

```
import cv2
```

(1) تضمين مكتبة open CV وهي المكتبة المسؤولة عن الرؤية الحاسوبية ومعالجة الصور الرقمية، هي مكتبة مفتوحة المصدر.

```
from playsound import playsound
```

(2) تضمين مكتبة Playsound وهي مكتبة متخصصة لتشغيل الصوتيات.

```
fire_cascade =  
cv2.CascadeClassifier('fire_detection.xml')
```

(3) إنشاء متغير بعنوان (`fire_cascade`) يستخدم بعدها خوارزمية التعلم الآلي، تستخدم لتحديد الكائنات في صورة أو مقطع فيديو، وتستند إلى مفهوم الميزات التي اقترحها بول فيولا ومايكل جونز في ورقتهما البحثية بعنوان "اكتشاف الكائنات السريعة باستخدام سلسلة معززة من الميزات البسيطة" في عام 2001.

إنه نهج قائم على التعلم الآلي حيث يتم تدريب الشبكات العصبونية على الكثير من الصور الإيجابية والسلبية.

ثم يتم استخدام هذا النهج (الخوارزمية) للكشف عن الكائنات في الصور الأخرى.

(`fire_detection.xml`): وهي قاعدة البيانات المستخدمة في البرنامج، وبمعنى أدق هي الخوارزمية المستخدمة في التعرف على النار.

```
cap = cv2.VideoCapture(0)
```

4) التعليمة في السطر الرابع لقراءة الفيديو من الكاميرا الأساسية للحاسوب (0): الرقم صفر يعني الكاميرا الأساسية، ووضعها في متغير (`cap`).

```
while (True):
```

5) حلقة (`while`) التكرارية، شرطها (`True`) بمعنى تكرار لانهائي للحلقة، الغرض من الحلقة التكرارية هو تقطيع الفيديو المسجل بواسطة الكاميرا الى عدة فريمات، وبالتالي يمكن اجراء عمليات معالجة الصورة وتحليل البيانات فيها حسب الخوارزميات المختلفة.

```
ret, frame = cap.read()
```

6) وهذا يعني قراءة الفيديو المسجل بواسطة الكاميرا في المتغير (`cap`) على شكل فريمات أي صور متجزئة من الفيديو، وتخزين اول صورة بمتحول (`frame`) تجهيزا لعرضها على الشاشة.

(`ret`): -اختصار لكلمة (`return`)- أي إعادة القيمة التي يقرأها البرنامج للمتغير السابق.

في كل عملية تكرار للحلقة المغلقة يتم المرور على فريم واحد من الفيديو ,وبالتالي تكرار الصور بشكل سريع يعطي انطباع أنه فيديو.

```
gray = cv2.cvtColor(frame,  
cv2.COLOR_BGR2GRAY)
```

(7) في متغير بعنوان (gray) نحول الصورة المتجزئة من الفيديو (frame) إلى اللون الرمادي.

(cv2.cvtColor): تابع تحويل الألوان (convert color)، وهي تابع ضمن (cv2)، وذلك من أجل تقليص المصفوفات اللونية إلى مصفوفة واحدة.

(COLOR_BGR2GRAY): من RGB إلى الرمادي.

```
fire = fire_cascade.detectMultiScale(frame,1.2,5)
```

(8) في متغير جديد يدعى (fire): نكتب امر التعرف المتعدد (detectMultiScale) على النار بواسطة الخوارزمية السابقة الموجودة في المتغير (fire_cascade)، وذلك عند الصورة المرغوبة (frame).

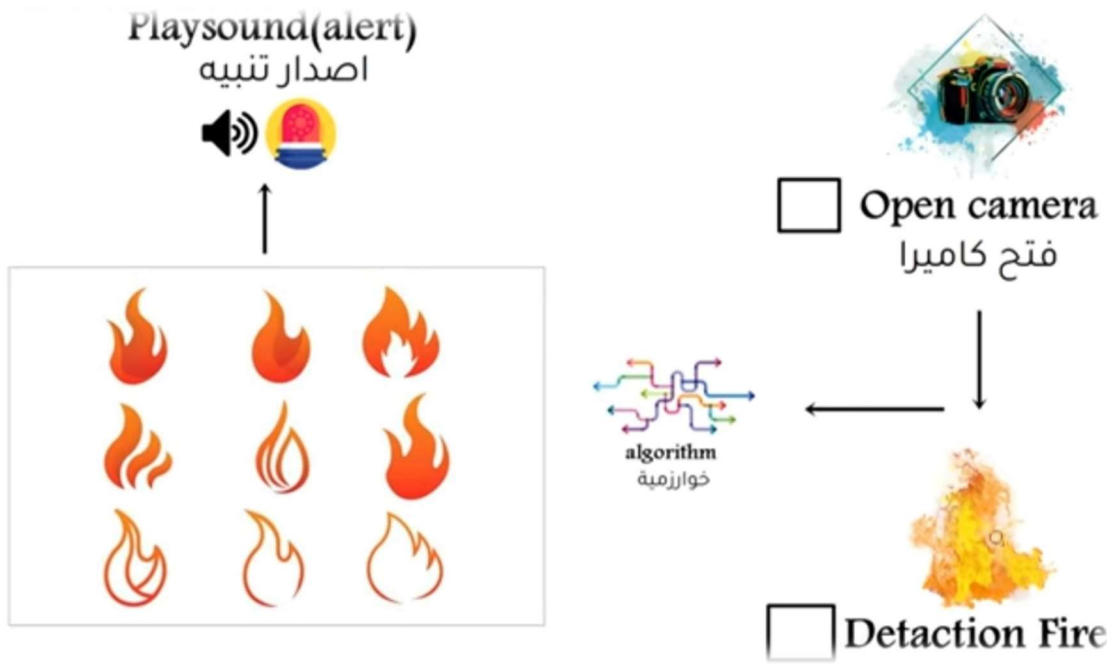
الخرج (fire) يكون عبارة عن مربع الصورة التي تظهر فيه النار: تتمثل بأربع قيم هي **الاتجاه من اليسار والاتجاه من الأعلى والعرض والارتفاع**.

الخاصية (1.2): وهي تمثل القيمة للعمق المراد تحديد ورصد النيران عنده على مستوى البكسل.

عادة يعتمد اختيار هذا الرقم على خبرة المهندس المصمم للمشروع.

الخاصية (5): وهي تمثل عدد العوامل المتشابهة للنيران، أي انه يجب ان تنطبق خمس عوامل على الأقل لنقول ان ما نراه في الفريم هو نار.

المقصود بالعوامل: درجة السطوع، الشكل الخارجي للنار ضمن المربع، الخ...



يبين الشكل بعض الاختلافات التي قد تتواجد في النيران.
يبين أيضا آلية سير البرنامج.

```
for (x,y,w,h) in fire:
```

9) انشاء حلقة تكرارية نوع (for) فيها المتغيرات التالية:
ال (x) تمثل الاتجاه من اليسار - رقم البكسل من اليسار -.

ال (y) تمثل والاتجاه من الأعلى -رقم البكسل من اعلى-.

ال (w) تمثل العرض.

ال (h) تمثل الارتفاع.

من اجل هذه المتغيرات في المحدد الذي يوجد فيه النار داخل الصورة (fire) سيتم تكرار الحلقة، بمعنى التكرار يتم عند التعرف على نار فقط.

ملاحظة: الأوامر من 10 الى 13 هي الأوامر التي تنفذ داخل حلقة (for).

```
roi_gray = gray[y:y+h, x:x+w]
```

10) في متحول جديد (roi_gray): نأخذ الفريم الرمادي (gray) ونحدد منه مربع البكسلات التي تعرض النار من (y) الى (y+h) في الارتفاع ومن (x) الى (x+w) في العرض.

```
roi_color = frame[y:y+h , x:x+w]
```

11) في متحول جديد (roi_color): نأخذ الفريم الملون (frame) ونحدد منه مربع البكسلات التي تعرض النار من (y) الى (y+h) في الارتفاع ومن (x) الى (x+w) في العرض.

```
print('Fire is detected')
```

12) طباعة على الكونسول أن هناك حريق تم التقاطه.

```
playsound('1.mp3')
```

13) تشغيل صوت الإنذار.

```
cv2.imshow('window_name', frame)
```

14) وهو أمر العرض الخاص للفريم الذي تم التقاط الحريق فيه، ويفضل ان يكون هذا الأمر قبل تشغيل الصوت لكي يعمل بشكل مثالي.

```
if cv2.waitKey(1) & 0xFF == ord('q'):  
    break
```

15) السطران الاخيران دالة شرطية (if):

القسم الأول من الشرط ((cv2.waitKey(1)): وذلك يعني طالما انه هناك امر بتشغيل النافذة التي تعرض الفيديو.

(&): الدالة المنطقية التي تدل على بوابة and، حرف (و).

القسم الثاني من الشرط ((0xFF == ord('q')): وهذا يعني إذا تم ضغط المحرف (q) على لوحة المفاتيح سيتم إيقاف الحلقة اللانهائية (while) بواسطة التابع (break).

إذا هناك شرطان لإيقاف البرنامج يجب ان يتحققا معا وهما: ان يكون الفيديو يعمل والضغط على المحرف (q).