

CC3182 – Visión por Computadora

## Hoja de Trabajo 2

### 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.
- **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. A menos que las instrucciones indiquen algo diferente.

### Task 1 - Análisis

Considere que usted está diseñando el sistema de visión para un robot de almacén que debe moverse entre estanterías para recoger productos. El robot tiene dos cámaras frontales:

1. Durante una prueba, el robot gira sobre su propio eje para escanear el entorno. El ingeniero junior a tu cargo sugiere usar Homografías para medir la distancia a los objetos mientras el robot gira. ¿Es este un enfoque correcto? Justifique su respuesta utilizando los conceptos de C1, C2 y Paralaje.
2. Si el robot comienza a avanzar (traslación) y detectas que la disparidad ( $d$ ) de una caja aumenta repentinamente entre el frame  $t$  y el frame  $t+1$ , ¿qué puedes inferir sobre la distancia ( $Z$ ) entre el robot y la caja? ¿Qué riesgo industrial implicaría un error en el cálculo de esta disparidad?

### Task 2 – Ingeniería de Dimensiones

Como director de proyectos, debe asegurar que los modelos de IA quepan en la memoria de los dispositivos (Edge Computing). Por ello, tiene una imagen de entrada de alta resolución proveniente de una cámara industrial de  $1280 \times 720$  píxeles. Se aplica una capa convolucional con los siguientes hiperparámetros

- Tamaño del Filtro (F):  $5 \times 5$
- Padding (P): 2
- Stride (S): 2

Considerando esto, respondan:

1. Utilizando la fórmula vista en clase, calcule las dimensiones ( $W_{out}, H_{out}$ ) del Mapa de Características resultante. Muestra el procedimiento.
2. ¿Qué sucedería con el tamaño de la salida si decides cambiar el Padding a  $P=0$  (Valid Padding)? ¿Cómo afectaría esto a la información de los bordes de la imagen (donde suelen estar las referencias de las paredes del almacén)?



### Task 3 – Criterio de Diseño

En la industria, el balance entre precisión y velocidad es clave. Analice los siguientes escenarios::

1. Usted está desarrollando un sistema de detección de grietas microscópicas en motores de avión. ¿Qué combinación de Stride y Pooling recomendaría para no perder detalles críticos en las primeras capas de la red? Justifique técnicamente..
2. Un cliente le pide que el sistema funcione en un procesador muy limitado (como una cámara inteligente con poca RAM). Explique cómo podrías utilizar el Stride y el Max Pooling estratégicamente para reducir la carga computacional sin eliminar las características más fuertes (activaciones) del Mapa de Características

### Task 4 – Implementación Práctica

Con esta parte se busca que puedan comprender la mecánica interna de la operación convolucional sin depender de librerías de alto nivel para la lógica central. Por ello realice lo siguiente:

- Utilizando un lenguaje de programación (Python) y librerías básicas para manejo de matrices (como NumPy), implementa una función llamada `manual_convolution(image, kernel, stride, padding)`.
- **Requisitos de la implementación:**
  - La función debe recibir una matriz 2D (imagen en escala de grises) y un filtro (kernel) de tamaño  $N \times N$ .
  - Debe aplicar primero el Zero-padding a la imagen de entrada según el valor de P.
  - Debe recorrer la imagen aplicando el producto punto (suma de productos elementales) respetando el Stride indicado.
  - La función debe retornar la matriz resultante (Feature Map).
- **Prueba de validación:** Defina un filtro de detección de bordes verticales (filtro de Sobel o similar) y aplíquelo a una imagen pequeña de prueba.
- **Entrega:** Suba su Jupyter Notebook con comentarios explicando cómo el desplazamiento del kernel afecta el tamaño de la matriz de salida.

#### 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. [0.5 pt] Task 1
2. [0.5 pt] Task 2
3. [0.5 pt] Task 3
4. [1.0 pt] Task 4

Total 2.5 pts

