# VISIÓN ARTIFICIAL

## APLICACIONES PRÁCTICAS CON OPENCV - PYTHON

TOMÁS DOMÍNGUEZ MÍNGUEZ

Contenidos Web

Marcombo

# VISIÓN ARTIFICIAL

## Aplicaciones prácticas con OpenCV - Python

**Tomás Domínguez Mínguez**

Acceda a www.marcombo.info
para descargar gratis
***contenido adicional***
complemento imprescindible de este libro

Código: ARTIFICIAL3

# VISIÓN ARTIFICIAL

## Aplicaciones prácticas con OpenCV - Python

Tomás Domínguez Mínguez

Marcombo

*Visión Artificial. Aplicaciones prácticas con OpenCV - Python*

Primera edición, 2021

© 2021 Tomás Domínguez Mínguez

© 2021 MARCOMBO, S. L.
www.marcombo.com

Diseño de cubierta: ENEDENÚ DISEÑO GRÁFICO
Maquetación: cuantofalta.es
Asesor técnico: Ferran Fàbregas
Correctores: Beatriz Garcia y Manel Fernández
Directora de producción: M.ª Rosa Castillo

«Cualquier forma de reproducción, distribución, comunicación publica o transformación de esta obra solo puede ser realizada con la autorización de sus titulares, salvo excepción prevista por la ley. Diríjase a CEDRO (Centro Español de Derechos Reprográficos, www.cedro.org) si necesita fotocopiar o escanear algún fragmento de esta obra».

ISBN: 978-84-267-3347-4
Producción del ePub: booqlab

*Todo lo que sucede*
*fue una vez un sueño*

# TABLA DE CONTENIDO

# Unidad 1
# INTRODUCCIÓN

La visión artificial es una disciplina científica formada por un conjunto de técnicas que permiten la captura, el procesamiento y el análisis de imágenes. El objetivo es que un ordenador sea capaz de extraer información útil para responder a preguntas sobre su contenido, como ¿qué aparece en la imagen?, ¿hay algún objeto en el que esté interesado?, ¿es una persona conocida?, ¿qué está haciendo?

Las técnicas necesarias para conseguir dicho objetivo proceden de diversas áreas como la ingeniería o la informática, cuya naturaleza matemática ha provocado que solo estuvieran al alcance de especialistas en la materia. Sin embargo, la aparición de librerías como OpenCV ha permitido ocultar los complejos algoritmos en los que se basan y ha democratizado su uso. Al igual que para ponerse al volante de un coche no es necesario ser mecánico, sino únicamente saber manejar los mandos que permiten conducirlo, para desarrollar aplicaciones de visión artificial no se tiene por qué haber estudiado una ingeniería, basta con conocer las funciones de aquellas librerías que permitan hacer lo que se pretenda, por ejemplo, identificar un determinado objeto, su color o su tamaño.

En este primer capítulo se darán unas pinceladas de lo que es la visión artificial, así como de la librería OpenCV, que pondrá a su alcance las técnicas necesarias para beneficiarse de todo lo que ofrece esta disciplina.

## 1.1 VISIÓN ARTIFICIAL

Hasta hace poco tiempo, las tareas que un ordenador era capaz de hacer con las imágenes consistían en almacenarlas, reproducirlas o, incluso, procesarlas para modificar su aspecto. Pero en ningún caso llegaba a obtener información que mostrara una compresión real de su contenido. Estas mismas imágenes que tanta información pueden ofrecer a una persona para un ordenador solo representaban píxeles que almacenan un nivel de luz o color.

El análisis que cualquiera de nosotros realizamos de forma natural cuando vemos algo ha estado vedado a los ordenadores. Afortunadamente, estas capacidades, antes relegadas exclusivamente al ámbito humano, comienzan a ser automatizadas gracias a la aplicación de las técnicas de visión artificial.

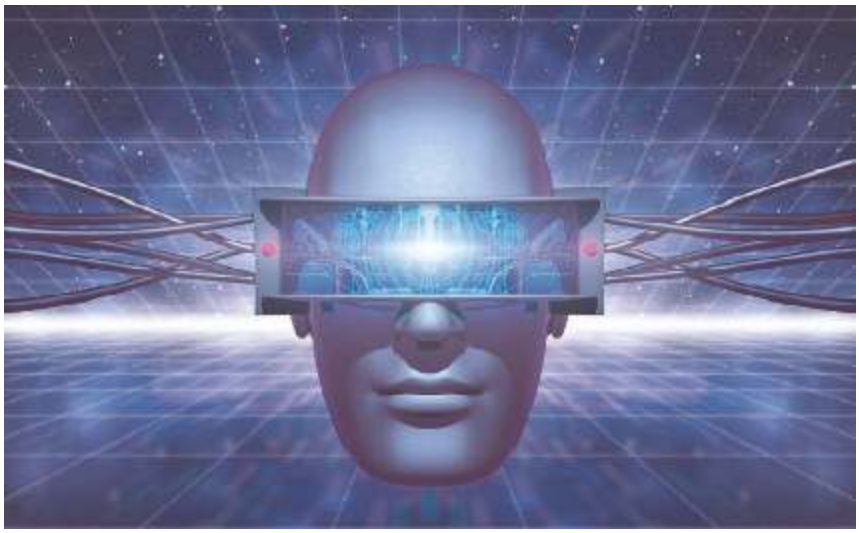
El objetivo de la visión artificial es interpretar las imágenes con el fin de extraer información de utilidad, por ejemplo, saber si delante de una cámara hay alguien conocido o un lugar de interés. Dicha información servirá para conseguir un fin, que en los ejemplos anteriores podría ser dar acceso a una aplicación (en vez de usar una clave) o proporcionar información turística relevante del sitio donde se encuentre. Ambos ejemplos son tan solo una muestra de los muchos campos en los que su empleo está ganando terreno. Y es más, con alguno de ellos experimentará a lo largo de las páginas de este libro:

- Seguridad. Como se acaba de comentar, la visión artificial puede utilizarse para desbloquear una aplicación, pero pronto comprobará que también permite el desarrollo de alarmas inteligentes que no se disparen cuando el movimiento detectado sea el de su mascota paseándose por la casa.

- Industria. Al comparar las imágenes de objetos manufacturados con las de otros utilizados como modelo, podrían automatizarse procesos de control de calidad. Aplicando estas mismas técnicas, realizará un programa de carácter más lúdico que podría ayudarle a solucionar pasatiempos, como el de las siete diferencias, que a veces tanto cuesta resolver.
- Comercio. Saber cómo quedaría un mueble en su sala de estar o el aspecto que tendría esta con las paredes pintadas de un determinado color son facilidades que pueden hacer que un cliente se decida a comprar un producto concreto. Dentro de esta área, se aborda la implementación de una aplicación orientada al negocio textil, en concreto, de venta de sombreros.
- Educación. Cada vez son más las aplicaciones de este tipo utilizadas como apoyo a la formación. A modo de ejemplo, se desarrolla un programa que podría emplearse para enseñar las principales figuras geométricas a los más pequeños, ya que es capaz de identificar las situadas delante de una cámara.

Por supuesto, la visión artificial puede aplicarse a muchos más campos, como la clasificación y búsqueda de imágenes, la conducción autónoma, la medicina, la agricultura, el turismo, los videojuegos, etc. Tratar de enumerarlos todos daría como resultado una extensa lista que quedaría rápidamente obsoleta, ya que va creciendo de día en día.

Para alcanzar el propósito perseguido en cada una de dichas áreas, esta disciplina se apoya, tal como se indicó al principio del capítulo, en tres pilares que permiten adquirir, procesar y analizar imágenes procedentes del mundo real. Veamos en qué consiste cada uno de ellos.



Mediante la adquisición de imágenes, el mundo analógico de luces y sombras que nos rodea se traduce a datos binarios compuestos por ceros y unos. Es lo que habitualmente se hace con una cámara (ya integrada en el ordenador o conectada por USB).

El procesamiento usa diferentes algoritmos matemáticos que, ocultos por librerías, resaltan determinadas características de las imágenes, preparándolas de esta forma al tipo de algoritmos de análisis que se lleve a cabo posteriormente. Operaciones como el recorte o el escalado de imágenes, el cambio de espacio de color o la aplicación de filtros permiten centrar la atención en los objetos relevantes, al facilitar la detección de sus bordes, esquinas, contornos o cualquier otro rasgo que los defina.

El análisis de las imágenes va un paso más allá del procesamiento. Basado también en algoritmos matemáticos (utilizados, de nuevo, a través de librerías que oculten su complejidad), partirá de los resultados del paso anterior para realizar un reconocimiento de objetos, su clasificación, identificación y seguimiento en la escena, por poner algún ejemplo.

En este libro aprenderá a usar la librería OpenCV para realizar estos procesos, y se ocultará la complejidad matemática de los algoritmos

en los que se basa cada uno de ellos. Eso le permitirá desarrollar, de forma sencilla y práctica, aplicaciones de visión artificial en tiempo real.

Adéntrese en este fascinante mundo con Pyhton de la mano de OpenCV.

## 1.2 OPENCV

OpenCV (Open Source Computer Vision Library) es una librería orientada al procesamiento y análisis de imágenes. Desarrollada inicialmente por Intel en 1999, ha ido ganando popularidad rápidamente, siendo hoy en día una de las más empleadas en el desarrollo de aplicaciones de visión artificial.


OpenCV (https://docs.opencv.org/) es muy utilizada en grupos de investigación, organismos gubernamentales y empresas como Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda y Toyota. Sus ámbitos de aplicación son muchos, y entre ellos destacan:

- Reconocimiento facial
- Identificación de objetos o personas

- Inspección y vigilancia
- Juegos y controles
- Recuento de objetos
- Análisis de imágenes médicas
- Robótica
- Realidad aumentada

En 2008, Willow Garage se hizo cargo del soporte de OpenCV 2.3.1, por lo que a partir de dicha versión viene con una interfaz de programación C++, Java y Python. Además, está disponible en diferentes plataformas, incluidas Windows, Linux, Mac OS X y Android. Este libro se centrará en OpenCV-Python, que combina las mejores cualidades del API (*application programming interface,* interfaz de programación de aplicaciones) de OpenCV y el lenguaje Python.

Además de ser multiplataforma y multilenguaje, es destacable el hecho de que OpenCV sea un producto con licencia BSD (Berkeley Software Distribution), con menos restricciones que otras como GPL (General Public License, Licencia Pública General de GNU), ya que permite incluso el uso de su código fuente en software comercial.

## **1.3** PYTHON

Como se acaba de comentar, OpenCV está disponible en los principales lenguajes de programación. De todos ellos, utilizará Python, quizás el empleado por un mayor número de programadores en el mundo. Los motivos de la enorme aceptación de este lenguaje seguramente estén en su sencillez de aprendizaje y facilidad de uso.

No solo permite el clásico modo de programación imperativo, sino que también admite una programación orientada a objetos, con lo que se abre al uso de infinidad de librerías que, como OpenCV, están basadas en este paradigma.

Su entorno multiplataforma hace que se pueda trabajar con él, independientemente de si dispone de un ordenador con Windows, Mac o Linux (incluso Raspberry). Como requisito adicional, necesitará disponer de una webcam, que podrá ser la integrada en el propio ordenador o cualquier otra conectada vía USB.

La versión de Python utilizada será la 3.x (a fecha de publicación del libro es la 3.7). No tiene sentido utilizar Python2 cuando fue oficialmente descontinuado el 1 de enero de 2020.

¿Todavía no conoce Python? Al final del libro dispone de un amplio anexo que representa en sí mismo un curso de iniciación a este lenguaje. En primer lugar, aprenderá cómo se instala y utiliza su entorno de desarrollo. Haciendo uso de él, practicará con los tipos de datos básicos, los operadores y las estructuras de datos o de control imprescindibles del lenguaje. La orientación a objetos tendrá un tratamiento especial, así como el desarrollo de funciones o el uso de módulos (librerías). Se explica todo lo necesario (incluso más) para que pueda entender el código de las múltiples prácticas

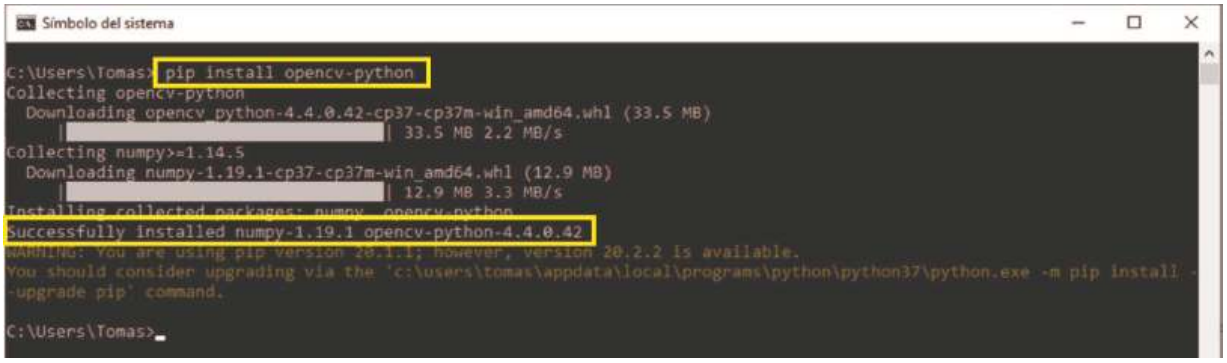
con las que se demuestran los conceptos de visión artificial introducidos en cada capítulo de la mano de OpenCV. El objetivo final es que, inspirándose en estos programas de prueba, usted pueda desarrollar sus propios proyectos. Los límites que pueda llegar a alcanzar estarán en su imaginación.

## Unidad 2
# INSTALACIÓN DE OPENCV

Puesto que va a trabajar con la librería OpenCV para Python, deberá tener instalado el entorno de desarrollo de este lenguaje en su ordenador. Si no fuera así, en el anexo final se indica cómo hacerlo. Aunque esta librería funciona tanto con la versión 2.7 como con la 3.4 y superiores, tal como se ha comentado en el apartado anterior, se utilizará únicamente Python 3.7.

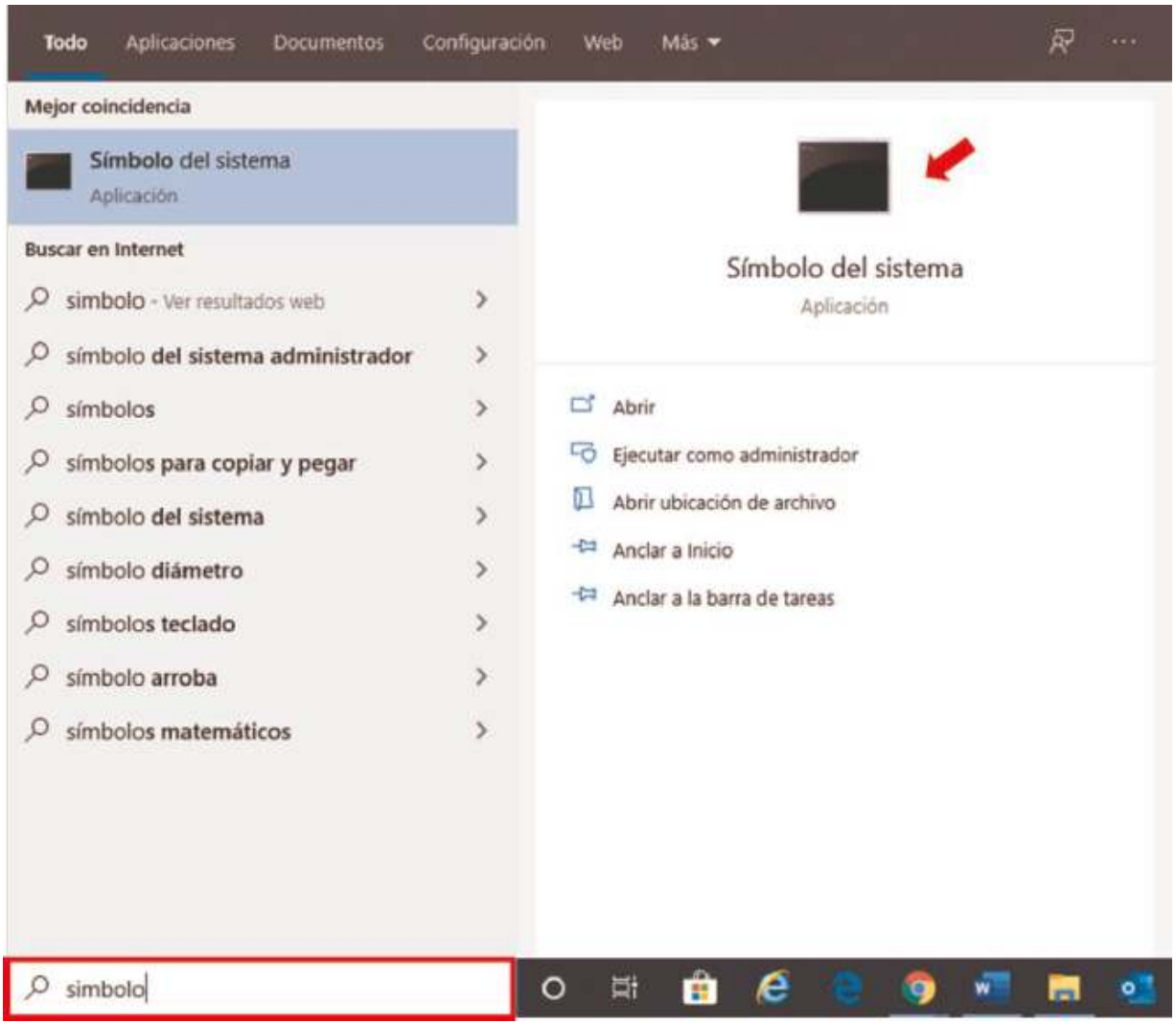
Evidentemente, para usar la librería OpenCV, primero hay que instalarla. Para ello, abra una ventana de símbolo del sistema y ejecute el comando:

```
pip install opencv-python
```




Recuerde que en algunas distribuciones de Linux el comando que debe utilizar es `pip3`.

La forma más sencilla de abrir una ventana de símbolo del sistema en Windows es escribiendo su nombre en el campo de búsqueda situado en la parte inferior izquierda del escritorio. Enseguida le aparecerá un icono sobre el que podrá pulsar para abrir dicha ventana.




En la instalación de este paquete, puede observar que también se carga la librería NumPy, utilizada para trabajar con matrices.

Aunque no entraremos en conceptos matemáticos, para entender por qué la visión artificial y las matrices están tan unidas, piense que una imagen puede definirse como una función *f(x, y)* cuyos argumentos son las coordenadas *x, y* de

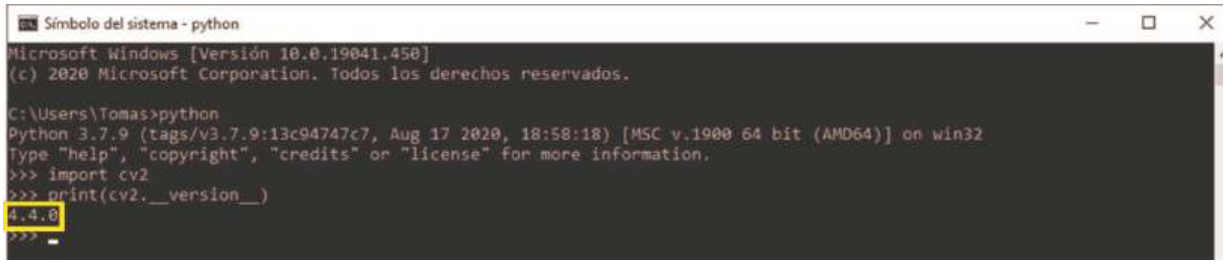
cada píxel, y el resultado es el nivel de luz (o color) del píxel situado en dichas coordenadas.

En otras palabras, una imagen no es más que un array de dos dimensiones (de tres en el caso de imágenes en color) cuyos elementos contienen el valor de cada uno de sus píxeles. A los arrays de dos dimensiones se les llama matrices. Si tuvieran tres serían tensores. Sin embargo, por simplicidad, generalmente se empleará el término matriz.

Para saber si OpenCV está correctamente instalado, entre en el intérprete de Python (comando `python`) y ejecute las siguientes sentencias:

```
python
>>>import cv2
>>>print(cv2.__version__)
```

El resultado debe devolver la versión de la librería instalada (en este caso la 4.4.0).




Toda la documentación de esta librería se encuentra en:

https://docs.opencv.org/4.4.0/

Aunque no es imprescindible, aproveche también para instalar la librería matplotlib ejecutando el siguiente comando:

```
pip install matplotlib
```

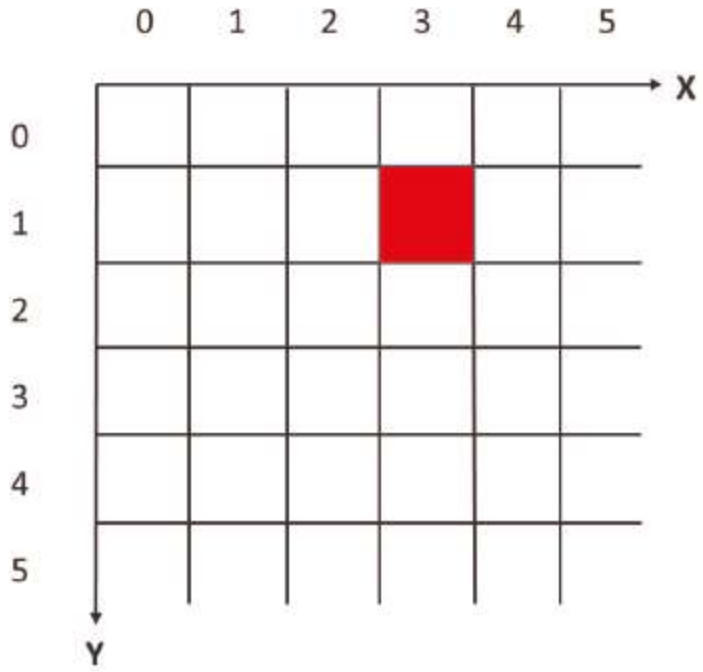
Dicha librería le permitirá generar gráficos a partir de datos contenidos en listas.

Por similitud con la terminología utilizada en otros lenguajes de programación, el término librería se utilizará como sinónimo de módulo o paquete (aunque no sean conceptos exactamente iguales).

## Unidad 3
# PRIMEROS PASOS

En los siguientes apartados estudiará las operaciones básicas de creación o carga de imágenes desde un archivo y su visualización en pantalla, así como la modificación de su contenido y posterior almacenamiento en disco.

Pero antes deberá conocer el sistema de coordenadas utilizado, en el que cada píxel se localiza en la imagen por su posición horizontal y vertical (*x, y*). Como puede ver en la siguiente figura, el origen (0, 0) no está localizado en el centro de la imagen, sino en la parte superior izquierda. El píxel marcado en color rojo estaría situado en la coordenada (3, 1).




Otro de los conceptos básicos previos al uso de las funciones de OpenCV es el color, formado por los componentes B, G, R (Blue, Green, Red; Azul, Verde, Rojo), o lo que es lo mismo, los valores de la intensidad de sus componentes azul, verde y rojo. Estos tres colores se consideran primarios porque cualquier otro color se puede obtener a partir de una mezcla de ellos. El valor mínimo de intensidad que pueden tener es 0, y el máximo, 255.

Seguramente le resulte más familiar el espacio de color RGB; por eso, preste especial atención al hecho de que OpenCV utiliza por defecto BGR. Más adelante sabrá cómo pasar de uno a otro o incluso a HSV, ampliamente utilizado en las técnicas de reconocimiento y seguimiento de objetos por su mayor eficiencia.

Ejemplos de especificación de colores utilizando la nomenclatura BGR serían:

- Blanco: (255, 255, 255)
- Negro: (0, 0, 0)

- Rojo: (0, 0, 255)
- Verde: (0, 255, 0)
- Azul: (255, 0, 0)
- Amarillo: (0, 255, 255)
- Púrpura: (255, 0, 255)
- Cian: (255, 255, 0)

Opcionalmente, un píxel puede tener asociado un cuarto parámetro que define su transparencia: el canal alfa. Sus valores van desde 0 (completamente trasparente) a 100 (totalmente opaco).

## 3.1 CARGA Y VISUALIZACIÓN DE LA IMAGEN ALMACENADA EN UN ARCHIVO

Para mostrar una imagen en pantalla, previamente hay que cargarla con la función de OpenCV:

```
imread(archivo)
```

Su único argumento contiene el nombre del archivo donde se encuentra la imagen (o la ruta, en el caso de que no esté en la misma carpeta que el programa). Esta función también admite un segundo argumento opcional para especificar la información que se carga de la imagen. Su valor podrá ser el de cualquiera de estas constantes:

- `IMREAD_GRAYSCALE`. Carga la imagen en blanco y negro. Su valor es 0.

- `IMREAD_COLOR`. Carga la imagen en color (se ignora el canal alfa). Su valor es 1.
- `IMREAD_UNCHANGED`. Carga la imagen incluyendo el canal alfa. Su valor es -1.

Si no se indica este argumento, la función toma por defecto el valor `IMREAD_COLOR`.

El canal alfa define la opacidad de un píxel, es decir, su grado de transparencia. No todos los formatos de imagen admiten este canal (que se añade a los tres correspondientes a los colores primarios). Así, por ejemplo, mientras JPG no lo permite, PNG sí contiene este tipo de información.

Entre los formatos de imágenes que se soportan destacan BMP, JPEG, PNG y TIFF, entre otros. OpenCV no tiene códec para GIF, por lo que el manejo de imágenes en este formato requiere la librería imageio, que tendrá que haber cargado previamente en su entorno Python. Si quiere saber cómo utilizarla, visite https://pypi.org/project/imageio/.

Esta función devolverá un objeto de la clase `ndarray`, perteneciente a la librería NumPy. Dicho objeto mantiene una matriz cuyos elementos establecen el nivel de luz o color de los píxeles de la imagen.

Una vez obtenida la imagen, la forma de mostrarla en pantalla es llamando a la función de OpenCV:

```
imshow(ventana, imagen)
```

El primer argumento determina el título de la ventana, mientras que el segundo contiene la imagen.

El tamaño de la ventana será el de la imagen. Si la resolución de esta fuera muy alta, solo se vería parte de ella.

En el siguiente programa se utilizan ambas funciones:

```
import cv2

img = cv2.imread('../imagenes/cuadro.jpg', 0)
cv2.imshow('Cuadro', img)
```

Con la primera sentencia se importa la librería OpenCV. A continuación, se carga la imagen "cuadro.jpg" (que está dentro de la carpeta "imagenes") con la función `imread()`. Su segundo argumento es 0, lo que significa que la imagen se obtiene en blanco y negro. La última sentencia la muestra en una ventana llamada "Cuadro".