

## Hoja de Trabajo 1

### Task 1:

1. Como director de un proyecto de conducción autónoma, debe dimensionar el hardware para un nuevo vehículo. El sistema utiliza 8 cámaras que capturan video a resolución 4K UHD (3840 x 2160). Debido a la necesidad de alto rango dinámico (HDR), los sensores operan a 12 bits por píxel (Raw Bayer Pattern) a 60 FPS. Métrica A: Enfocada puramente en el flujo vehicular.

**a. Calcule el tamaño exacto de una sola imagen (frame) cruda en Megabytes (MB).**

Resolución:  $3840 \times 2160 = 8,294,400$

Tamaño (bits):  $8,294,400 \times 12 = 99,532,800$  bits

Tamaño (MB):  $99,532,800 \div (8 \times 1,000 \times 1,000) = 12.44$  MB → **Tamaño MB**

**b. Calcule el ancho de banda necesario (en Gbps) para transmitir el flujo de las 8 cámaras al procesador central sin compresión.**

Para una cámara:  $99,532,800 \text{ bits/frame} \times 60 \text{ FPS} = 5,971,968,000 \text{ bits/s} \approx 5.97 \text{ Gbps}$

Como son 8 cámaras:  $5.97 \text{ Gbps} \times 8 = 47.76 \text{ Gbps}$  → **Ancho de banda**

**c. Si su procesador tiene una memoria RAM reservada de 16 GB exclusivamente para el buffer de video, ¿cuántos segundos de historia puede almacenar antes de empezar a sobrescribir datos?**

Por segundo:  $12.44 \text{ MB} \times 60 \text{ FPS} = 746.4 \text{ MB/s}$

Como son 8 cámaras:  $746.4 \text{ MB/s} \times 8 = 5,971.2 \text{ MB/s}$

RAM:  $16 \text{ GB} = 16,000 \text{ MB}$

Segundos almacenables:  $16,000 \text{ MB} \div 5,971.2 \text{ MB/s} = 2.68 \text{ s}$  → **Segundos de historia de video**

**d. Basado en su resultado, ¿es viable enviar estos datos "crudos" a la nube en tiempo real usando 5G? Justifique**

No es viable porque nuestro sistema de 8 cámaras requiere casi 48 Gbps de ancho de banda, y una red 5G logra llegar a 10 Gbps en condiciones óptimas (Locke, 2024). Dado esto, la red tendría una magnitud de ancho de banda por debajo de la requerida para enviar los datos crudos a la nube en tiempo real.

2. Considere un píxel con valor de intensidad  $I_{in}=50$  en una imagen estándar de 8 bits (0–255). Se aplican dos procesos de mejora secuenciales en el siguiente orden:

**I. Corrección Gamma con  $\gamma=0.5$  (para expandir sombras).**

**II. Ajuste Lineal con ganancia  $\alpha=1.2$  y brillo  $\beta=-10$  (para contrastar).**

Realice los cálculos en el dominio de flotantes normalizados [0,1] como dicta la buena práctica y convierta a entero de 8 bits solo al final.

**a. Calcule el valor final del píxel I\_out.**

$$I_{in(normalizado)} = \frac{50}{255} = 0.1961$$

$$I_1 = \left(\frac{50}{255}\right)^{0.5} = \sqrt{\frac{50}{255}}$$

$$I_2 = \alpha * I_1 + \beta = (1.2) \left(\sqrt{\frac{50}{255}}\right) + \left(-\frac{10}{255}\right) = 0.5314 - 0.0392 = 0.4922 \rightarrow \text{píxel I\_out}$$

$$I_{out} = 0.4922 * 255 = 125.5 \cong 125$$

**b. ¿Hubo saturación (clipping) en el proceso?**

No hubo clipping, porque el valor de I\_out se encuentra en el rango normalizado [0,1].

**c. Si hubiéramos realizado las operaciones usando uint8 directamente sin convertir a float (truncando decimales en cada paso intermedio), ¿cuál habría sido el error numérico resultante?**

$$I_1 = 255 * \left(\frac{50}{255}\right)^{0.5} = 112.9 \cong 112$$

$$I_2 = \alpha * I_1 + \beta = (1.2)(112) - 10 = 134.4 - 10 = 124.4 \cong 124 \rightarrow \text{píxel I\_out}$$

$$\text{Error numérico} = 125 - 124 = 1 \text{ nivel}$$

Realizando las operaciones usando uint8 sin convertir a float, el error numérico resultante sería de 1 nivel.

3. Usted está programando un robot clasificador de pelotas. Tiene dos objetos: una pelota roja brillante bajo el sol  $R_{rgb}=(255,0,0)$  y la misma pelota roja en una sombra profunda  $S_{rgb}=(50,0,0)$

**a. Calcule la distancia entre estos dos colores en el espacio RGB.**

$$d = \sqrt{(255 - 50)^2 + (0 - 0)^2 + (0 - 0)^2} = \sqrt{205^2} = 205 \rightarrow \text{distancia}$$

**b. Convierta ambos colores al espacio HSV (asuma rangos normalizados  $H \in [0,1], S \in [0,1], V \in [0,1]$  para simplificar, sabiendo que el Hue del rojo es 0).**

Pelota roja bajo el sol:

$$V = \max(R_{rgb}) \div 255 = \max(255,0,0) \div 255 = 255 \div 255 = 1$$

$$S = (V - \min(R_{rgb})) \div V = (1 - 0) \div 1 = 1$$

$$R_{HSV} = (0, 1, 1)$$

Pelota roja bajo la sombra:

$$V = \max(R_{rgb}) \div 255 = \max(50, 0, 0) \div 255 = 50 \div 255 = 0.1961$$

$$S = (V - \min(R_{rgb})) \div V = (1 - 0) \div 1 = 1$$

$$S_{HSV} = (0, 1, 0.1961)$$

**c. Calcule la diferencia absoluta canal por canal en HSV**

$$R_{HSV} - S_{HSV} = (|0 - 0|, |1 - 1|, |1 - 0.1961|) = (0, 0, 0.8039)$$

**d. Argumente matemáticamente por qué un algoritmo de agrupación (clustering) simple fallaría en RGB pero funcionaría en HSV para determinar que ambos píxeles pertenecen al mismo objeto "pelota roja".**

La iluminación escala los componentes RGB, por lo tanto, su distancia es de 205 cuando el color es el mismo. En cambio, usando HSV, el color se rige por el valor de H y S, que en este caso sigue siendo el mismo. Debido a esto, el clustering funciona en HSV porque detecta que el color es el mismo pero con una iluminación diferente, a comparación del RGB donde piensa que es un color distinto debido a su distancia.

**Task 2:**

**Ejercicio 1:**

Implemente la función `manual_contrast_brightness(image, alpha, beta)`. Para ello, debe convertir la imagen a float32, normalizar, aplicar la fórmula lineal  $g(x) = \alpha f(x) + \beta$ , hacer clipping para asegurar el rango [0, 1] y regresar a uint8. Note que no puede usar `cv2.convertScaleAbs`. Debe hacerlo con pura manipulación de matrices NumPy.

**\*Ver en archivo Task2\_HojaDeTrabajo1**

**Ejercicio 2:**

Implemente la función `manual_gamma_correction(image, gamma)`. Para ello:

- Implemente la ecuación  $V_{out} = V_{in}^{\gamma}$
- Recuerde que la operación de potencia es costosa. Aunque en producción usaríamos una LUT (Look- Up Table), aquí quiero que vectorice la operación de potencia sobre la matriz flotante.

**\*Ver en archivo Task2\_HojaDeTrabajo1**

**Ejercicio 3:**

Implemente la función `hsv_segmentation(image)`. Para ello:

- Cargue una imagen de prueba (algo colorido).
- Conviértala a HSV.
- Defina manualmente los rangos (lower\_bound, upper\_bound) para aislar un color específico (ej. El amarillo de un banano o el rojo de una manzana).
- Genere una máscara binaria y úsela para mostrar solo el objeto segmentado sobre un fondo negro.

**\*Ver en archivo Task2\_HojaDeTrabajo1**

### **Task 3:**

**1. En la diapositiva 15 se mencionó que "Iterar píxel a píxel en Python es un Pecado Capital". Explique en términos de gestión de memoria y CPU por qué una operación vectorizada en NumPy es órdenes de magnitud más rápida que un for loop.**

Una operación vectorizada es más rápida que un for loop ya que al usar una operación de NumPy este realiza solamente una operación para toda la imagen, ya que NumPy trabaja de forma compilada, que es mucho más eficiente que realizar una operación pixel por pixel. Relacionando esto con la CPU y la memoria al realizar solamente una operación compilada se puede aprovechar de mejor manera el caché del procesador y se trabaja directamente sobre bloques contiguos en la memoria.

**2. Al visualizar imágenes con matplotlib, ¿qué sucede si olvida que OpenCV carga las imágenes en formato BGR? ¿Cómo se ve visualmente el error?**

Matplotlib espera una imagen en formato RGB, pero OpenCV retorna en formato BGR. Si mostramos la imagen sin convertir a RGB lo que va a pasar es que se intercambia el color rojo (R) con el color azul (B). De manera visual veríamos los colores rojos como azules y los colores azules como rojos.

**3. Al visualizar imágenes con matplotlib, ¿qué sucede si olvida que OpenCV carga las imágenes en formato BGR? ¿Cómo se ve visualmente el error?**

Matplotlib espera una imagen en formato RGB, pero OpenCV retorna en formato BGR. Si mostramos la imagen sin convertir a RGB lo que va a pasar es que se intercambia el color rojo (R) con el color azul (B). De manera visual veríamos los colores rojos como azules y los colores azules como rojos.

**Referencias:**

Locke, J. (2024, 2 febrero). Key Features and Benefits of 5G Technology. Digi. <https://es.digi.com/blog/post/5g-benefits#:~:text=La%20banda%20baja%205G%20incluye,y%20admiten%20banda%20ancha%20m%C3%B3vil.&text=Banda%20baja%20%2D%20Esta%20gama%20de,en%20recintos%20grandes%20y%20concurridos>.