Práctica #12

Materia: Sistemas de Visión Artificial

Grupo: 7°E1

Isaac Alejandro Gutiérrez Huerta 19110198

19/06/2022

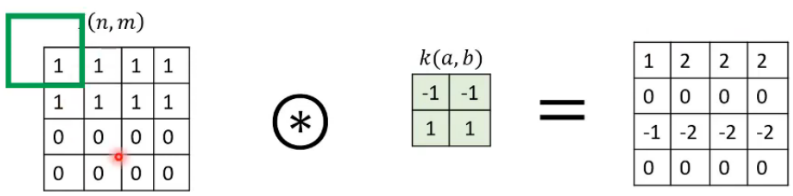
Práctica #12

Objetivo:

Objetivo demostrar paso a paso del teorema de la convolución.

Teoría:

$$S(i,j) = \sum_{n}\sum_{m} I(n,m).\,k(i-n, j-m)$$



Lo que se hace en está haciendo en las operaciones es multiplicar los números que se solapan y posteriormente sumar los resultados. En este caso el cuadrado verde representa la matriz kernel, nos podemos dar cuenta que la única celda que se solapa con la matriz inicial es la 2,2 (0+0+0+1). Por lo tanto, el resultado de la matriz resultante en la posición (1,1) es 1.

Para la siguiente celda el kernel se desplaza una posición hacia la izquierda y se efectúan nuevamente las operaciones.

Mientras más iteraciones se hagan, el resultado será más preciso.

En el siguiente código se utiliza el kernel **k= ([[-1,-1,-1],[-1,8,-1],[-1,-1,-1]])** que está pensado para la detección de bordes. Dependiendo del kernel utilizado será el resultado obtenido.

Asimismo, en este ejemplo se realizaron 3 iteraciones de la operación de convolución, mientras más iteraciones mejor será el resultado.

Código:

```
#Isaac Alejandro Gutiérrez Huerta 19110198 7E1

#Sistemas de Visión Artificial
```

```
import os
from scipy import signal
import matplotlib.pyplot as plt
import numpy as np
from skimage.color import rgb2gray
from skimage import io

Matriz = np.array([[1,1,1,1], [1,1,1,1], [0,0,0,0], [0,0,0,0]])
kernel = np.array([[1,1], [-1,-1]])
temp = signal.convolve2d(Matriz , kernel, mode='same')
print(temp)

def show_convolve2d(imagen, kernel):

    plt.ion()

    imagen_list = []
    for d in range(3):
        temp = signal.convolve2d(imagen[:,:,d] , kernel,  boundary='symm',mode='same')
        imagen_list.append(temp)

    imagen_filt = np.stack(imagen_list, axis=2)
    imagen_filt[imagen_filt > 255] = 255
    imagen_filt[imagen_filt < 0] = 0
    imagen_filt = imagen_filt.astype("uint8")

    plt.subplot(1,2,1)
    io.imshow(imagen)
    plt.axis('off')

    plt.subplot(1,2,2)
    io.imshow(imagen_filt)
```

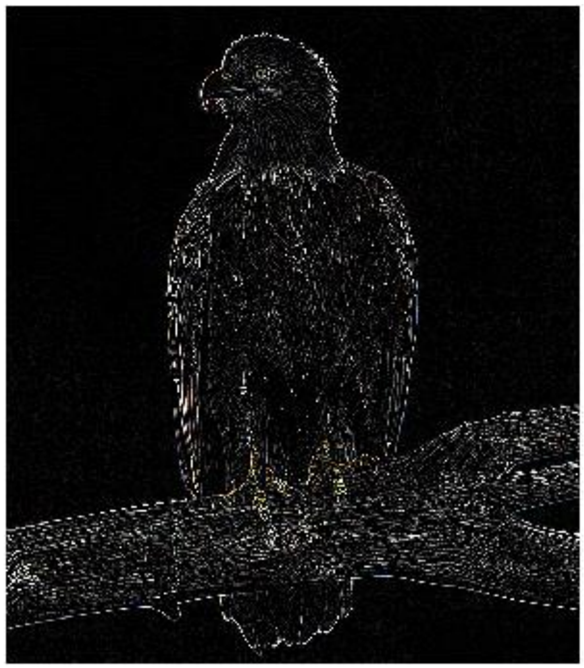
```
    plt.axis('off')

    io.show()

filename = os.path.join('Aguila2.png')
imagen = io.imread(filename)

k=np.array([[-1,-1,-1],[-1,8,-1],[-1,-1,-1]])
show_convolve2d(imagen ,k)
```

Resultados:



Matemática:

El teorema de convolución dice que la transformada de Laplace de la convolución de dos funciones es igual al producto de sus transformadas individuales. La cual se describe de la siguiente manera:

$$\mathscr{L}[f(t)\cdot g(t)] = \mathscr{L}[f(t)]\cdot\mathscr{L}[g(t)]$$

Y se define de la siguiente forma:

$$\mathscr{L}[f(t)] = \int_0^\infty f(t)e^{-st}dt$$

Lo cual es:

$$\mathscr{L}[f(t) \cdot g(t)] = \int_0^\infty (f(t) \cdot g(t))e^{-st}dt$$

O lo que es igual a

$$f(t) \cdot g(t) = \int_0^t f(x)g(t-x)dx$$

Si reemplazamos, nos quedaría de la siguiente manera:

$$\mathscr{L}[f(t) \cdot g(t)] = \int_0^\infty \int_0^t f(x)g(t-x)e^{-st}dxdt$$

Ahora se aplica el teorema de Fubini para cambiar el orden de la integración y obtener nuevas regiones:

$$\begin{array}{c|c} 0 < t < \infty & a < t < b \\ 0 < x < t & x_1(t) < x < x_2(t) \end{array}$$

$$\begin{array}{c|c} 0 < x < \infty & c < x < d \\ x < t < \infty & t_1(x) < t < t_2(x) \end{array}$$

Al reescribir la función y hacer los cambios de variables obtenemos:

$$\mathscr{L}[f(t) \cdot g(t)] = \int_0^\infty f(x) \int_0^\infty g(v)e^{-s(v+x)}dvdx$$

Al aplicar la ley de exponentes queda:

$$\mathscr{L}[f(t) \cdot g(t)] = \int_0^\infty f(x)e^{-sx} \int_0^\infty g(v)e^{-sv}dvdx$$

Y al aplicar la propiedad de producto de integrales las separamos:

$$\mathscr{L}[f(t) \cdot g(t)] = \int_0^\infty f(x)e^{-sx}dx \int_0^\infty g(v)e^{-sv}dv$$

Para finalizar obtenemos que

$$\mathscr{L}[f(t) \cdot g(t)] = \int_0^\infty f(x)e^{-sx}dx \int_0^\infty g(v)e^{-sv}dv = \mathscr{L}[f(t)] \cdot \mathscr{L}[g(t)]$$

Este teorema sirve en la visión artificial para analizar y procesar imágenes con un porcentaje alto de mejoramiento, entre más veces se haga la convolución, mejor resultado se obtendrá.

Enlace de GitHub:

https://github.com/IsaacGutierrezCETI/Practica-12.git