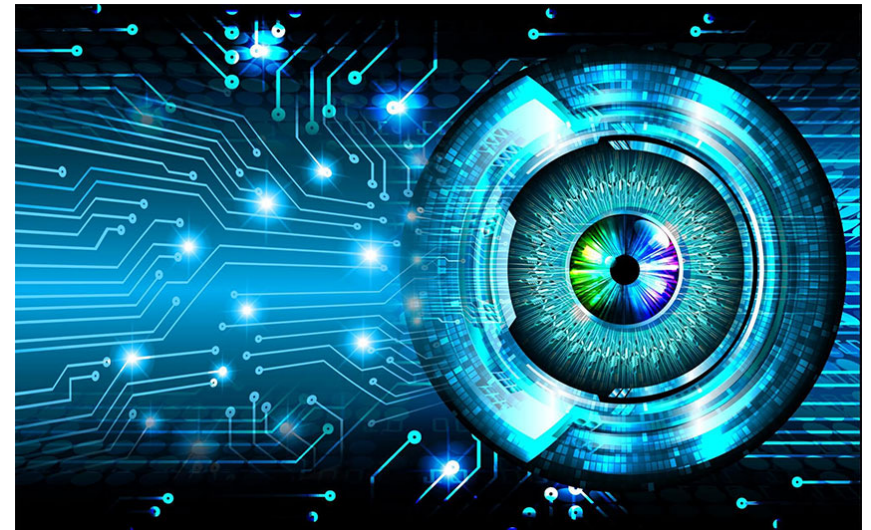


Laboratorio 3: Representación de una imagen

Laboratorio 3: Representación de una imagen



- La **visión artificial** o por computadora se está utilizando cada vez más para el **análisis y tratamiento de imágenes** mediante algoritmos de inteligencia artificial.
- Uno de los usos más importantes de OpenCV en la visión por computadora es la **detección de rostros y objetos**, sobre todo en ámbitos como la fotografía, el marketing o la seguridad.

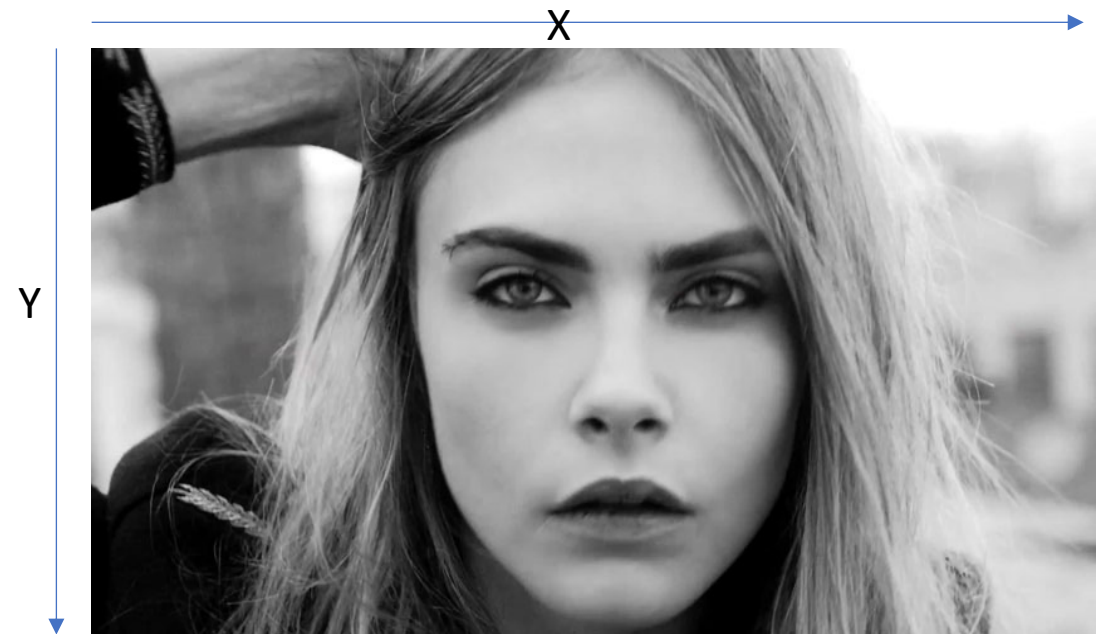


- **OpenCV** (Open Source Computer Vision) comenzó como un proyecto de investigación en Intel. Actualmente es la **biblioteca de visión por computadora** más grande en términos de funciones poseídas.
- Contiene implementaciones de **más de 2500 algoritmos**.
- Tiene interfaces para **múltiples lenguajes**, incluidos Python, Java y C++.
- **OpenCV-Python** es la **API** de Python para OpenCV. Compatible con las plataformas más utilizadas,
- Para poder usar esta librería de manera óptima, se deben tener conocimientos en:
 - ✓ Librería **Numpy**
 - ✓ Librería **Matplotlib**



Imágenes como matrices

- ✓ Una **imagen** no es más que una **matriz** estándar de Numpy que contiene **píxeles de puntos de datos**. Cuanto mayor sea el número de píxeles en una imagen, mejor es su resolución.
- ✓ Una imagen es una función de dos variables que nos arroja un valor. Puede verse como una matriz bidimensional que esta descrita como: Imagen: filaXcolumna->valor (cada uno de los valores se le conoce como pixel)
- ✓ El numero de pixeles de una imagen determina su resolución.



Matriz de MXN

-
- Para ser procesado por una computadora, una imagen debe convertirse en una **forma binaria**. El color de una imagen se puede calcular de la siguiente manera:

*Número de colores / sombras = 2^{bpp}
(donde bpp representa bits por pixel)*

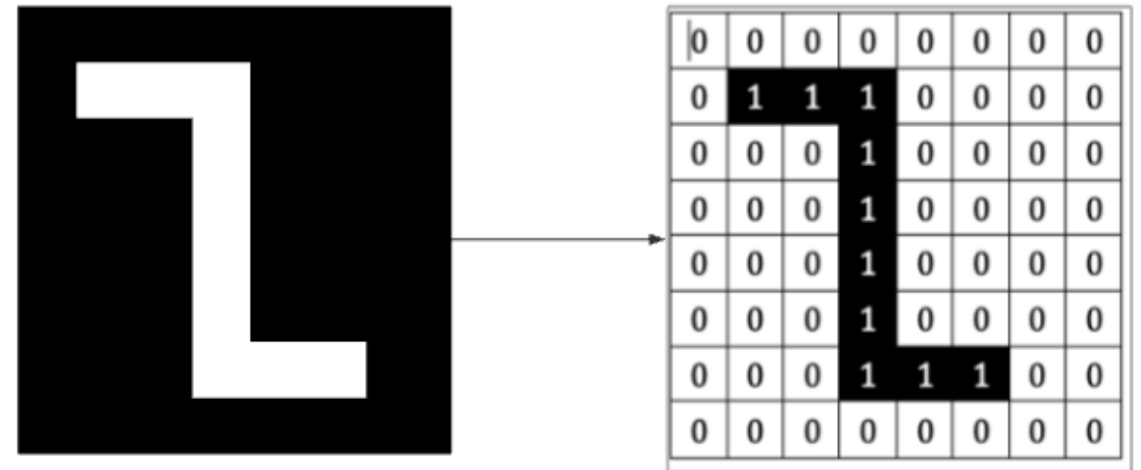
- Por lo tanto, cuanto más cantidad de bits/píxel, más colores posibles en las imágenes.

Bits/Pixel	Possible colours
1	$2^1 = 2$
2	$2^2 = 4$
3	$2^3 = 8$
4	$2^4 = 16$
8	$2^8 = 256$
16	$2^{16} = 65000$

Representación de los diferentes tipos de imágenes

Imagen binaria

Una imagen binaria consta de 1 bit/píxel y, por lo tanto, solo puede tener **dos colores posibles**, es decir, blanco (1) o negro (0).



Representation of a black and white image in form of a binary where '1' represents pure white while '0' represents black. Here the image is represented by 1 bit/pixel which means image can be represented by only 2 possible colours since $2^1 = 2$

Imagen en escala de grises

Una imagen en escala de grises consta de 8 bits por píxel. Esto significa que puede tener **256 sombras diferentes** donde 0 píxeles representarán el color negro mientras 255 denota el blanco.

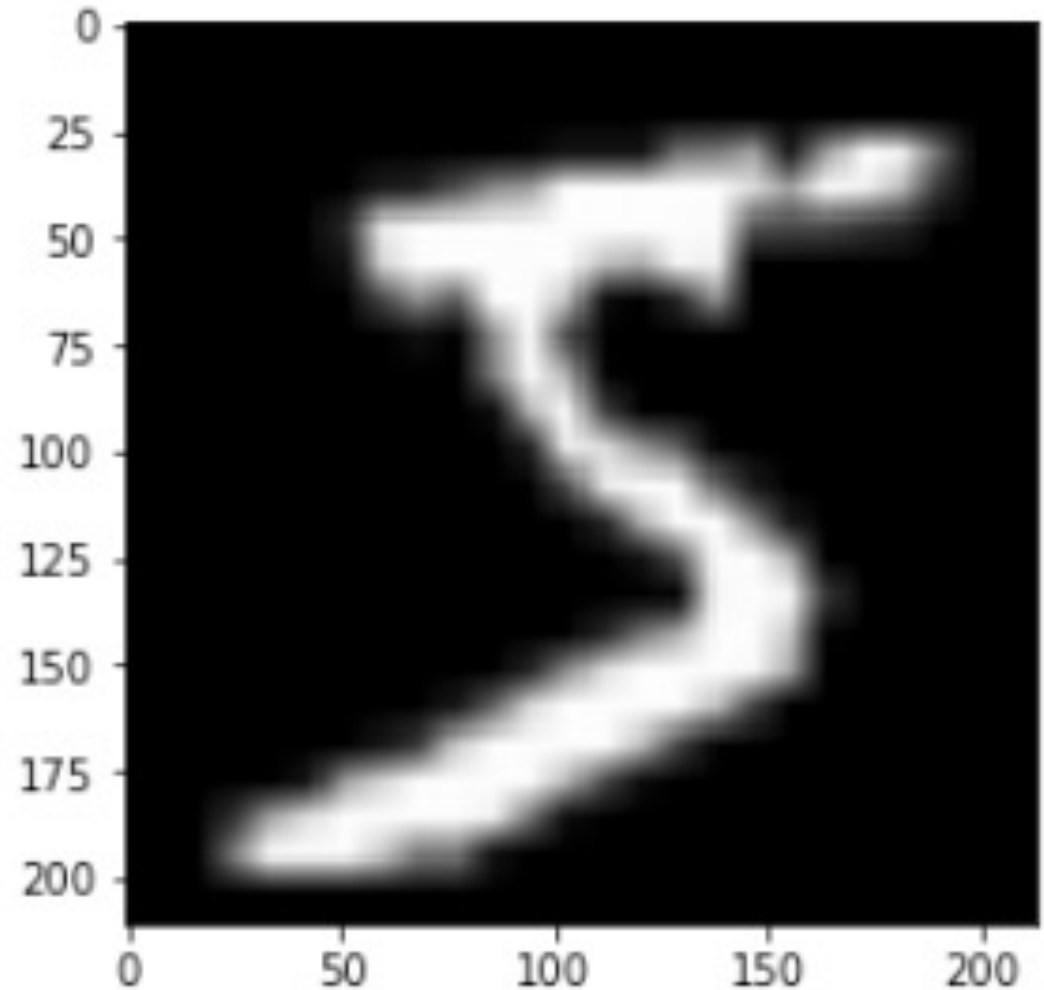
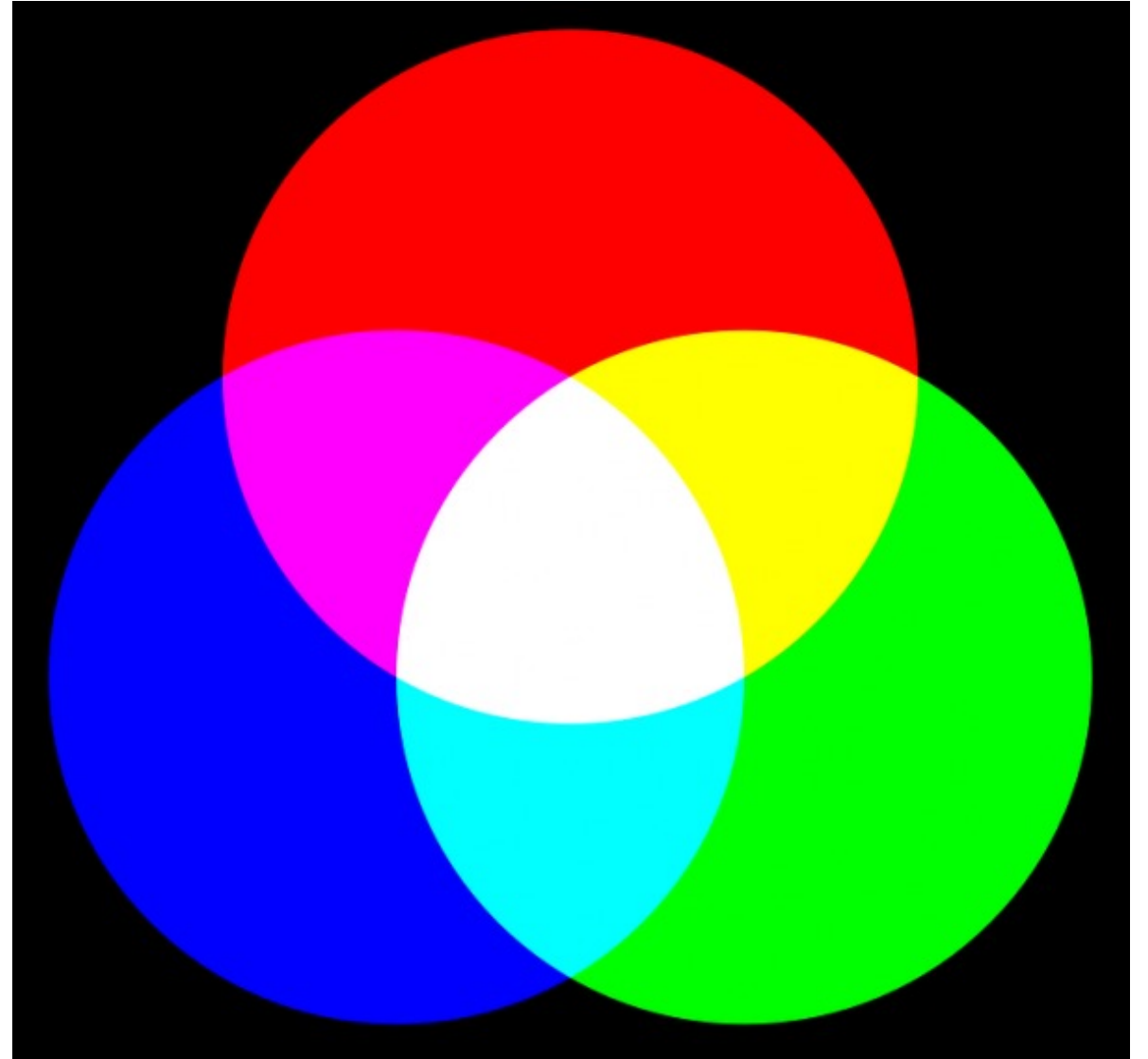


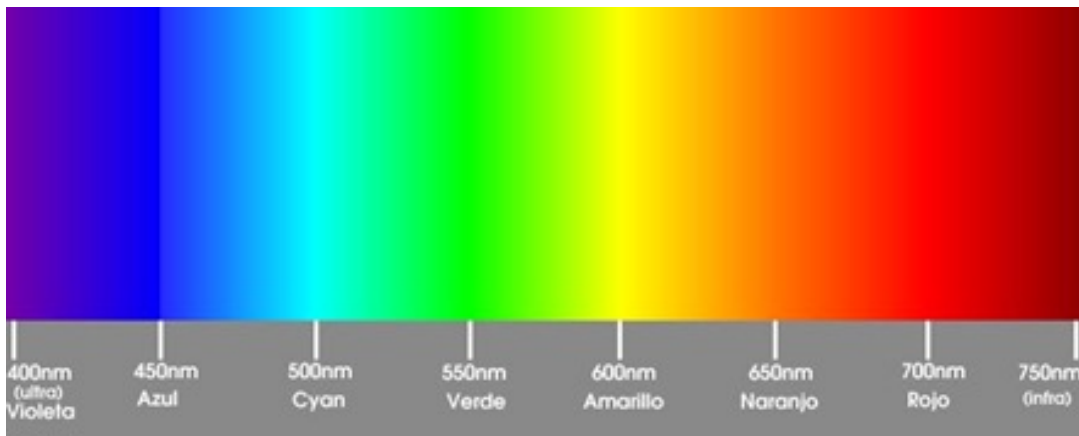


Imagen en color

Las imágenes en color se representan como una combinación lineal de **rojo** (λ onda larga), **verde** (λ onda media) y **azul** (λ onda corta), y todos los demás colores se pueden lograr mezclando estos colores primarios en las proporciones correctas.

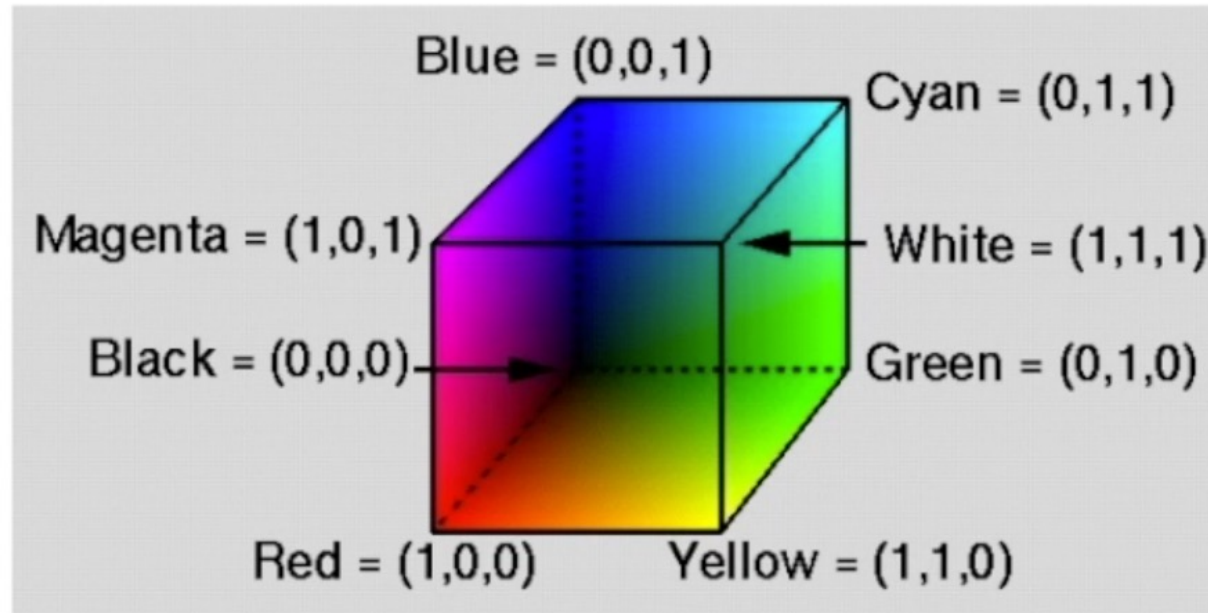


Imágenes a Color

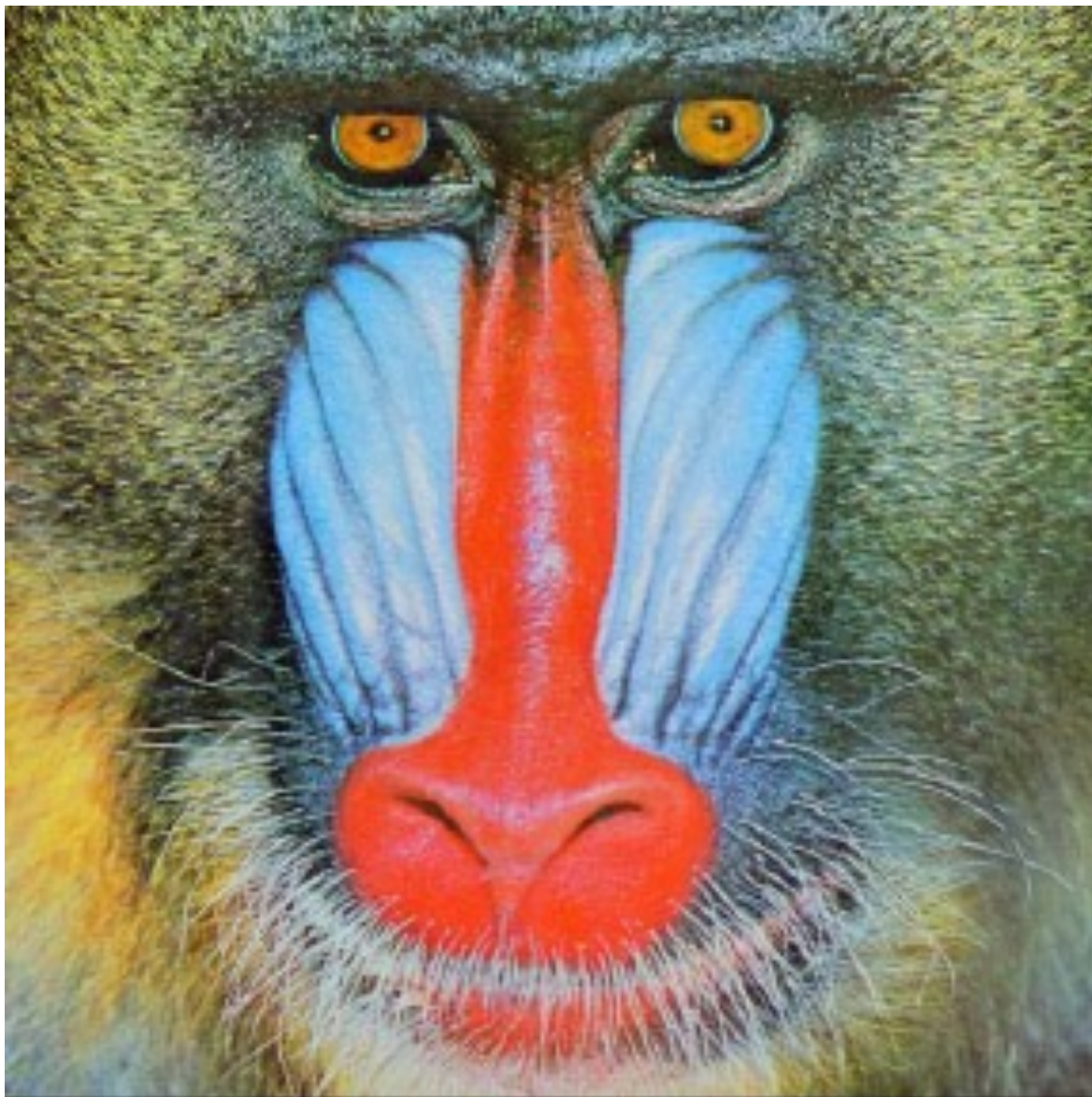


- Una imagen en color también consta de 8 bits por píxel. Como resultado, se pueden representar 256 tonos de colores diferentes con 0 que denota negro y 255 blanco.
- Cada píxel almacena un color. Los colores pueden representarse en diferentes espacio de color.
- Espacio RGB
- Como vemos que tenemos tres colores primarios entonces vamos a tener tres matrices superpuestas que corresponden a cada color, o bien se le llama tres dimensiones.
- Representamos la suma del espectro visible.

- Por ejemplo tenemos el cubo RGB donde podemos representar los niveles de color como un conjunto de tuplas $(0,0,0)$ para negro hasta $(1,1,1)$ para el blanco



- Por tal motivo, el valor de un pixel se representa como $v=(v_r, v_g, v_b)$



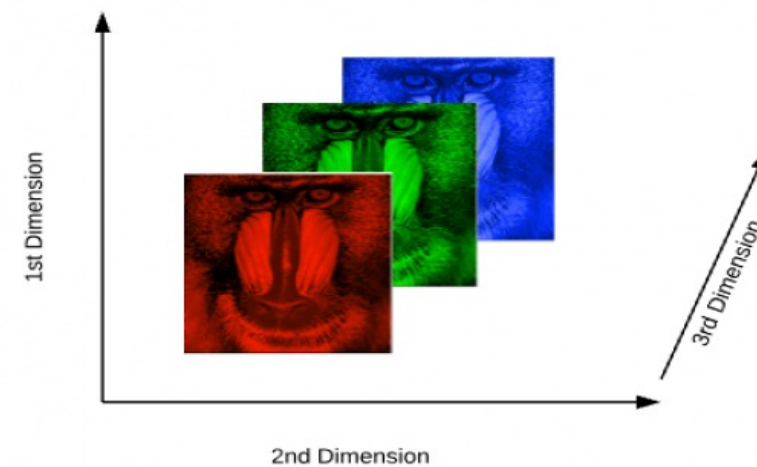
Shape

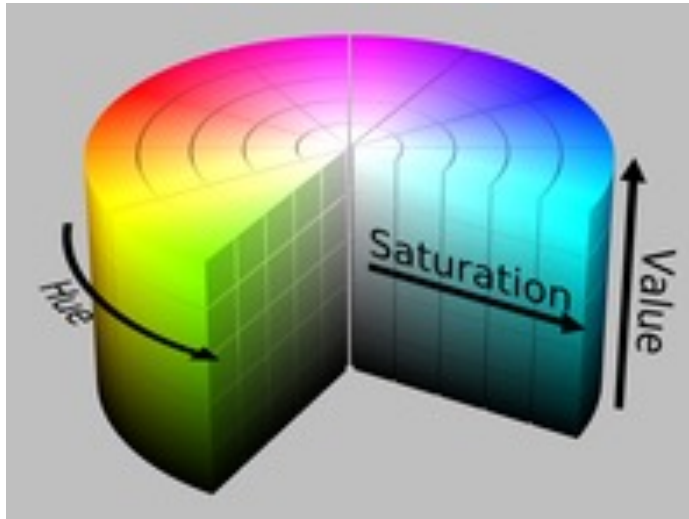
$(288, 288, 3)$

288: Pixel width

288: Pixel height

3: color channel





Existen de igual manera otros espacios de color. Por ejemplo el ***Hue Saturation Intensity***. Cada uno es utilizado según el tipo de problema que se nos presenta.

- Tono (Hue): Es el color de acuerdo a la longitud de onda. Que puede ser verdes, azul, amarillo, azul, rojo.
- Saturación: es la cantidad presente del color. Así nos permite distinguir entre un rojo y un rosado.
- Intensidad: es la cantidad de luz, por ejemplo distinguir entre un rojo oscuro y un claro.

Conversión entre espacios

$$I = \frac{1}{3}(R + G + B)$$

$$S = 1 - \frac{3}{R + G + B}[\min(R, G, B)]$$

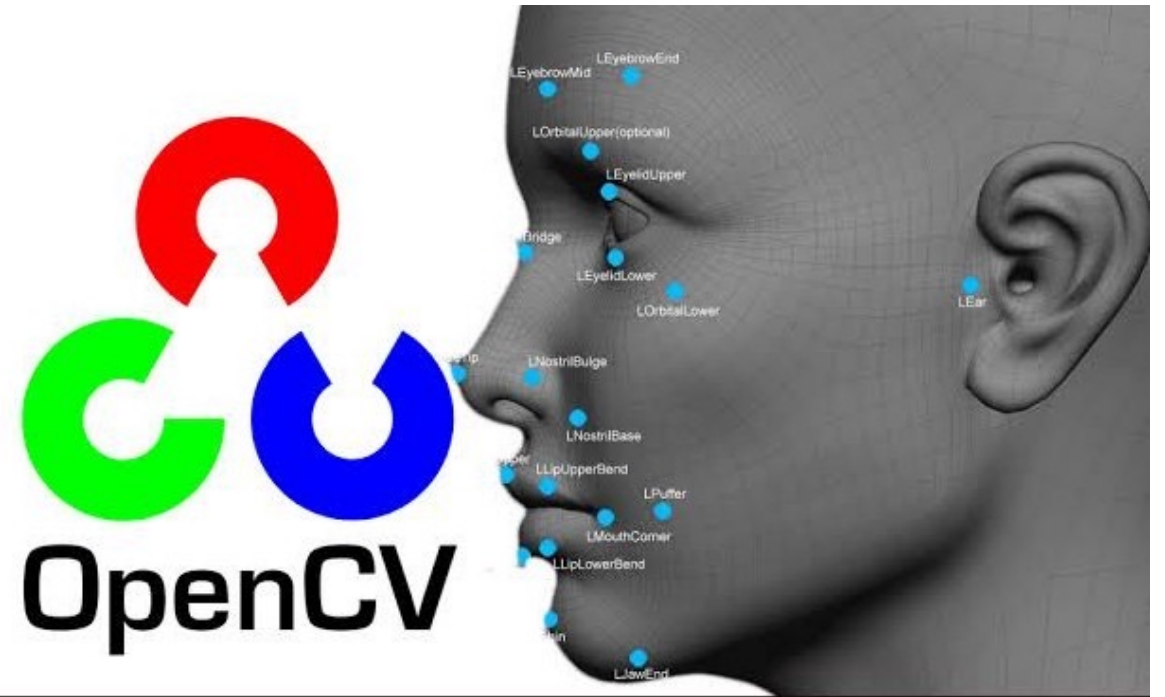
$$H = \cos^{-1} \left(\frac{1/2[(R - G) + (R - B)]}{[(R - G)^2 + (R - B)(G - B)]^{1/2}} \right)$$

Imágenes y OpenCV

Se puede utilizar OpenCV para realizar **operaciones simples con imágenes** como:

- Abrir y guardar imágenes
- Dibujar formas simples en imágenes
- Escribir en imágenes

Estas son operaciones básicas necesarias para crear una **base** antes de avanzar y poder utilizar todas las **funciones avanzadas** que OpenCV ofrece.



Conclusiones

- Las imágenes en escala de grises son útiles puesto que es más fácil trabajar con ellas porque solo tiene una dimensión, y la capacidad de procesamiento es menor que la de una se tres dimensiones.
- En el caso de las imágenes binarias es útil trabajar con ellas porque podemos resaltar objetos que son de nuestro interés.
- Las posibilidades de análisis y tratamiento de imágenes con la biblioteca OpenCV son inmensas, desde detectar caras y clasificarlas según género hasta crear modelos de realidad aumentada o usar clasificadores para detectar objetos.
- En la página oficial de OpenCV (<https://opencv.org/>) encontrarás documentación, tutoriales, foros y multitud de funcionalidades más de gran utilidad.
- El movimiento se demuestra andando y para aprender todas las posibilidades de esta gran biblioteca solo existe un camino, practicar y practicar “picando” código.



Lecturas

- Página oficial de OpenCV. Disponible en: <https://opencv.org/>
- ¿Qué es OpenCV? Instalación en Python y ejemplos básicos. Disponible en: <https://revistadigital.inesem.es/informatica-y-tics/opencv/>
- Finding Lena, the Patron Saint of JPEGs. Disponible en: <https://www.wired.com/story/finding-lena-the-patron-saint-of-jpegs/>