

Instituto Tecnológico y de Estudios Superiores de Monterrey



Visión para robots.

Tarea 1

Convolución 2D en imágenes en escala de grises

Profesor:

Dr. César Torres Huitzil

Alumnos:

Sergio Iván Villegas Arenas A01625055

09 de marzo de 2023

Metodología.

Para la realización de esta tarea se decidió usar la plataforma de Google Colab para programar.

Primero se buscó todo el código hecho en clase el cual nos pueda ser útil, entre estos se encontraba el como leer una imagen e imprimirla, el uso de la función `rgb2gray` para convertir la imagen en blanco y negro y la base de los nodos `for` para aplicar la convolución.

Después se investigo como realizar el “zero padding” en una convolución, encontrando un documento en Kaggle muy completo. En este se pudo encontrar el como se crea un kernel Gaussiano dado el tamaño del kernel y su sigma.

Y el como se realiza el Zero padding el cual consiste en crear un arreglo lleno de 0 del tamaño de la nuestra imagen + el tamaño de nuestro kernel menos 1. Este menos uno es necesario para que la imagen resultante termine del mismo tamaño que nuestra imagen. Y luego insertar nuestra imagen en dicho arreglo.

Una vez hecho esto, en nuestros nodos de ciclos `for` se calcula el promedio de la imagen completa.

Se busco información para comprender mejor el código encontrado y se descubrió la importancia del zero padding a la hora de realizar convoluciones, ya que sin este la matriz resultante queda de un tamaño menor a la original dada a las fronteras del tamaño de nuestro kernel y nuestra imagen, pero con el zero padding nos permite que nuestra imagen se permita mantener en el mismo tamaño.

En el caso de los kernels ya dados, de igual manera se realiza el zero padding con la única diferencia que en nuestros nodos en nuestra salida se calcula la sumatoria de las multiplicaciones de los pixeles de nuestra imagen multiplicada por nuestro kernel.

Una vez comprobando que nuestro código funcione correctamente, se realizaron las funciones para que se den los datos de entrada y recibamos nuestra imagen ya convolucionada.

Lo ultimo que se hizo fue la interpolación de nuestra imagen para que en vez de que este en valores entre 0 y 1, este con valores entre 0 y 255.

Resultados obtenidos.

Imagen original



	Sigma=1	Sigma=3	Sigma=8
N=3	<p>Image size: 50540</p>	<p>Image size: 50540</p>	<p>Image size: 50540</p>
N=5	<p>Image size: 50540</p>	<p>Image size: 50540</p>	<p>Image size: 50540</p>
N=9	<p>Image size: 50540</p>	<p>Image size: 50540</p>	<p>Image size: 50540</p>

Image size: 50540

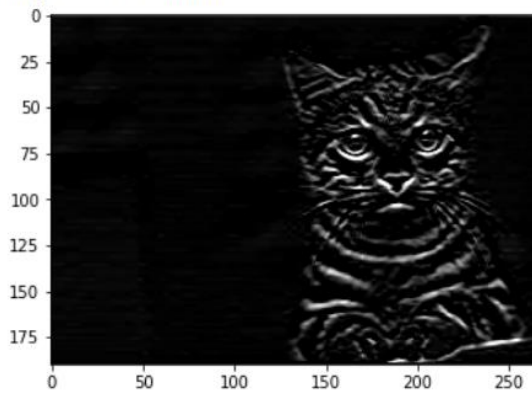
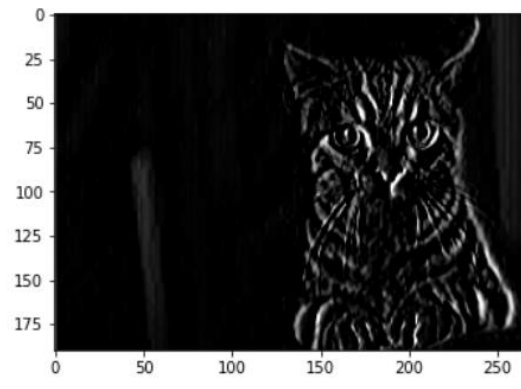


Image size: 50540



Con los resultados obtenidos primero podemos comprobar que nuestra imagen de salida es del mismo tamaño de entrada.

En el caso de los Gaussianos, se pudo observar que mientras mas grande es el tamaño del kernel mas se difumina la imagen, mas sin embargo no se nota una gran diferencia por no decir que es casi nula, entre las diferencias de sigma.

Dado el documento de Kaggle, se descubrió que para un mejor resultado de convolución debemos rotar nuestras kernels, esto se pudo comprobar con los kernels que nos permiten diferenciar los bordes, donde se veían mejores resultados a la hora de rotar nuestro kernel. Mas sin embargo en las kernels gaussianos no se encontré una diferencia significativa al rotar o no el kernel.

Conclusiones.

Esta practica ayudo mucho para entender el concepto de convolución, y sirvió un poco mas como practica de investigación ya que el código encontrado en la página de Kaggle estaba muy completo, pero sin tanta información, la cual es importante para entender por qué hace que.

Además, que ayudo para aprender más de Python, ya que no es un lenguaje el cual domine, e investigar las operaciones encontradas me permitió tener mayor conocimiento en este lenguaje. Lo que facilitara la comprensión de códigos más adelante.

Referencias.

<https://medium.com/@draj0718/zero-padding-in-convolutional-neural-networks-bf1410438e99>

<https://www.kaggle.com/code/thesherpafromalabama/manual-image-convolution-zero-padding>