

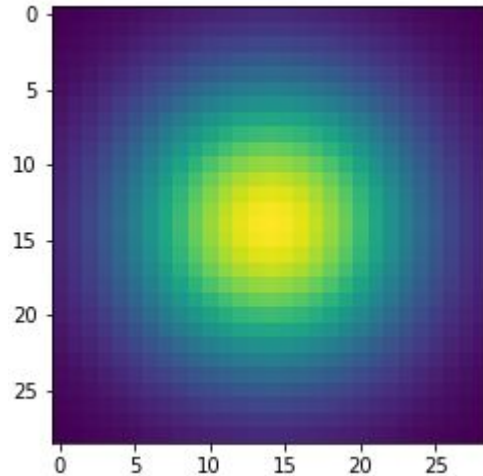
# Visión por Computadora

## Proyecto 1

Daniel Velázquez  
daniel.velazquez@fiuni.edu.py

# Parte 1: Filtrado de imágenes.

Kernel Gaussiano



Describe tu solución de la implementación `my_conv2d_numpy()` con tus palabras.

- Primero creo mi imagen resultante con ceros del mismo shape que la imagen de entrada.
- Luego obtengo la cantidad de columnas y filas que tiene el kernel para saber que tanto sumar a mi posición  $(i, j)$  de mi imagen con padding zero
- Obtengo que tanto padding necesito para top, bottom por un lado, e izq y der por otro lado de acuerdo a la división realizada. Con esos datos uso la función de OpenCV para generar el padding
- Por último con un for itero mi matriz obteniendo la region de interes, multiplicando por mi kernel, sumando sus valores y asigno ese valor a la posición  $(i,j)$  actual

# Parte 1: Filtrado de imágenes.

Filtro identidad



Suavizado con el Filtro de caja

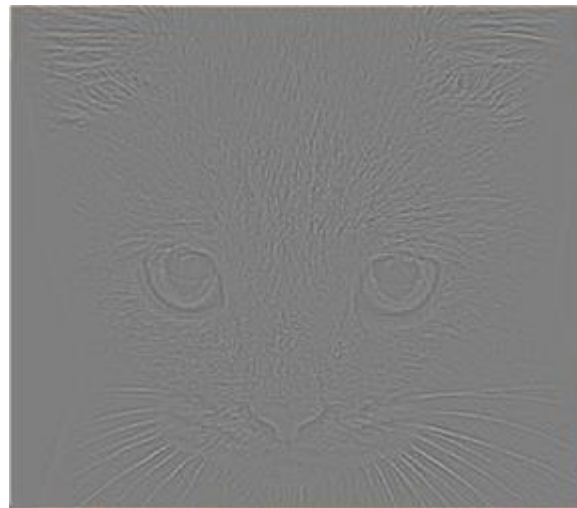


# Parte 1: Filtrado de imágenes.

Filtro de Sobel



Filtro Laplaciano Discreto



# Parte 1: Imágenes Híbridas

Describe los 3 pasos principales para crear una imagen híbrida (`create_hybrid_image()`). Explica cómo aseguramos que el rango de las imágenes es el apropiado para mostrarlo en matplotlib

1. Aplicar el filtro gaussiano a la imagen 1 (frecuencias bajas)
2. Restar la el resultado del filtrado de la imagen 2 a la imagen 2 para obtener sus frecuencias altas
3. Sumar ambas imágenes

Para asegurar que el rango de las imágenes sea el correcto se aplica clipping (`np.clip(hybrid, 0,1)`) a la imagen híbrida que se retorna

Las imágenes las guarde usando mi notebook (`test_part1`)

**cat+dog**



Valor cutoff frequency = 7

# Parte 1: Imágenes Híbridas

**moto+bici**



Valor cutoff frequency = 7

**avión + ave**



Valor cutoff frequency = 7

# Parte 1: Imágenes Híbridas

Einstein + Marilyn



Valor cutoff frequency = 7

submarino + pez



Valor cutoff frequency = 7



## Parte 2: Imágenes Híbridas con PyTorch

Cat + dog



Motorcycle + Bicycle





## Parte 2: Imágenes Híbridas con PyTorch

Plan + Bird



Einstein + Marilyn



# Parte 2: Imágenes Híbridas con PyTorch

## Submarine + Fish



## Part1 vs. Part2.

Comparar el tiempo de ejecución de las partes 1 y 2 que obtuviste en la notebook. ¿Cuál método resultó ser más rápido?

Part 1: 4.615 seconds

Part 2: 0.638 seconds

# Parte 3

Considera una imagen 5x5 de 1 canal y un filtro 3x3. ¿Cuáles son las dimensiones de la convolución si usamos los siguientes parámetros?

Stride=1, padding=0 -> 3x3

Stride = 2, padding =0 -> 2x2

Stride = 1, padding =1 -> 5x5

Stride =2, padding=1 -> 3x3

¿Cuáles son las dimensiones de entrada y salida de la convolución de la imagen del perro con un filtro 3x3 y los siguientes parámetros?

entrada = 361x410

Stride=1, padding=0 -> 359x408

Stride = 2, padding =0 -> 180x204

Stride = 1, padding =1 -> 361x410

Stride =2, padding=1 -> 181x205

## Parte 3

inserte el resultado del filtrado del perro aquí. El resultado 1



inserte el resultado del filtrado del perro aquí. El resultado 2



## Parte 3

inserte el resultado del filtrado del perro aquí. El resultado 3

Screenshot



inserte el resultado del filtrado del perro aquí. El resultado 3

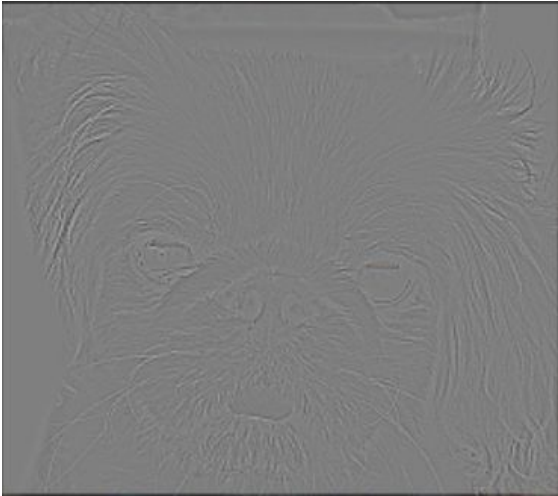
Imagen guardada



## Parte 3

inserte el resultado del filtrado del perro aquí. El resultado 4

Screenshot



inserte el resultado del filtrado del perro aquí. El resultado 4

Imagen guardada



# Conclusión

¿De qué manera afecta el resultado de las imágenes híbridas el hecho de usar diferentes cutoff\_frequencies?

Según lo que probé cuanto más alta sea el cutoff frequency se resalta más la imagen de intensidades altas

¿Qué es lo que más te costó y porque? ¿Que no pudiste implementar y porque?

Lo que mas me costo al comienzo fue hacer que mi convolución acepte kernels e imágenes no cuadradas porque me base en el código que había hecho en procesamiento de imágenes y también de una página pero que solo aceptaba imágenes y kernels cuadrados. Luego para la parte con pytorch fue un poco complicado entender cómo usar sus métodos, formatear las imágenes/kernels para que pueda aceptarlos.

Implementar todos los métodos requeridos