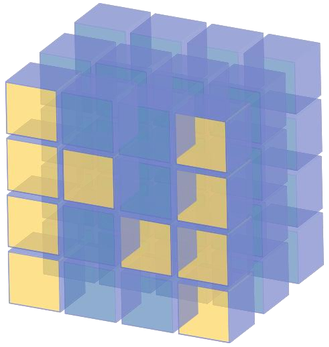
PROCESAMIENTO DIGITAL DE IMAGENES

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

Módulos principales



NumPy

matplotlib

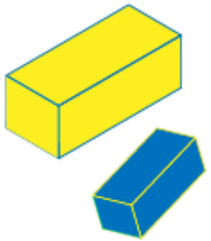
OpenCV

pip install opencv-python

PROCESAMIENTO DIGITAL DE IMÁGENES

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

GENERACIÓN DE COLORES



Los colores pueden crearse a través de la combinación de colores primarios, las reglas de combinación son:

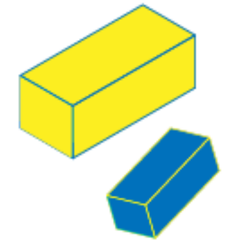
- Se puede usar colores primarios o algún color obtenido a partir de estos
- Se puede combinar más de 2 colores
- Se puede repetir un color previamente usado
- No importa el orden de combinación

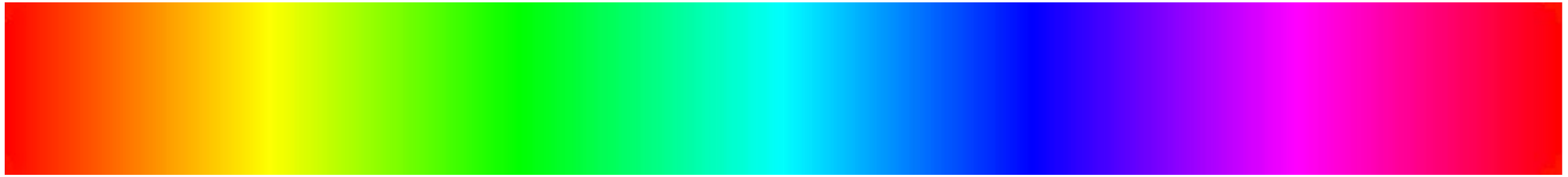
Por definición, un color primario no puede ser obtenido mediante la combinación de los otros colores primarios. Por defecto, los colores primarios que se usan suelen ser:

- Rojo, verde y azul
- Cian, magenta y amarillo.

PROCESAMIENTO DIGITAL DE IMÁGENES

GENERACIÓN DE COLORES





Rojo Verde Azul Rojo

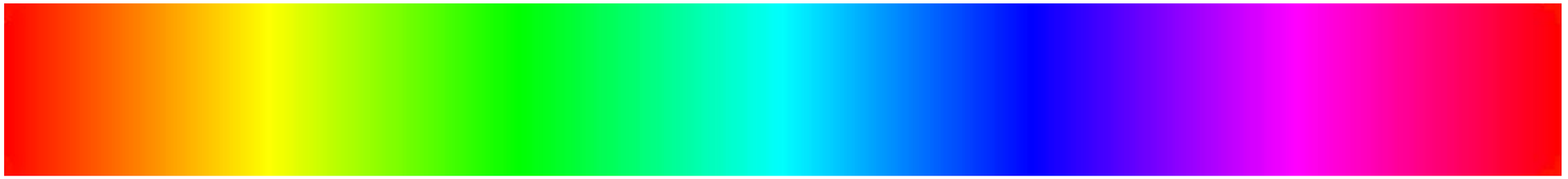
Cian Magenta Amarillo Cian

PROCESAMIENTO DIGITAL DE IMÁGENES

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

GENERACIÓN DE COLORES USANDO ROJO, VERDE Y AZUL

Rojo Verde Amarillo

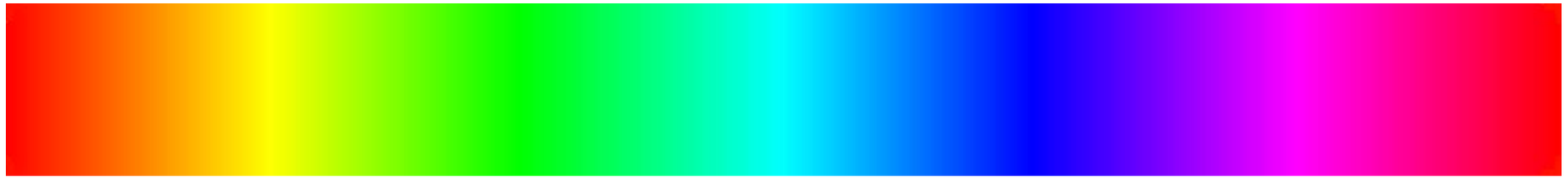


PROCESAMIENTO DIGITAL DE IMÁGENES

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

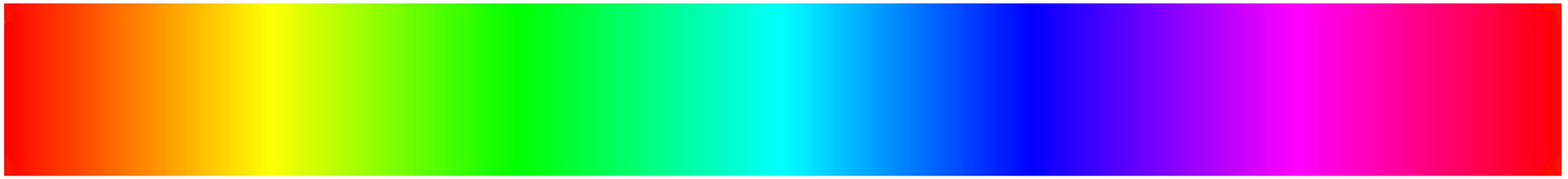
GENERACIÓN DE COLORES USANDO ROJO, VERDE Y AZUL

Verde + Azul = Cian



GENERACIÓN DE COLORES USANDO ROJO, VERDE Y AZUL

Azul + Rojo = Magenta



PROCESAMIENTO DIGITAL DE IMÁGENES

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

GENERACIÓN DE COLORES USANDO CIAN, MAGENTA Y AMARILLO

Cian + Magenta = Azul

PROCESAMIENTO DIGITAL DE IMÁGENES

GENERACIÓN DE COLORES USANDO CIAN, MAGENTA Y AMARILLO

Magenta + Amarillo = Rojo

PROCESAMIENTO DIGITAL DE IMÁGENES

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

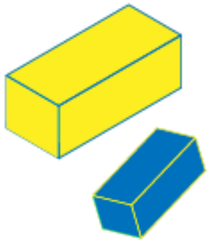
GENERACIÓN DE COLORES USANDO CIAN, MAGENTA Y AMARILLO

Amarillo + Cian = Verde

PROCESAMIENTO DIGITAL DE IMÁGENES

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

MODELOS DE COLOR

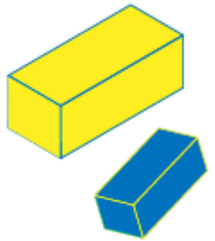


FORMATO	SIGNIFICADO	TIPO	DEFINICION
RGB	Rojo, verde, azul (Red, green, blue)	Aditivo	Al usar fuentes de luz partimos desde la oscuridad (color negro), a partir del cual agregamos (adicionamos) luces de colores. Si combinamos los 3 colores obtenemos el color blanco.

PROCESAMIENTO DIGITAL DE IMÁGENES

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

MODELOS DE COLOR

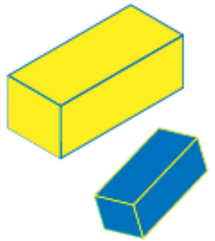


FORMATO	SIGNIFICADO	TIPO	DEFINICION
CMY	Cian, magenta, amarillo (Cyan, magenta, yellow)	Sustractivo	Estos colores se forman como resultado del reflejo que da una superficie al absorber luz. Si combinamos los 3 colores obtenemos el color negro.

PROCESAMIENTO DIGITAL DE IMÁGENES

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

MODELOS DE COLOR



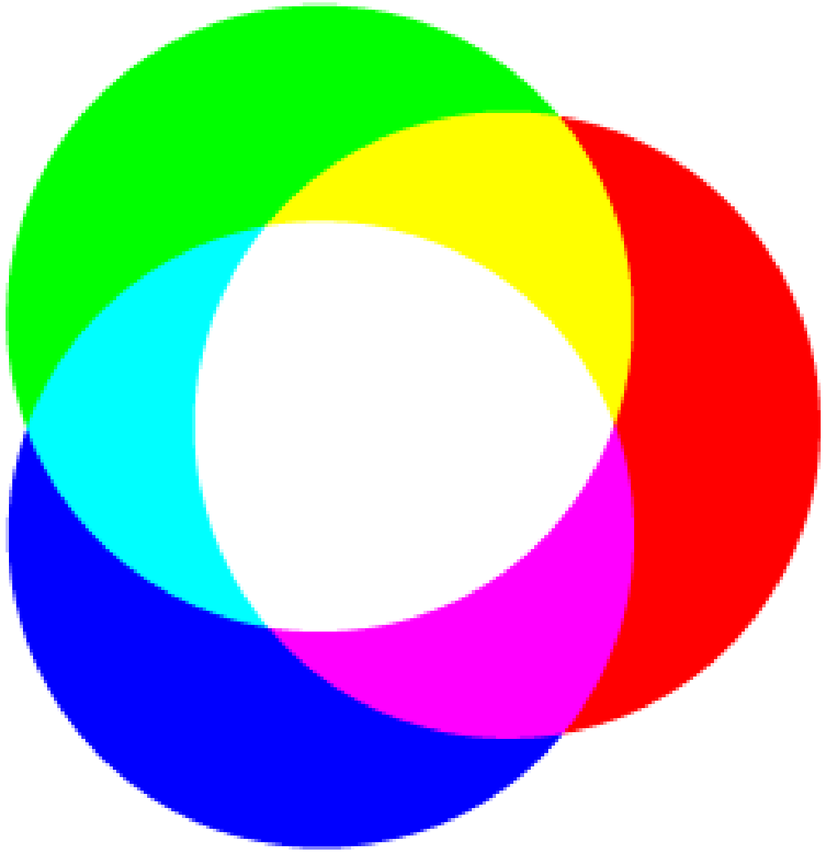
FORMATO	SIGNIFICADO	TIPO	DEFINICION
CMY + K	Cian, magenta, amarillo (Cyan, magenta, yellow) + Negro (K)	Sustractivo	Al momento de querer usar el formato CMY, no se suele tener la suficiente pureza en estos 3 colores; esto resulta en un color marrón oscuro si los combinamos, en lugar del color negro. Por ello, se agrega el color negro como un extra a estos 3 colores; sin embargo, el negro no es un color primario.

PROCESAMIENTO DIGITAL DE IMÁGENES

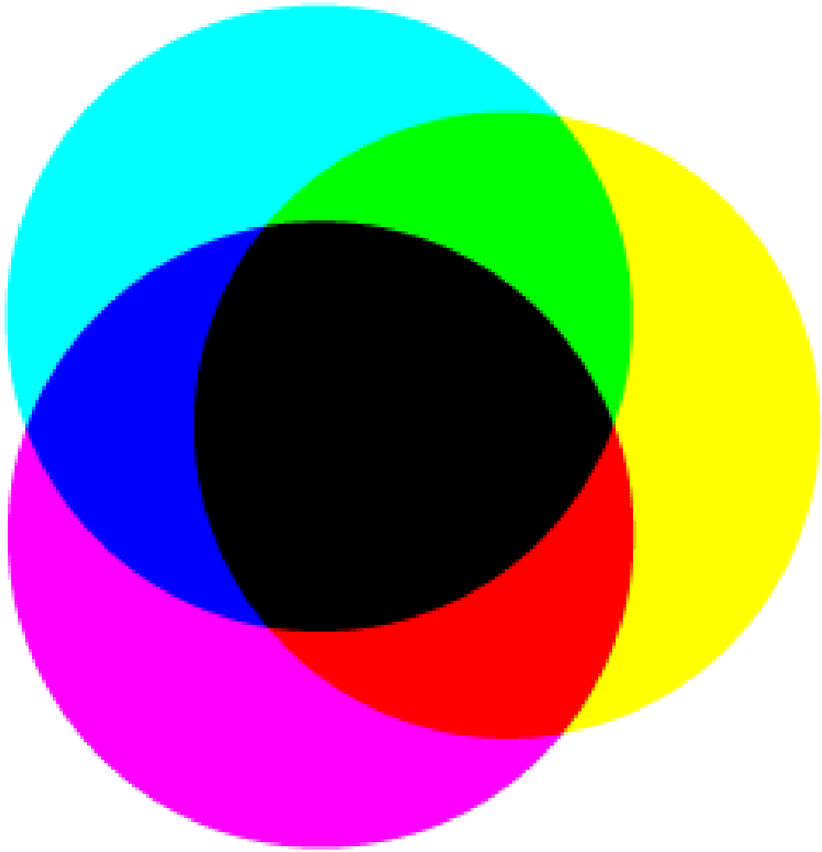
UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

MODELOS DE COLOR

Rojo, verde y azul



Cian, magenta y amarillo

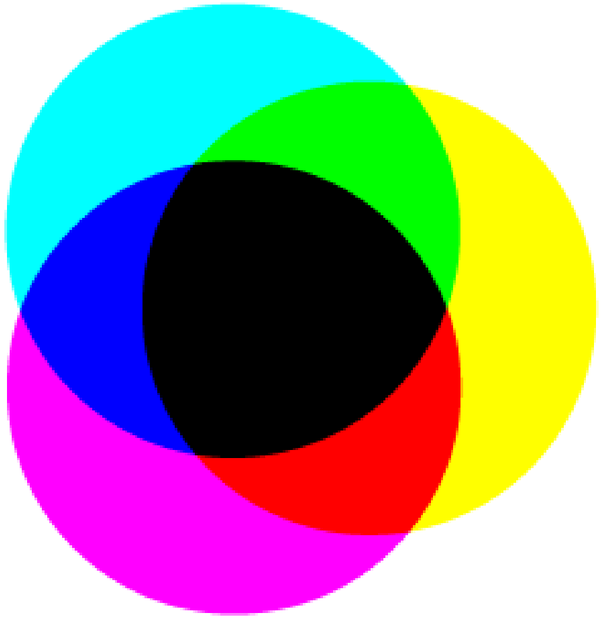


PROCESAMIENTO DIGITAL DE IMÁGENES

MODELOS DE COLOR

ACERCA DEL FORMATO CMY + K

- Se usa la letra K para el color negro en lugar de la B porque se refiere a «key plate»
- El término «key plate» se usaba en la impresión, significa placa maestra, y era responsable de imprimir el detalle artístico de una imagen, normalmente usando tinta negra
- El color cian es un filtro que absorbe el color rojo
- El color magenta es un filtro que absorbe el color verde
- El color amarillo es un filtro que absorbe el color azul



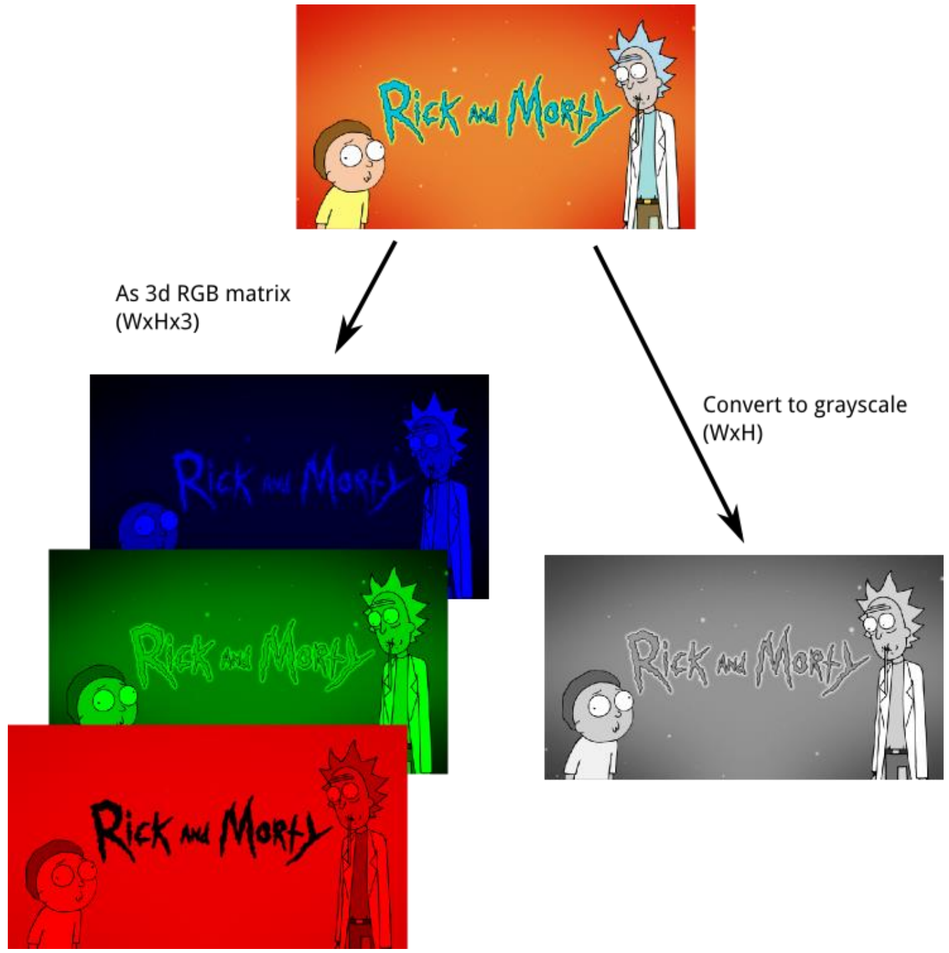
PROCESAMIENTO DIGITAL DE IMÁGENES

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

ESPACIOS DE COLOR

Un espacio de color es un sistema a través del cual se interpretan los colores en una imagen o video. El espacio de color depende del modelo de color usado, siendo que el modelo de color define los colores primarios a usar, mientras que el espacio de color define el orden de los colores. Por ejemplo, podemos usar el modelo RGB para formar los espacios RGB y BGR.

Trabajaremos con 2 espacios: RGB y GRAY

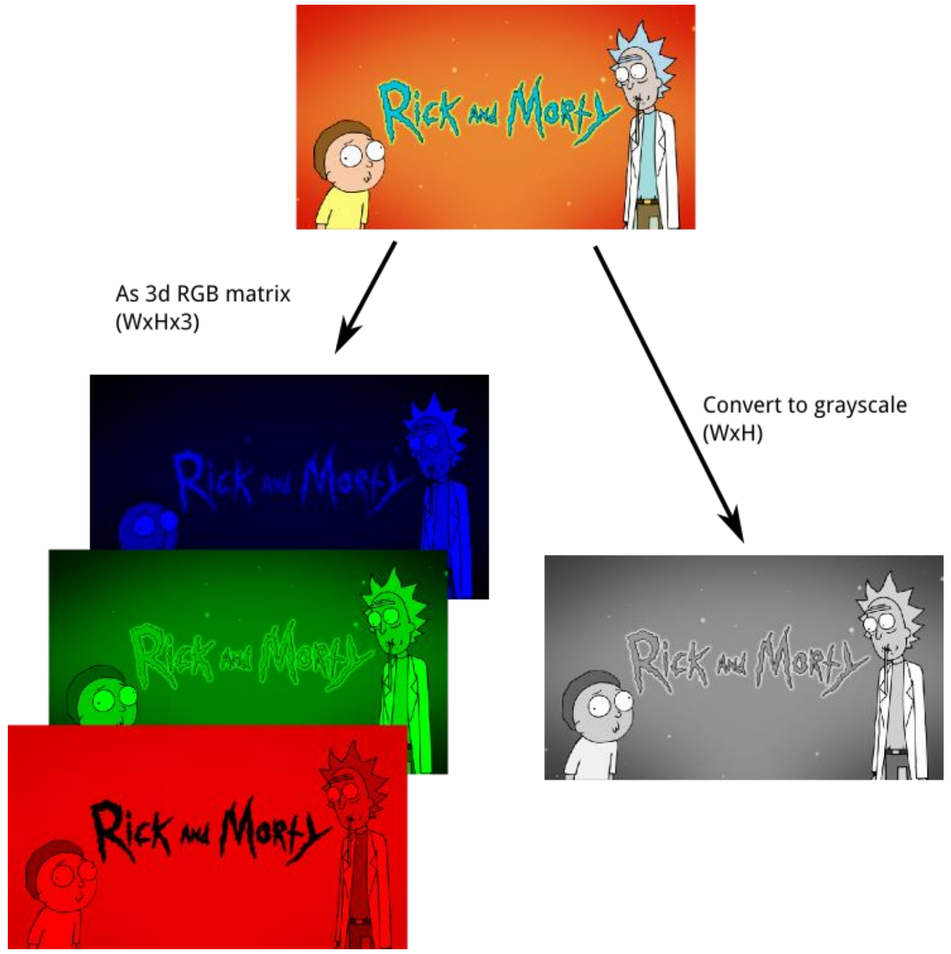


PROCESAMIENTO DIGITAL DE IMÁGENES

ESPACIOS DE COLOR

- Cada uno de los colores usados en el espacio de color es conocido como canal; siendo que, estos canales se aplican por cada píxel en una imagen
- Se estila tener 256 valores diferentes en cada canal
- Los valores de cada canal pueden ser representados con valores enteros o decimales

Trabajaremos con 2 espacios: RGB y GRAY



PROCESAMIENTO DIGITAL DE IMÁGENES

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

Ejemplo:

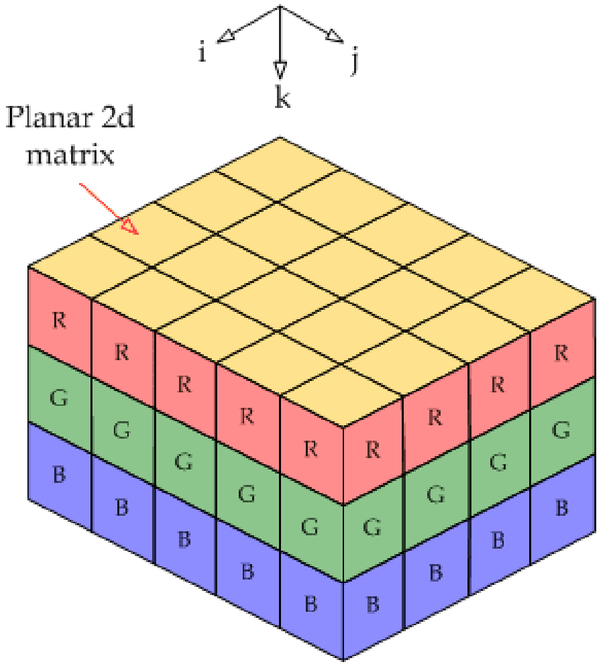
```python
import cv2

image = cv2.imread("random_scene.jpg")
cv2.imshow("Imagen", image)

while True:
    key = cv2.waitKey(0) & 0xFF

    if key == ord("c"):
        break

cv2.destroyAllWindows()
```

Ejemplo:

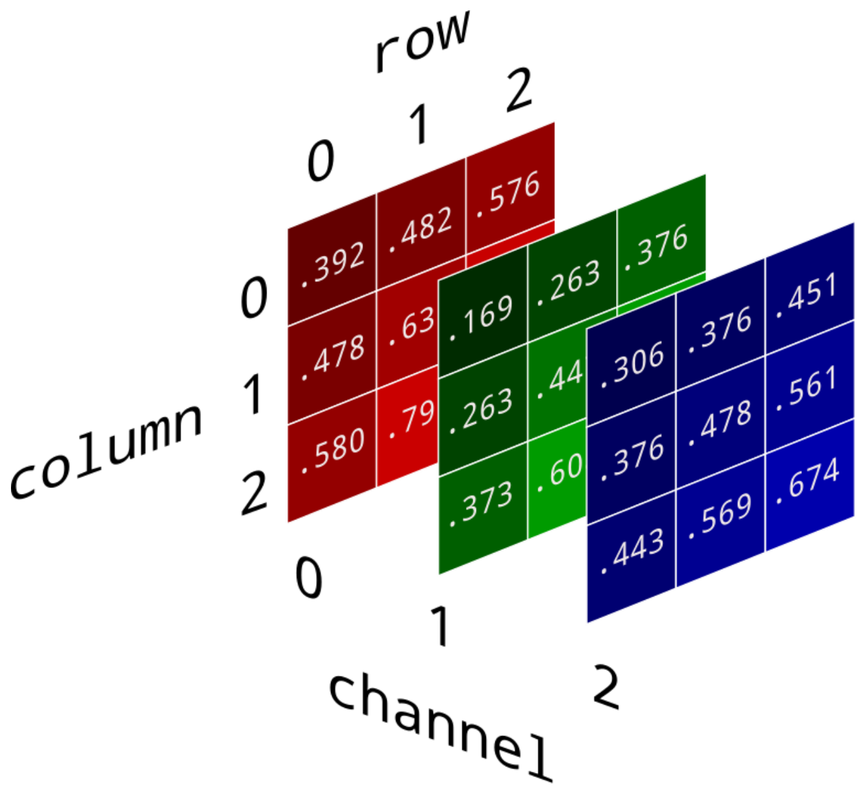
```python
import cv2

image_as_BGR = cv2.imread("random_scene.jpg")

image_as_BGR_copy = image_as_BGR.copy()
image_as_BGR_copy[:, :, 1] = 0 #G
image_as_BGR_copy[:, :, 2] = 0 #R

cv2.imshow("Imagen con solo el canal azul", image_as_BGR_copy)

image_as_BGR_copy = image_as_BGR.copy()
image_as_BGR_copy[:, :, 0] = 0 #B
image_as_BGR_copy[:, :, 2] = 0 #R

cv2.imshow("Imagen con solo el canal verde", image_as_BGR_copy)

image_as_BGR_copy = image_as_BGR.copy()
image_as_BGR_copy[:, :, 0] = 0 #B
image_as_BGR_copy[:, :, 1] = 0 #G

cv2.imshow("Imagen con solo el canal rojo", image_as_BGR_copy)

while True:
    key = cv2.waitKey(0) & 0xFF

    if key == ord("c"):
        break

cv2.destroyAllWindows()
```

ESPACIOS DE COLOR (RGB)

Cada uno de los píxeles que posee una imagen se divide en cada uno de los canales del espacio de color. De esta forma, si una imagen es del tamaño 20x20, entonces se tendrá 400 píxeles, donde cada uno de ellos tendrá 3 valores referidos a cada canal.



Graphical presentation of RGB 3d matrix

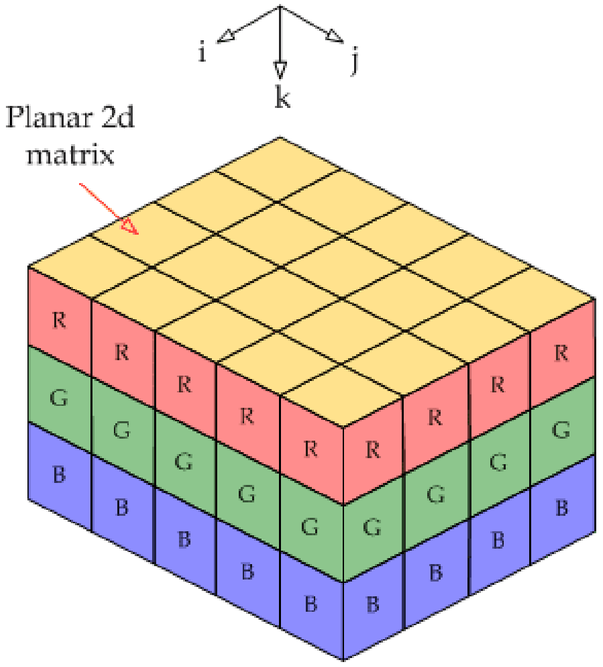


PROCESAMIENTO DIGITAL DE IMÁGENES

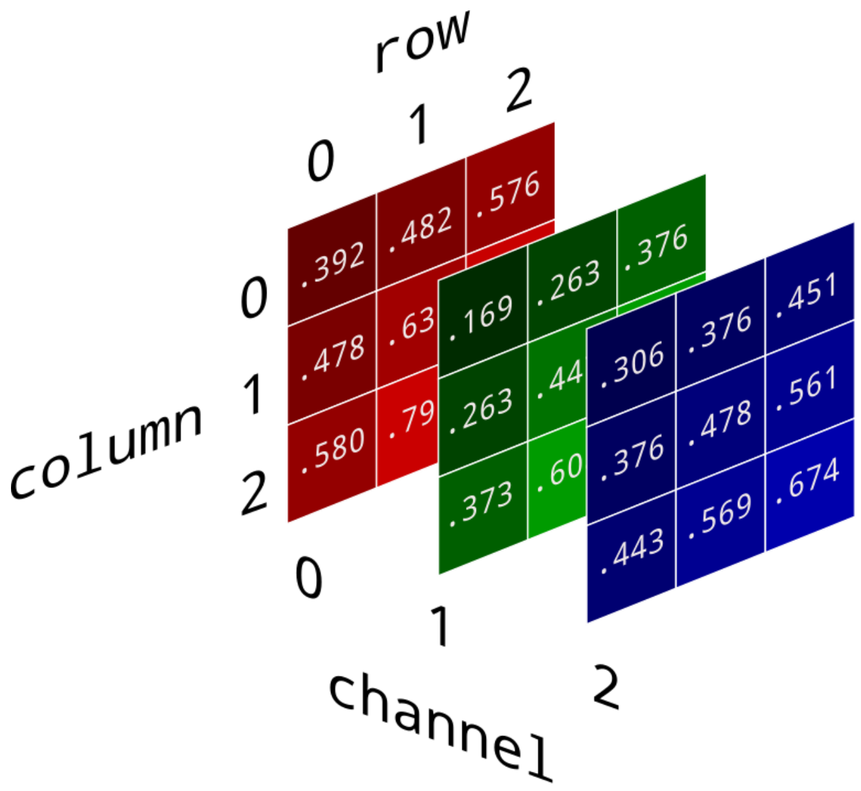
UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

ESPACIOS DE COLOR (RGB)

- El color blanco se obtiene usando el máximo valor que puede tener cada uno de los canales
- El color negro se obtiene usando el mínimo valor que puede tener cada uno de los canales
- El color de cada píxel se obtiene combinando los valores de sus canales



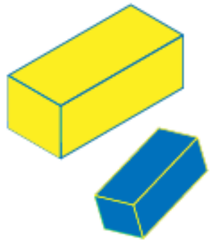
Graphical presentation of RGB 3d matrix



PROCESAMIENTO DIGITAL DE IMÁGENES

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

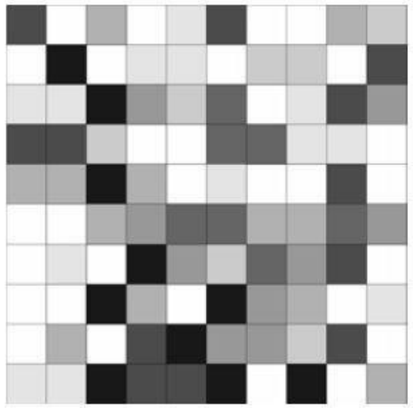
Ejemplo:



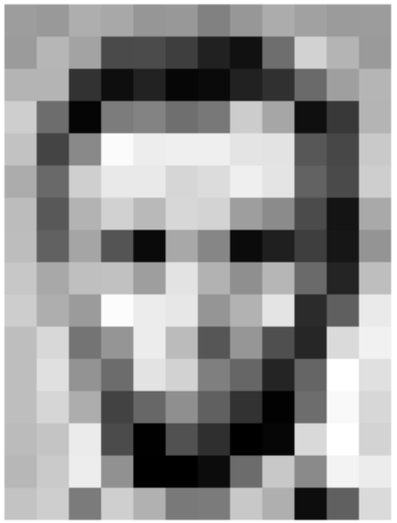
```python
import cv2
import matplotlib.pyplot as plt

image_as_BGR = cv2.imread("random_scene.jpg")
image_as_RGB = cv2.cvtColor(image_as_BGR, cv2.COLOR_BGR2RGB)

plt.title("Imagen como RGB")
plt.imshow(image_as_RGB)
plt.xticks([])
plt.yticks([])
plt.show()
```

ESPACIOS DE COLOR (GRAY)

En este caso solo se cuenta con un canal, siendo que se usa la escala de grises a través de este canal. Al igual que con el espacio de color RGB, el mínimo valor representa el color negro y el máximo valor representa el color blanco.



254	107
255	165



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

PROCESAMIENTO DIGITAL DE IMÁGENES

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

Ejemplo:

```python
import cv2
import matplotlib.pyplot as plt

image_as_BGR = cv2.imread("random_scene.jpg")
image_as_GRAY = cv2.cvtColor(image_as_BGR, cv2.COLOR_BGR2GRAY)

plt.title("Imagen como GRAY")
plt.imshow(image_as_GRAY, cmap=plt.get_cmap("gray"))
plt.xticks([])
plt.yticks([])
plt.show()

cv2.imshow("Imagen como GRAY", image_as_GRAY)

while True:
    key = cv2.waitKey(0) & 0xFF

    if key == ord("c"):
        break

cv2.destroyAllWindows()
```

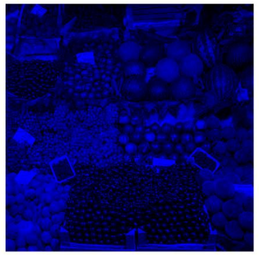
CAMBIO DE ESPACIOS DE COLOR

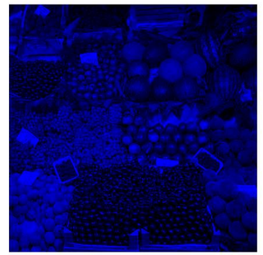
Cambiar entre espacios de color que usan los mismos modelos de color solo requiere cambiar el orden de los colores. Por ejemplo, para cambiar de RGB a BGR basta con cambiar de lugares los canales rojo y azul.

RGB 

R 

G 

B 

 B

 G

 R

PROCESAMIENTO DIGITAL DE IMÁGENES

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

Ejemplo:

```python
import cv2
import matplotlib.pyplot as plt

image_as_BGR = cv2.imread("random_scene.jpg")
image_as_RGB = cv2.cvtColor(image_as_BGR, cv2.COLOR_BGR2RGB)
image_as_BGR = cv2.cvtColor(image_as_RGB, cv2.COLOR_RGB2BGR)

plt.title("Imagen como RGB")
plt.imshow(image_as_RGB)
plt.xticks([])
plt.yticks([])
plt.show()

plt.title("Imagen como BGR")
plt.imshow(image_as_BGR)
plt.xticks([])
plt.yticks([])
plt.show()
```

CAMBIO DE ESPACIOS DE COLOR

Cambiar entre el espacio de color RGB a GRAY implica una reducción de información, esto es debido que con RGB tenemos 3 canales mientras que con GRAY se tiene 1 canal. Para realizar este cambio de espacios se pondera cada uno de los canales mediante la siguiente fórmula:

$$a*R + b*G + c*B, \text{ tal que } a + b + c = 1$$

Por ejemplo: $0.2126*R + 0.7152*G + 0.0722*B$

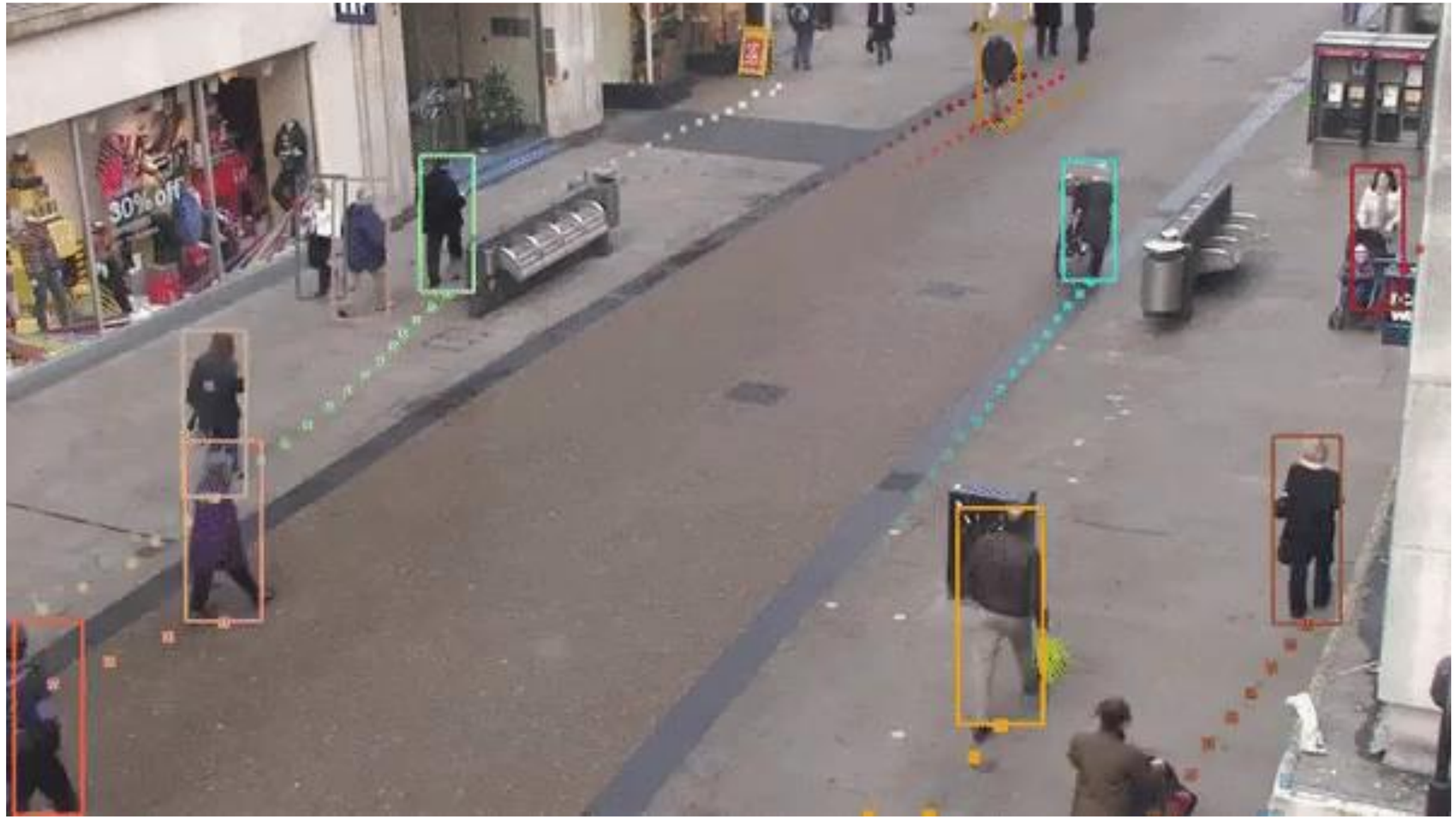
PROCESAMIENTO DIGITAL DE IMÁGENES

Ejemplo:

```python
import cv2
import matplotlib.pyplot as plt

image_as_BGR = cv2.imread("random_scene.jpg")
image_as_RGB = cv2.cvtColor(image_as_BGR, cv2.COLOR_BGR2RGB)
image_as_GRAY = cv2.cvtColor(image_as_RGB, cv2.COLOR_RGB2GRAY)

plt.title("Imagen como RGB")
plt.imshow(image_as_RGB)
plt.xticks([])
plt.yticks([])
plt.show()

plt.title("Imagen como GRAY")
plt.imshow(image_as_GRAY, cmap=plt.get_cmap("gray"))
plt.xticks([])
plt.yticks([])
plt.show()
```

SEGUIMIENTO DE OBJETOS

Consiste en localizar objetos en los fotogramas de un video, independientemente si es que estos se encuentran o no en movimiento.



PROCESAMIENTO DIGITAL DE IMÁGENES

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

SEGUIMIENTO DE OBJETOS

Para objetos con colores monocromáticos, el seguimiento de objetos se consigue mediante la detección de los colores.

PROCESAMIENTO DIGITAL DE IMÁGENES

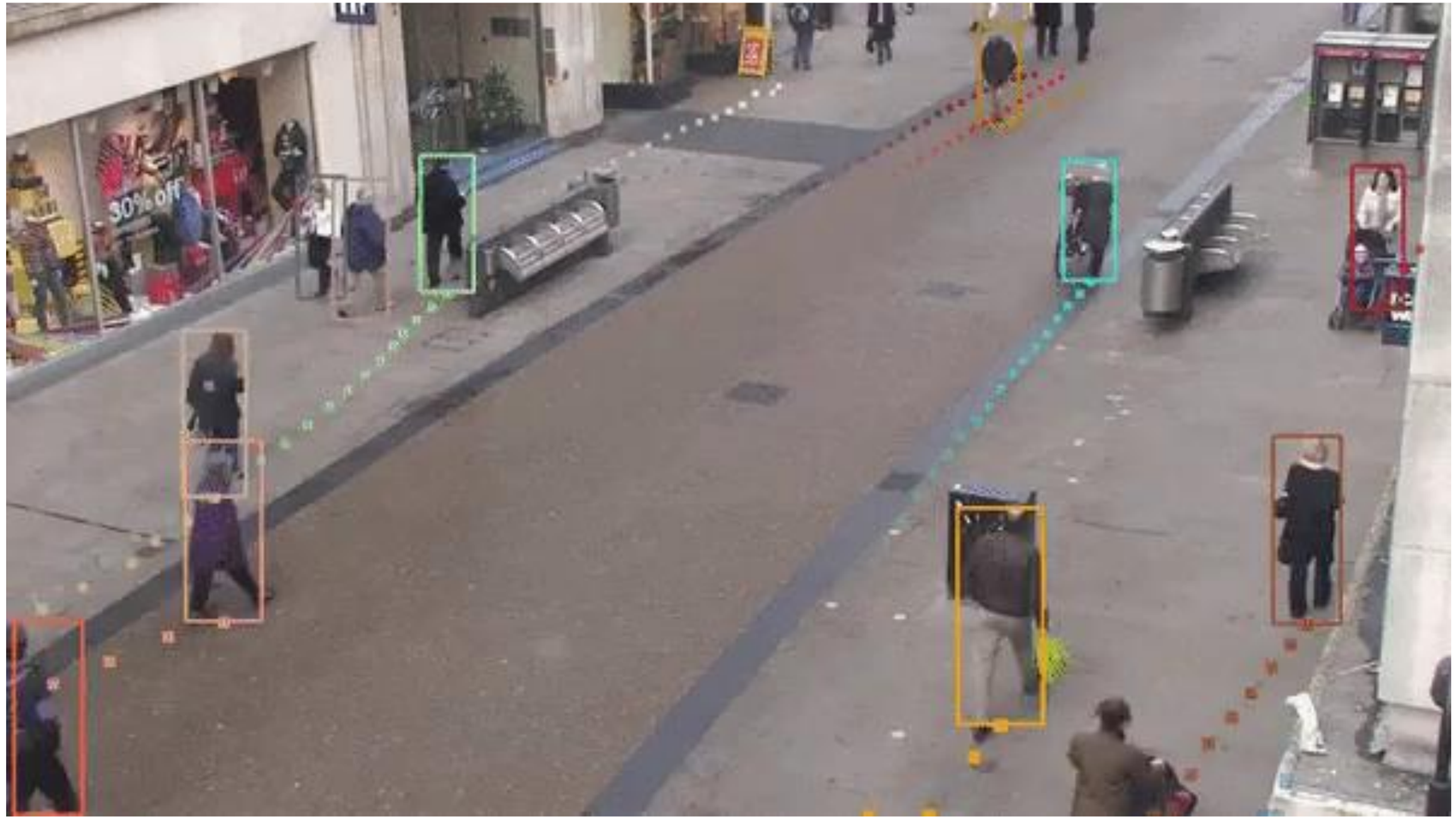
UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

Ejemplo:

```python
import cv2
import numpy as np

capture = cv2.VideoCapture("./circles.avi")

while capture.isOpened():
    return_value, frame = capture.read()

    if return_value == True:
        frame_as_HSV = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

        blue_mask = cv2.inRange(frame_as_HSV, (100, 100, 20), (125, 255, 255))

        blue_circles = cv2.bitwise_and(frame, frame, mask=blue_mask)

        blue_mask = cv2.bitwise_not(blue_mask)

        background = np.full(frame.shape, (250, 255, 0), np.uint8)
        background = cv2.bitwise_and(background, background, mask=blue_mask)

        frame = cv2.bitwise_or(blue_circles, background)

        cv2.imshow("Foreground", blue_circles)
        cv2.imshow("Background", background)
        cv2.imshow("Video", frame)

        key = cv2.waitKey(24) & 0xFF

        if key == ord("c"):
            break
    else:
        break

capture.release()

cv2.destroyAllWindows()
```

SEGUIMIENTO DE OBJETOS

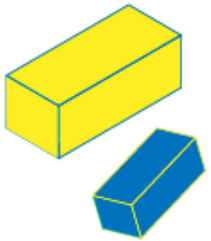
Consiste en localizar objetos en los fotogramas de un video, independientemente si es que estos se encuentran o no en movimiento.



PROCESAMIENTO DIGITAL DE IMÁGENES

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

SEGUIMIENTO DE OBJETOS

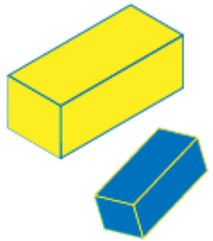


Una forma de realizar el seguimiento de un objeto es a través de su detección en cada fotograma del video. Sin embargo, esto presenta las siguientes desventajas:

Desventajas:

- Si el objeto es ocultado, entonces este método fallará puesto que no detectará donde se encuentra el objeto
- Detectar el objeto en cada fotograma es lento debido al procesamiento requerido
- No se aprovecha la información de los fotogramas previos, a través de los cuales se puede obtener la apariencia, posición, tamaño, región visible, dirección y velocidad de movimiento del objeto
- En caso el objeto desaparezca repentinamente y aparezca uno similar en otra región del fotograma, entonces el objeto similar será detectado independientemente de si es o no el objeto original puesto que no se tiene información previa

SEGUIMIENTO DE OBJETOS



Las desventajas mencionadas se resuelven detectando el objeto en el primer fotograma, para que a partir de ahí se proceda a seguirlo. De este modo, la información previa puede usarse para predecir la posición futura del objeto en el siguiente fotograma y así realizar su búsqueda en una región reducida.

Nuevas Desventajas:

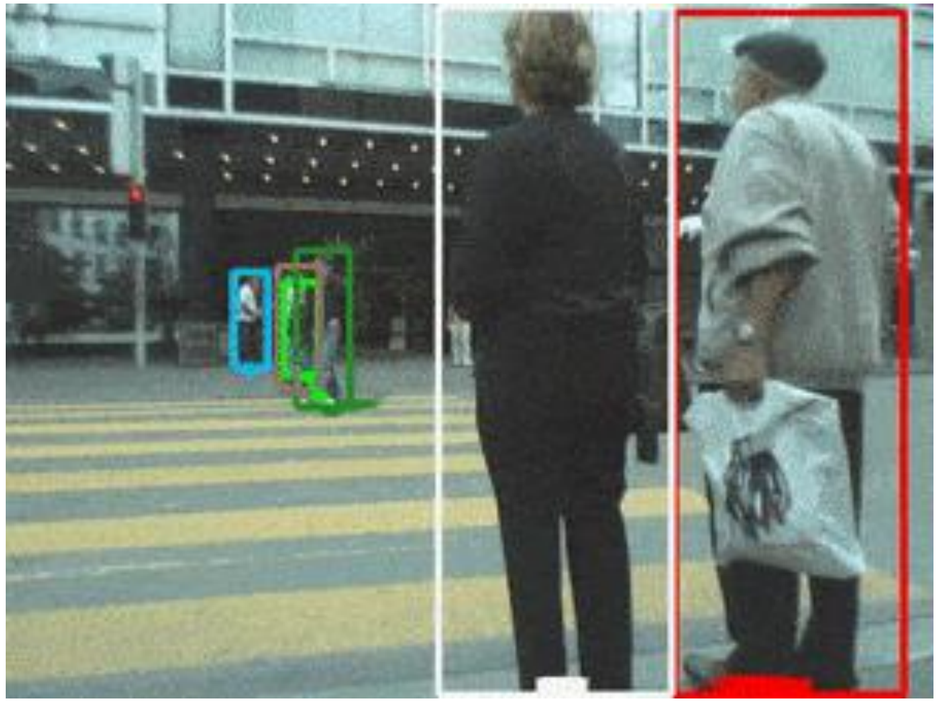
- Se puede perder la ubicación del objeto si este permanece oculto durante varios fotogramas consecutivos
- Si el objeto cambia repentinamente su velocidad y tamaño, la información previa no es suficiente para continuar siguiendo el objeto
- Con el paso de los fotogramas, el cuadro delimitador que rastrea el objeto se aleja lentamente, aparece a un costado o solo delimita una porción del objeto en cuestión debido a las 2 desventajas previas

SEGUIMIENTO DE OBJETOS

PROCESAMIENTO DIGITAL DE IMÁGENES

SEGUIMIENTO DE OBJETOS

Las nuevas desventajas se resuelven detectando el objeto de vez en cuando a través de los fotogramas, de tal forma que se pueda actualizar el cuadro delimitador del objeto.



PROCESAMIENTO DIGITAL DE IMÁGENES

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO

UMAKER | CENTRO DE CAPACITACIÓN DE DESARROLLO TECNOLÓGICO