}{\partial x}$

مرشح حساب الانحدار في الاتجاه  $y$  هو:

-1
0
1

عامل التصفية  $Y$  ما يلي عندما نطبق هذا المرشح على صورة:

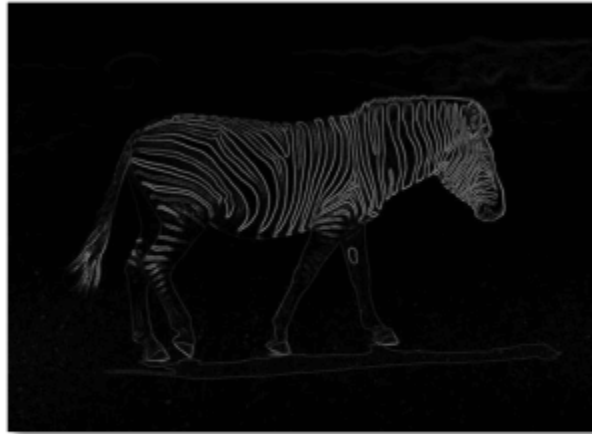


$$\frac{\partial f}{\partial y}$$

• حساب المطال

لحساب حجم التدرج اللوني ، يتم استخدام الصيغة التالية:

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$



$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

• حساب الاتجاه

يتم إعطاء اتجاه التدرج من خلال:

$$\theta = \tan^{-1} \left( \frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$$

لنأخذ مثلاً ، لنفترض أن لدينا المصفوفة أدناه:



	93	
56		94
	55	

سيكون التدرج في المحور x ببساطة  $38 = 56 - 94$  ، و  $38 = 55 - 93$  في المحور y.

المطال سيكون:

$$\nabla f = \begin{bmatrix} 38 \\ 38 \end{bmatrix}$$

$$|\nabla f| = \sqrt{(38)^2 + (38)^2} = 53.74$$

وسيكون اتجاه التدرج:

$$\arctan(38 \div 38)$$

## القسم العملي

### 1- المكتبات المستخدمة:

#### ○ مكتبة opencv:

أوبن سي في (OpenCV) أو المكتبة البرمجية المفتوحة للرؤية الحاسوبية هي مكتبة اقترانات برمجية تهدف بشكل أساسي لتطوير الرؤية الحاسوبية، طورتها شركة إنتل (Intel) وتعاقب على دعمها بعد ذلك شركتي ويلو غارج (Willow Garge) للروبوتيات وإتسير (Itseez) وأخير شركة إنتل مجدداً بعد استيلائها على شركة Itseez منذ 2016. المكتبة مجانية كونها تحت رخصة المصدر المفتوح (open source BSD license) ويمكن استخدامها على معظم الأنظمة الحاسوبية التي تدعم لغة سي وسي++. تركّز بشكل أساسي على معالجة اللحظة للصور (real-time).

يتم تنزيل المكتبة المتوافقة مع بايثون باستخدام التعليمات التالية:

**pip install opencv-python**

#### ○ مكتبة numpy:

مكتبة Numpy من المكتبات التي لا تستطيع تجاهلها عند خوضك غمار علم البيانات وذلك لسببين رئيسيين، الأول هو أن العديد من مكتبات علم البيانات وتعلم الآلة تعتمد اعتماداً قوياً عليها، والسبب الثاني هو أن هذه المكتبة تتيح لك قدرة التعامل مع المصفوفات بطريقة أفضل من ال Lists الموجودة تلقائياً كشكل من أشكال تراكيب البيانات في البايثون.

تتمثل هذه الأفضلية التي تُقدمها Numpy على ال Lists في أن المصفوفات في مكتبة Numpy سريعة في عمليات القراءة والكتابة وتُعتبر أكثر كفاءة ومتانة.

تُعد مصفوفة Numpy أحد أهم تراكيب البيانات في مكتبة Numpy والتي سُميت بهذا الاسم إختصاراً لمصطلح Numerical Python. ومن المصطلح ندرك أن هذه المكتبة هي مكتبة متخصصة في الحوسبة العلمية بلغة البايثون، وتحتوي على تشكيلة متنوعة من الأدوات والتقنيات التي من الممكن ان تستخدمها لحل مشاكل رياضية في مجالي العلوم والهندسة. أحد أهم هذه الأدوات في المكتبة هي المصفوفة ذات الأبعاد المتعددة والكفاءة الفائقة (High Performance Multidimensional Array)، والتي

تستطيع أن تُطبق عليها الكثير من الدوال الرياضية والعمليات الحسابية لتُعطيك القدرة على حل العديد من المشاكل.

يتم تنزيل المكتبة باستخدام التعليمة التالية:

**pip install numpy**

○ مكتبة **imutils**:

imutils عبارة عن تغليف يعتمد على OpenCV لتحقيق الغرض من استدعاء واجهة OpenCV بشكل أكثر إيجازًا. يمكنه بسهولة تنفيذ سلسلة من العمليات مثل ترجمة الصور ، التدوير ، القياس ، والهيكلة.

يتم تنزيل المكتبة باستخدام التعليمة التالية:

**pip install imutils**

## 2- البرنامج

سنقوم باكتشاف المشاة باستخدام HOG تعدد HOGs كاشفات ميزات رائعة ويمكن استخدامها أيضًا لاكتشاف الكائنات باستخدام SVM ولكن نظرًا للعديد من خوارزميات اكتشاف الأشياء الحديثة مثل YOLO و SSD الموجودة هناك ، فإننا لا نستخدم HOGs كثيرًا لاكتشاف الكائنات.

السطر 1-4 - استيراد المكتبات المطلوبة.

```
import cv2
from imutils.object_detection import non_max_suppression
from imutils import resize
import numpy as np
```

السطر 6 - صنع كائن HogDescriptor.

```
hog = cv2.HOGDescriptor()
```

السطر 7 - استيراد المصنف الافتراضي للكشف عن الأشخاص من مكتبة HOG. هذه هي الخطوة الرئيسية حيث نقوم باستيراد نموذج مدرب مسبقاً للكشف عن المشاة باستخدام HOG.

```
hog.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())
```

السطر 9-10 - اقرأ الصورة وقم بتغيير حجمها.

```
img = cv2.imread('P.jpg')
img = resize(img,height=500)
```

السطر 12 - لنكتشف الأشخاص.

```
rects,weights =
hog.detectMultiScale(img,winStride=(4,4),padding=(8,8),scale=1.05)
```

السطر 14 - الاحتفاظ بنسخة من الصورة الأصلية لاستخدامها في المستقبل.

```
copy = img.copy()
```

السطر 15-16 - رسم المستطيلات حول الأشخاص (إن وجدت).

```
for x,y,w,h in rects:
    cv2.rectangle(copy,(x,y),(x+w,y+h),(0,0,255),2)
```

السطر 18-19 - إظهار النتائج.

```
cv2.imshow('before suppression',copy)
cv2.waitKey(0)
```

السطر 21-22 - تنفيذ الحد الأقصى من الكشف أي إزالة التشويش وبالتالي إزالة المستطيلات حول العناصر الغير مرغوبة.

```
r = np.array([[x,y,x+w,y+h] for x,y,w,h in rects])  
pick = non_max_suppression(r, probs=None, overlapThresh=0.65)
```

السطر 24-25 - الآن رسم هذه المستطيلات الجديدة مرة أخرى بعد إزالة المستطيلات حول العناصر الغير مرغوبة.

```
for xa,ya,xb,yb in pick:  
    cv2.rectangle(img, (xa,ya), (xb,yb), (0,255,0), 2)
```

السطر 27-28 - إظهار النتائج.

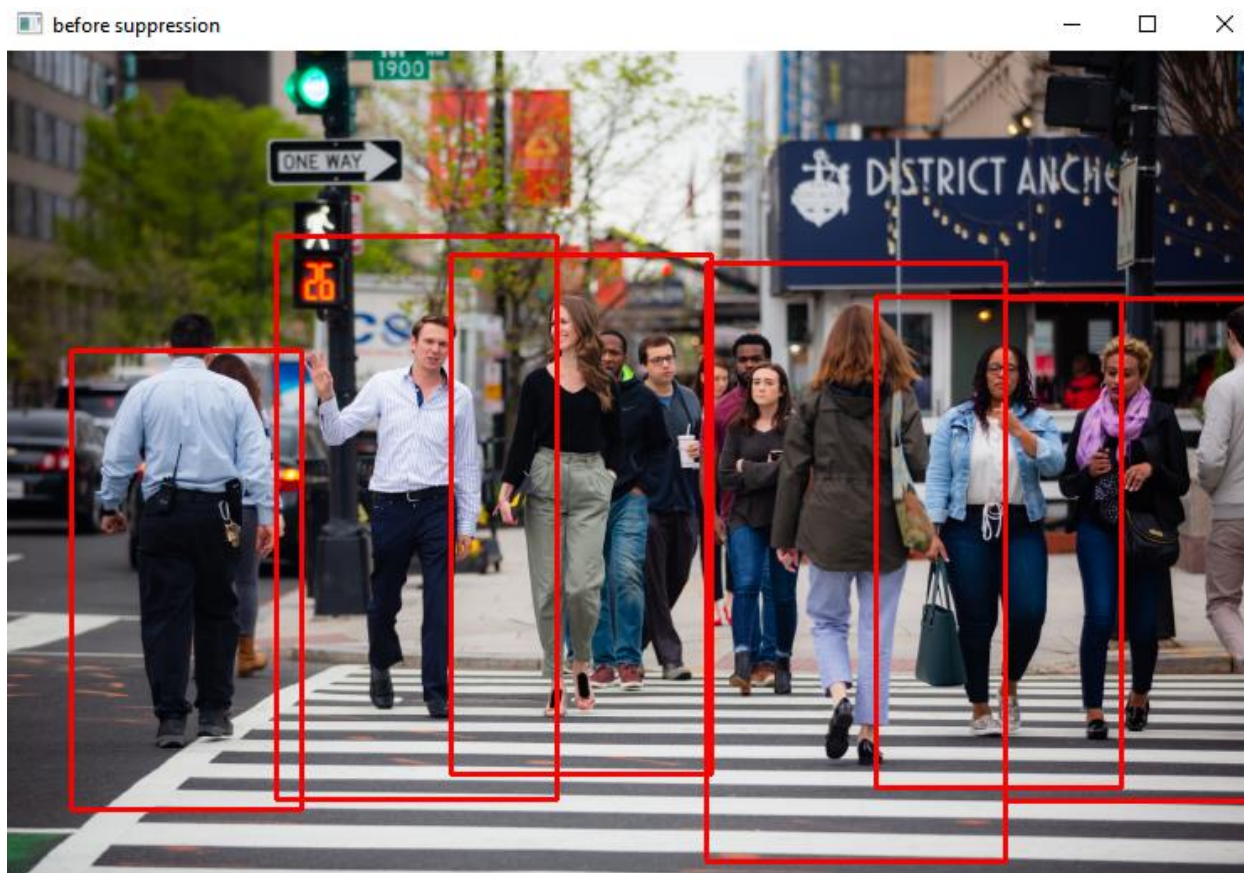
```
cv2.imshow('after suppression',img)  
cv2.waitKey(0)
```

السطر 30 -إغلاق جميع النوافذ المفتوحة.

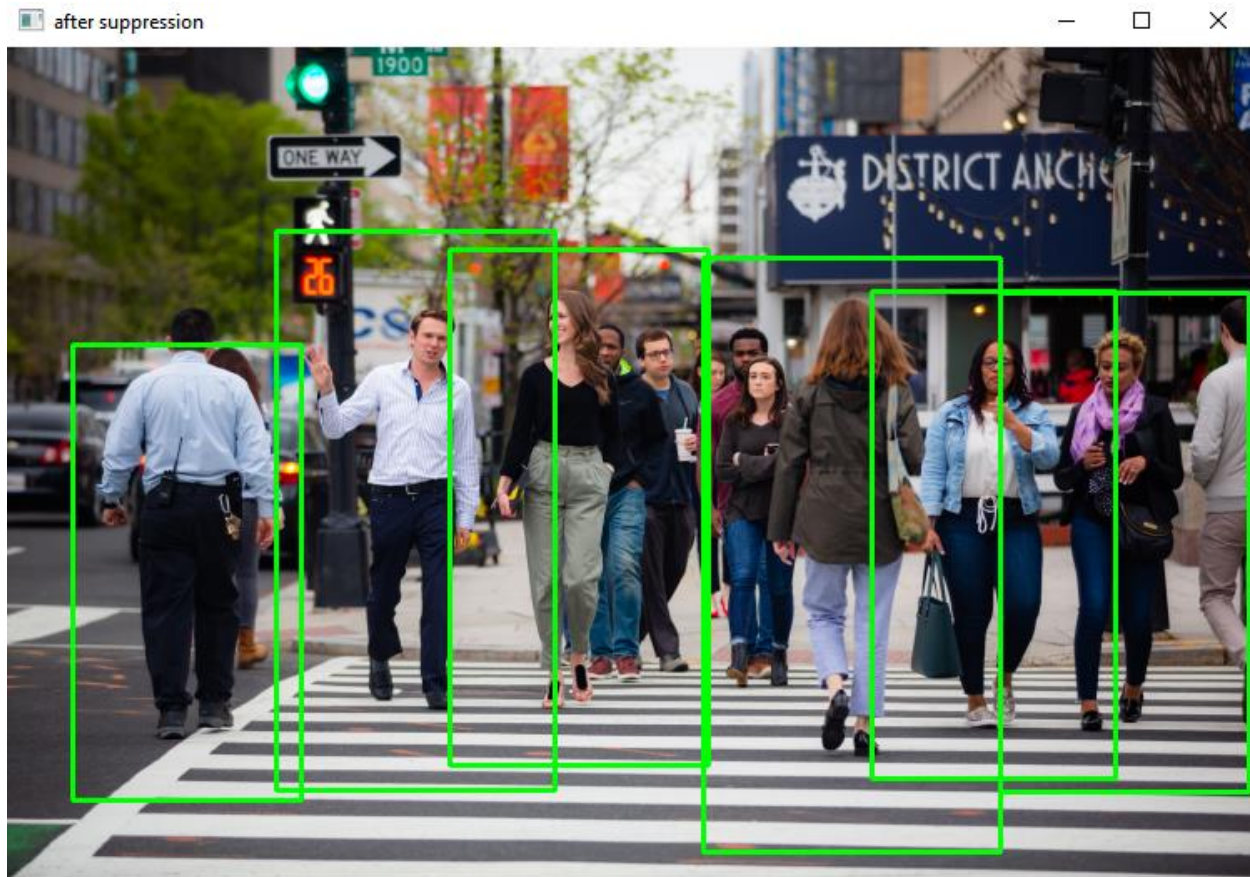
```
cv2.destroyAllWindows()
```

## النتائج والمناقشة:

بالنسبة للصورة P.jpg نلاحظ أنه يتم كشف الأشخاص بدقة دون وضع مستطيلات حول كائنات غير الأشخاص في الصورة:



الشكل 1 كشف الأشخاص قبل ازالة التشويش



الشكل 2 كشف الأشخاص بعد إزالة التشويش

بالنسبة للصورة P2.jpg نقوم بتغيير اسم الصورة ضمن الكود ونقوم بتشغيل الكود مرة أخرى فنلاحظ قبل إزالة التشويش يتم رسم مستطيلات حول أشياء لا تمثل أشخاص:





الشكل 3 الصورة الثانية قبل إزالة التشويش

يمكننا أن نرى أن غالبية المربعات الموجودة في الصورة يتم إخفاؤها في صورة ما بعد إزالة التشويش.



الشكل 4 الصورة الثانية بعد إزالة التشويش



## الاستنتاجات:

يمكننا أن نرى أنه نظرًا لأننا أجرينا اكتشاف المشاة باستخدام HOG ، فإننا لا نحصل على نتائج دقيقة للغاية. بحيث أنها تلتقط بعض الضجيج أيضاً. لهذا السبب نستخدم YOLO أو SSD لاكتشاف الكائنات في الوقت الحاضر حيث الدقة هي مطلبنا الرئيسي.

## التوصيات:

- يمكن تصميم نظام يعطي دقة أكثر وذلك باستخدام تقنيات تعلم الآلو وتدريب نماذج على بيانات مكونة من صور في إضاءة منخفضة لزيادة دقة الكشف.
- يمكن تصميم نظام كشف مشاة يعمل في الزمن الحقيقي.

## المراجع:

- 1- <https://www.thepythoncode.com/article/hog-feature-extraction-in-python#:~:text=The%20Histogram%20of%20Oriented%20Gradients,image%20or%20region%20of%20interest.>
- 2- <https://python.hotexamples.com/examples/skimage.feature/-/hog/python-hog-function-examples.html>
- 3- [https://www.w3schools.com/python/numpy/numpy\\_intro.asp#:~:text=What%20is%20NumPy%3F,in%202005%20by%20Travis%20Oliphant.](https://www.w3schools.com/python/numpy/numpy_intro.asp#:~:text=What%20is%20NumPy%3F,in%202005%20by%20Travis%20Oliphant.)
- 4- <https://numpy.org/>
- 5- <https://www.geeksforgeeks.org/opencv-python-tutorial/>
- 6- [https://docs.opencv.org/3.4/d0/de3/tutorial\\_py\\_intro.html](https://docs.opencv.org/3.4/d0/de3/tutorial_py_intro.html)
- 7- <https://pypi.org/project/imutils/>