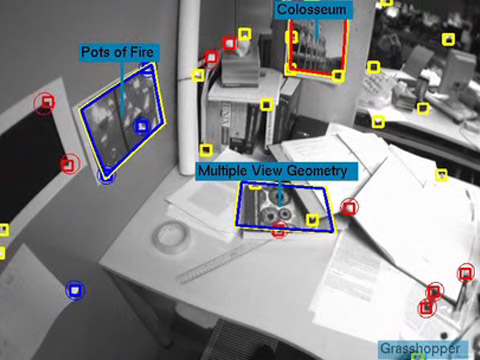
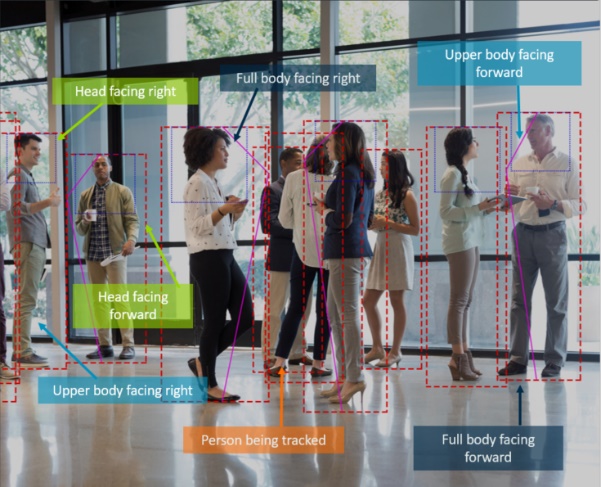
**¿Por qué se seleccionó el proyecto?**

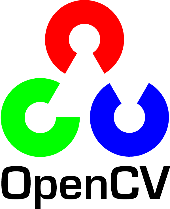
El proyecto seleccionado actualmente no era el que teníamos en mente, se había visto proyecto llamado BlinkToText el cual constaba de un reconocimiento facial el cual detectaba cuando estaban los ojos cerrados y posteriormente seleccionaba letras de la pantalla, esto para personas que no pueden utilizar sus extremidades. Pero al no poder realizarlo ya que otro equipo lo dio de alta primero optamos por buscar detección de objetos incluyendo la cara.

Encontramos proyectos donde se reconocía el rostro en imágenes lo cual nosotros quisimos pasar a lo que son videos ya sean en vivo (webcam) o pueden ser almacenados en la computadora y también como inicialmente se mencionó, imágenes.

En sí nos pareció interesante el cómo poder reconocer objetos solamente con mostrárselos a la computadora y fue la principal causa para seleccionar dicho proyecto.



El proyecto en sí se le encontró como **Face Detection** el cual consta de detectar caras a partir de una imagen que se le da al programa con ayuda de Open CV en el lenguaje de programación Python.

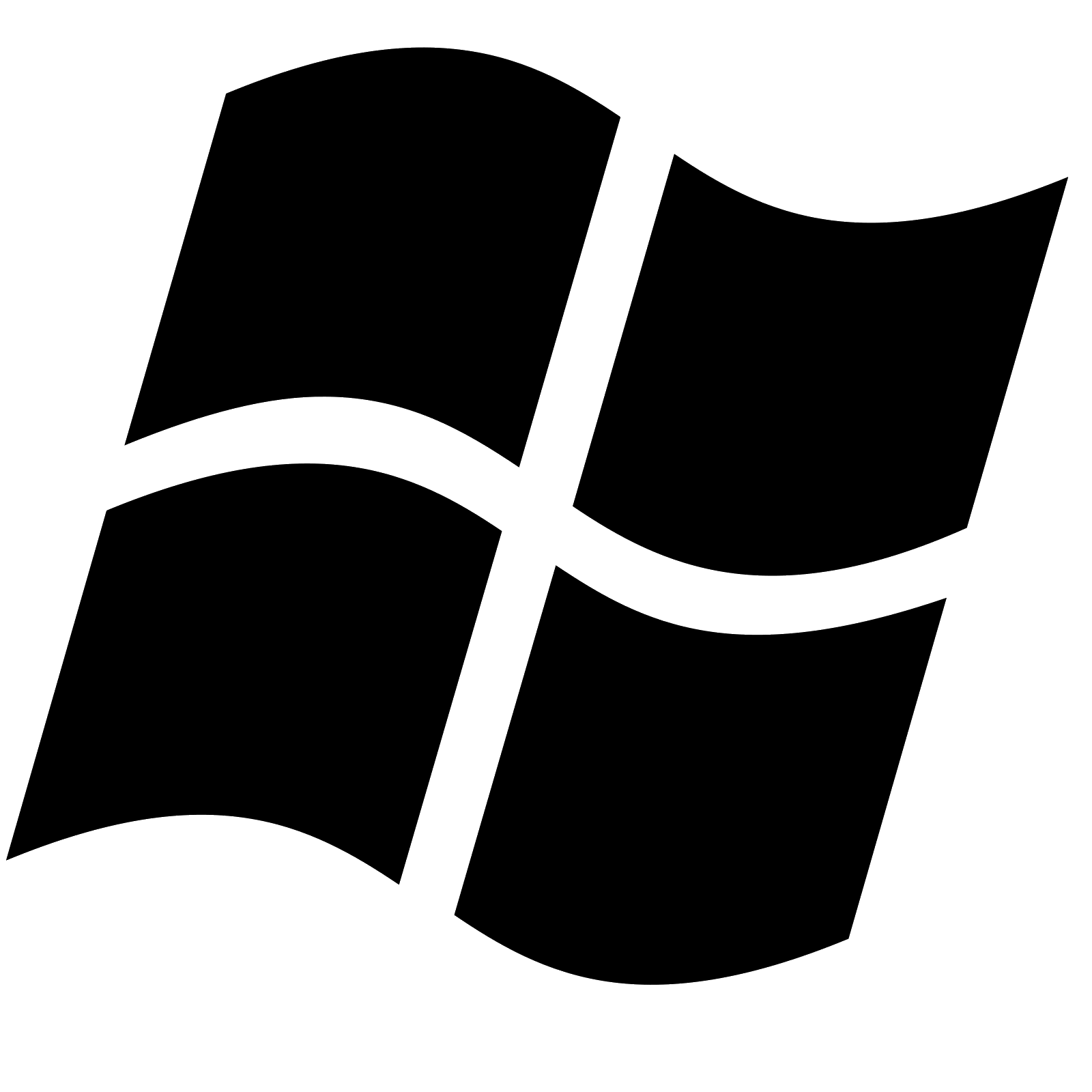




**¿Cómo se realizó el proyecto?**

Primero se necesitó echar le un vistazo al artículo del proyecto el cual nos hablaba sobre la detección de rostros con ayuda de Open CV, Python, en cualquier editor de texto.

Para esto se necesitó de instalar las librerías necesarias para Python (3.5 o 2.7 que fue las que probamos para realizar dicho proyecto) como lo son numpy y Open CV cada uno con sus comandos necesarios, el programa se probó tanto en Ubuntu como en Windows.

En **Windows** se obtuvo con:

* Open CV desde: <https://sourceforge.net/projects/opencvlibrary/files/>
* Numpy con: python -m pip install --user numpy scipy matplotlib ipython jupyter pandas sympy nose

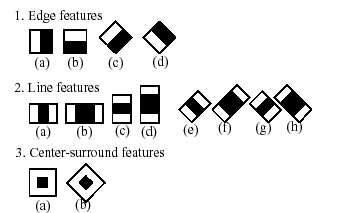
En **Ubuntu** se obtuvo con:

* Open CV con: sudo apt-get install python-opencv
* Numpy con: sudo apt-get install python-numpy python-scipy python-matplotlib ipython ipython-notebook python-pandas python-sympy python-nose

Una vez teniendo las librerías listas se pasó a probar dicho proyecto el cual consta de utilizar archivos HAAR, que son clasificados por clasificador HAAR, pero;

**¿Qué es el Clasificador HAAR?**

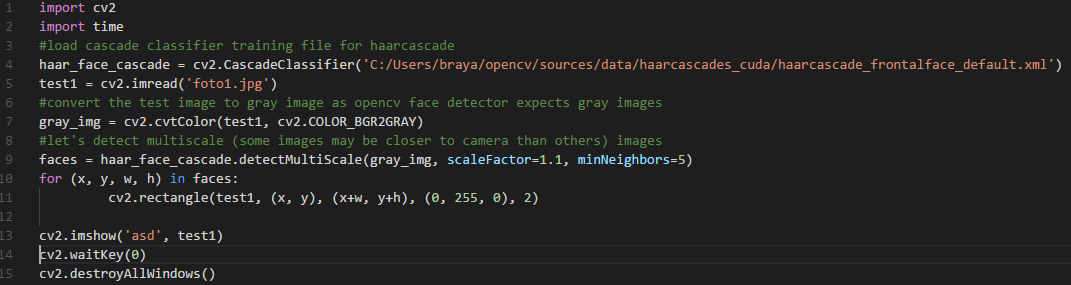
El clasificador HAAR es un algoritmo de machine learning creado por Paul Viola y Michael Jones en el cual se trata de contener imágenes positivas en una carpeta y en otra carpeta imágenes negativas de lo que se va a realizar, en este caso caras, en la carpeta positiva solo contendrá eso, y en la otra lo que sea menos caras.



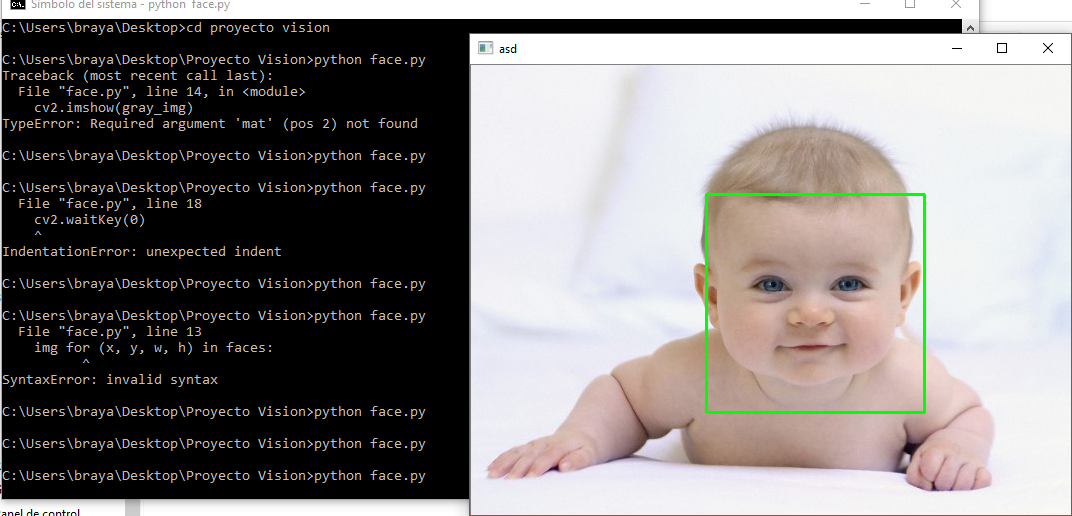
También se observo que existe otro algoritmo llamado LBP el cual también es bueno pero tenia diversas desventajas con el HAAR, el cual por ejemplo como ventajas el HAAR tiene una puntuación de acertar muy alta con un bajo falso positivo, pero en lo que es lo computacional, LBP es muchísimo más fácil, rápido y simple, pero por lo tanto menos eficiente para detectar objetos.

El proyecto nos muestra como se utilizo el HAAR y como obtienen la cara de un bebé en una foto, con la diferencia que ellos utilizaron la librería matploit pero ellos mismos explican que es opcional porque no es necesariamente útil.

Ejemplo:



Esto da como resultado esto:



Como se observa únicamente se detectan caras en dicho proyecto, por lo cual nosotros quisimos reconocer mas cosas como objetos, en nuestro caso hemos hecho archivos xml para lo que son ojos, sonrisa, bananas, plumas y carros.

**¿Cómo lo logramos?**

Siguiendo distintos tutoriales de los cuales nos decían que al tener una carpeta con fotos positivas y otras con fotos negativas se podría hacer todo el archivo xml para que fuera detectando cosas como lo fue con los rostros.

En una carpeta teníamos archivos de bananas por ejemplo: 

Mientras que en otra imágenes aleatorias que no tienen nada que ver con bananas, y una vez dicho esto se corre el programa