CC3182 – Visión por Computadora

Hoja de Trabajo 1

Instrucciones

- Esta es una actividad en grupos de no más de 3 integrantes.
  - Recuerden **unirse al grupo de canvas**
- No se permitirá ni se aceptará cualquier indicio de copia. De presentarse, se procederá según el reglamento correspondiente.
- Tendrán hasta el día indicado en Canvas.
  - No se confíen, aprovechen el tiempo en clase para entender todos los ejercicios y avanzar lo más posible.

Task 1 - Análisis Teórico

Considere cada uno de los siguientes escenarios y responda según corresponda:

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).
   b. Calcule el ancho de banda necesario (en Gbps) para transmitir el flujo de las 8 cámaras al procesador central sin compresión.
   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?
   d. Basado en su resultado, ¿es viable enviar estos datos "crudos" a la nube en tiempo real usando 5G? Justifique
2. Considere un píxel con valor de intensidad Iin=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 γ=0.5 (para expandir sombras).
   II. Ajuste Lineal con ganancia α=1.2 y brillo β=-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.
   b. ¿Hubo saturación (clipping) en el proceso?
   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?
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.
   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).
   c. Calcule la diferencia absoluta canal por canal en HSV
   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".

Task 2 – Práctica

Su objetivo es implementar un pipeline de pre-procesamiento manual, manipulando tensores y gestionando tipos de datos (uint8 vs float32) sin utilizar funciones de "caja negra" para la matemática. Puede utilizar el esqueleto de código lab_semana1.py que se adjuntó en el portal. Se permite el uso de numpy, opencv-python (solo para I/O y conversiones de espacio de color), matplotlib.

Ejercicio 1: Contraste y Brillo Vectorizado

Implemente la función manual_contrast_brightness(image, alpha, beta). Para ello, debe convertir la imagen a float32, normalizar, aplicar la fórmula lineal $g(x) = \alpha f(x) +$, 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.

Ejercicio 2: Corrección Gamma Manual

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.

Ejercicio 3: Segmentación Cromática

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.

Task 3 – Preguntas Post-Práctica

Responda brevemente (máximo 3 líneas por respuesta:

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

NOTA: Limiten el uso de IA generativa. Intenten primero buscar en fuentes de internet y si en verdad necesitan usarla, asegúrense de colcar el prompt que utilizar para cada task donde corresponda, así como una explicación de por qué ese prompt funcionó.

Entregas en Canvas

1. Documento PDF con las respuestas a cada task
2. Archivo .py, o link a repositorio de GitHub (No se acepta entregas en otros medios)

Evaluación

1. [1.5 pt] Task 1
2. [0.5 pt] Task 2
3. [0.5 pt] Task 3

Total 2.5 pts